

Doctoral Thesis

Research on Image Processing for Assisting
the Visually Impaired to Access Visual
Information

Jianjun CHEN

Toyama Prefectural University

September 2015

Abstract

In 2010, the World Health Organization (WHO) reported that there are approximately 285 million people worldwide, who suffer visual impairment. Every year, this number grows by up 2 million due to eye diseases, accidents, aging and the growth of newborn population. Vision loss affects almost all activities of daily living, for instance, it becomes difficult or impossible for visually impaired people to read, walk, drive or recognize objects and find places. Therefore, if assistive technology can support visually impaired people in at least one of these tasks, it is going to make a very relevant social impact. This dissertation turns to the development of technology for assisting the visually impaired to access visual information.

Visual information is widely used for studying, route finding, finding public places, and more. However, visually impaired people, especially blind people, are difficult to access visual information unless it is represented non-visually, such as with Braille, tactile graphics, or speech. In this thesis, we focus on development of image processing to assist tactile graphic production and scene text reading.

For human beings, it is difficult to study mathematics, physics, chemistry, and biology through text only. Therefore, graphs and figures are frequently used to present visual information in textbooks. However, most of these graphs and figures are in visual form, they cannot be utilized by visually impaired students. Through tactile graphics, figures can be understood by the visually impaired. This is because tactile graphics are represented by raised patterns which can be felt with fingertips. However, production of tactile graphics is not so simple a task, it needs professional knowledge of its corresponding field. Most of the work for producing tactile graphic teaching materials is done by teachers, while some of the work is done by volunteers. Until now, most tactile graphics are produced using less intelligent computer-aided systems. Therefore,

technologies for automatically producing tactile graphic teaching materials are needed. In this thesis, the first aim is to develop a system for automatically translating mathematical graphs into tactile graphics. This is because mathematical graphs are often line drawings. So it is possible that computers are able to recognize mathematical graphs from printed materials for producing tactile graphics.

For the figures in physics, chemistry, and biology textbooks, they are difficult to be translated into tactile graphics due to the following reasons: for example, (1) figures in physics textbooks are usually not line drawings, they are difficult to be translated into tactile graphics by using image processing techniques; (2) the structures of figures in physics textbooks are usually complex, but tactile graphics must be simple for reading; (3) the ability to read tactile graphics deeply depends on individuals, providing tactile graphics individually is needed. Therefore, such figures should be redrawn before producing tactile graphics. The second aim of this thesis is to develop a system for automating translation of hand-drawn figures into tactile graphics.

Signs and public notices are ubiquitous indoors and outdoors, and they are often used for route finding, finding public places and other locations. However, information on signs is inaccessible to many visually impaired people. Therefore, technology developed for automatically reading text from natural scenery becomes an important application for assisting visually impaired people. Thus in this thesis, the third aim is to propose a system for automated text reading from natural scenery.

Based on the reasons above, this dissertation focuses especially on the use of image processing to support visually impaired people to access visual information. In this thesis, we first propose a method for automatically translating mathematical graphs into tactile graphics. Second, a method for automating translation of hand-drawn figures into tactile graphics is proposed. Finally, we propose a new method for automated text reading from natural scene images.

Contents

Chapter 1	1
Introduction	1
1.1 Visually Impaired People	1
1.2 A Review for Assistive Technologies	2
1.2.1 Corrective Lens for Vision Assistance	2
1.2.2 Assistance in Mobility and Orientation	2
1.2.2.1 Aids for Mobility	2
1.2.2.2 Aids for Orientation	3
1.2.2.3 Aids for Accessing Environment	4
1.2.3 Assistance in Verbal Information Access	4
1.2.3.1 Aids for Reading Printed Text	4
1.2.3.2 Aids for Reading Text on Screen	5
1.2.4 Assistance in Visual Information Access	5
1.2.5 Assistance in Object Recognition	6
1.3 Research Background and Objective	7
1.3.1 Background	7
1.3.1.1 Tactile Graphic Production	7
1.3.1.2 Scene Text Reading	8
1.3.2 Objectives	8
1.3.2.1 Translation of Mathematical Graphs into Tactile Graphics	9
1.3.2.2 Translation of Hand-drawn Figures into Tactile Graphics	9
1.3.2.3 Extraction of Text from Natural Scene Images	11
1.4 Organization	12
References	13
Chapter 2	19
Tactile Graphics	19
2.1 Introduction	19
2.2 Types and Forms of Tactile Graphics	21
2.3 Guidelines for the Design of Tactile Graphics	23
2.3.1 Characteristics for Tactile Graphics	23
2.3.2 Basic Principles for the Design of Tactile Graphics	23
2.3.3 Tactile Graphic Symbols	24
References	26
Chapter 3	27
Automatic Translation of Mathematical Graphs into Tactile Graphics	27

3.1	Introduction.....	27
3.2	Separation of Mathematical Graphs	30
3.3	Extraction of Broken Line Graph Elements.....	31
3.3.1	Dotted Line Classification	32
3.3.2	Chain Line Classification.....	33
3.3.3	Broken Line Classification.....	35
3.3.4	Merging Clusters.....	35
3.4	Extraction of Solid Line Graph Elements	39
3.4.1	Segmentation of Large Elements	40
3.4.2	Merging Primitive Elements	41
3.5	Fitting and Classifying Graph Element.....	42
3.5.1	Straight Line, Circle and Arc Classification	43
3.5.2	Ellipse Classification	43
3.5.3	Cubic Bézier Curve Fitting	44
3.6	Experimental Results	46
3.6.1	Results of Broken Line Extraction.....	46
3.6.2	Results of Solid Line Extraction	49
3.6.3	Results of Graph Element Fitting.....	50
3.6.4	SVG and Edel Document Production	50
3.7	Summary	51
	References	52
	Chapter 4.....	55
	Automatic Translation of Hand-drawn Figures into Tactile Graphics.....	55
4.1	Introduction.....	55
4.2	Outline For Our Method and Basic Procedures	56
4.2.1	Preprocessing and Traffic Signal Detection	58
4.2.2	Segmentation and Shape Classification	60
4.3	Object Classification.....	63
4.3.1	Arrow Classification	64
4.3.2	Cross Classification	66
4.3.3	Route and Railway Classification	67
4.4	Experimental Results	69
4.5	Usability Evaluation for Our System	70
4.5.1	Method.....	70
4.5.2	Results.....	72
4.6	Summary	76
	References	78
	Chapter 5.....	81
	Text Extraction from Natural Scene Images.....	81
5.1	Introduction.....	81
5.2	Related Studies in Text Extraction.....	83
5.3	Preprocessing.....	84

5.4	Homogeneous Region Segmentation	85
5.4.1	Contrast Enhancement Using Toggle Mapping.....	85
5.4.2	Grayscale Image Smoothing	85
5.4.3	Homogeneous Region Segmentation	86
5.4.4	Candidate Signboard Region Detection.....	86
5.5	Character Detection from Candidate Signboard Regions	89
5.5.1	Edge Detection.....	89
5.5.2	Single Character Detection	91
5.5.3	Character Classification	94
5.6	Experiment and Results.....	97
5.6.1	Experimental Images.....	97
5.6.2	Evaluation Results of Single Character Detection	97
5.6.3	Discussion.....	99
5.7	Summary	99
	References	102
	Chapter 6.....	105
	Conclusions	105
6.1	Summary	105
6.2	Future Works	106
	Acknowledgements	109
	List of Publications.....	111

Chapter 1

Introduction

This chapter first describes the situation of visually impaired people worldwide. Then, a general overview of the assistive technologies for the visually impaired is depicted and discussed. After that, the objectives of this dissertation are proposed. Finally, the contents of each chapter are summarized.

1.1 Visually Impaired People

There is a saying, “eyes are the windows to the soul”. For human beings with normal vision, they obtain information depend heavily on vision in daily living. Unfortunately, blind people obtain the information only through touching, listening, smelling, tasting, and moving.

In 2010, World Health Organization (WHO) made a survey worldwide. The statistical result showed that there were approximately 285 million people suffer visual impairment, of whom 39 million were blind; 246 million had low vision [1]. About 90% of the visually impaired lived in developing countries. Approximately 65% of visually impaired people and 82% of blind people were aged 50 and older. An estimated 19 million of the visually impaired were children aged under 15. The number of visually impaired people is increasing rapidly with the growth of newborn population and so on; and every year, this number grows by up to 2 million worldwide.

Because of visual disability, the abilities for performing daily tasks and interacting with surrounding world are limited or influenced. For example, it becomes difficult or impossible for visually impaired people to read, walk, drive, or recognize objects and find

places. Normally, in daily life, study and work, the visually impaired need assistance from their family members, friends and caregivers. Therefore, if assistive technology can support visually impaired people in at least one of daily tasks, it is going to make a very relevant social impact [2].

1.2 A Review for Assistive Technologies

Until now, a number of technologies have been developed to support visually impaired people to live a more independent life. In the following subsections, part of typical assistive technologies are reviewed and categorized.

1.2.1 Corrective Lens for Vision Assistance

Corrective lenses are glass or plastic worn on or in front of the eye, and they are mainly used to treat myopia, hyperopia, presbyopia and astigmatism [3]. Corrective lenses aim to focus the object image at a single point on the retina. Thus, the use of corrective lenses is to improve vision by improving focus and correcting blur. However, corrective lenses are useless for blind people.

1.2.2 Assistance in Mobility and Orientation

Visually impaired people, especially the blind hope to walk or travel independently. Therefore, many technologies have been developed for the solution of walking assistance in the following three aspects: obstacle avoidance; navigation and orientation; and accessing environment.

1.2.2.1 Aids for Mobility

Mobility can be defined as the capability to move safely, quickly, and effectively from one place to another. Not only blind people need the aids of mobility, but also some low-vision individuals may also need such support to help them walk without tripping or falling, cross streets, use public transportation and more.

One of the most traditional mobility tools is the long cane (i.e., white cane) which is

used to detect obstacles in the path of the visually impaired. However, a long cane provides only limited information about the environment due to its short length. Dogs can be trained to lead visually impaired people around obstacles, so they are called guide dogs. However, it is time consuming and cost expensive for training a dog, and a low probability of success restricts their use. Guide dogs are usually not responsible for orientation in a large environment.

With the development of modern technology, many different types of assistances are proposed to support the visually impaired in mobility. For example, Electronic Travel Aids (ETAs) are commonly introduced to help visually impaired people avoid obstacles [4, 5]. For ETA devices, different types of range sensors are utilized to provide an audio or vibration in response while receiving reflected waves to nearby objects [6 - 12]. In recent years, a number of computer vision-based [13 - 18] and digital video camera-based [19 - 22] ETAs have been developed. However, multiple factors contribute to low market acceptance for these devices, such as cost, portability and performance.

1.2.2.2 Aids for Orientation

Way-finding (or orientation) refers to the ability to know one's location in an environment, and to find a route to a destination. In an unfamiliar environment, landmarks and signs are utilized to sighted people for way-finding, but they cannot be used directly for blind people to obtain the orientation and track the location.

Braille for the feet and traffic lights with acoustic devices are common methods for assisting the visually impaired to walk outdoors. But they require wide laying and installation. Nowadays, a number of assistive navigation systems are designed based on the application of Global Positioning System (GPS). However, GPS devices have a similar characteristic: poor resolution in urban-environments and unavailable indoors [23]. A robotic indoor navigation system is proposed by Kulyukin et al. [24]. In this system, Radio Frequency IDentification (RFID) is utilized to read IC tags, but it is difficult for blind people to locate the tags at close range. Wi-Fi techniques are gaining momentum now, and they are expected to provide solutions for indoor localization [25]. However, Wi-Fi localization requires widely deployment in order to keep complete

coverage, and it is a time consuming process for calibration.

1.2.2.3 Aids for Accessing Environment

Street signages, traffic information signs and other public notices are commonly utilized for sighted people to find public places, and locate entrances or exists. However, these visual information are usually inaccessible to many visually impaired people. Therefore, a number of devices have been developed to support the visually impaired to access public places and locations. For example, audible indicators are used to alert doors being open and the time to cross pedestrian crossings. Infrared talking signs can be widely installed to aid blind people to cross light-controlled intersections and locate bus stops and so on. Through a hand-held receiver, blind users can receive the broadcasted information [26]. But infrared signs require costly installation and maintenance. Tactile maps [27] or talking tactile maps [28] also are widely used to indicate the layout of a street environment.

Recent years, the technology developed for automatic reading text from natural scene images becomes an important application in assisting visually impaired people. Therefore, several aid systems have been developed to help visually impaired people read text from natural sceneries or product labels [29 - 35].

1.2.3 Assistance in Verbal Information Access

Reading is an essential daily task, but people with vision disabilities often are difficult to access the vast array of verbal information that sighted people can obtain easily. Verbal information mainly includes books, newspapers, documents, bills, and menus. Therefore, assistive technology should be developed to increase the accessibility for verbal information.

1.2.3.1 Aids for Reading Printed Text

Visually impaired people to access printed text mainly through the following ways: sighted people is asked to narrate the text; magnifier is used to enlarge printed text; translator is applied to translate printed text into Braille or audible speech. With the

progress of technology, the first method is on the wane. Magnification method is commonly an application of optical lenses, but it is useless for individuals who are completely blind.

Currently, Optical Character Recognition (OCR) based devices are developed to recognize printed text. First, printed text is captured by a digital camera; and then the digital image is recognized to generate a text file; finally, the text file is turned into Braille (embossed onto heavy paper or presented with a tactile display such as DotView) or speech (outputted through a voice synthesizer). Optical to TActile CONverter (Optacon) is a reading aided device that can translate printed text into Braille [36]. Another aided reading system is Moon. In this system, the embossed letter symbols are presented similar to ordinary printed types. Although the embossed letters can be read by many visually impaired people with less training, they are not very widespread [37].

Digital Accessible Information SYstem (DAISY) aims to provide digital audiobooks for people with print disabilities [38]. At present, eXtensible Markup Language (XML) and MP3 are recommended as DAISY standard to represent text content and audio content. An XML format can be easily converted into another format, such as Braille and audio format. For printed books, sighted people can easier to browse the table of content or index, skim the text, and locate immediately specific parts they are seeking. However, DAISY does allow these performances.

1.2.3.2 Aids for Reading Text on Screen

Computer operation and web surfing have become an essential task in everyday life. For visually impaired people, especially blind people are difficult to operate a computer especially surf the web.

Screen readers are a form of assistive technology that allow visually impaired people to read the content displayed on the computer screen through a speech synthesizer or Braille display. A screen reader is an interface which bridges the computer operation system and the user's command [39 - 41].

1.2.4 Assistance in Visual Information Access

Visual information generally includes pictures, figures, graphs, diagrams, and colors and so on. Visually impaired people to access these information mainly through the following two ways: (1) touching, that is, pictures, diagrams, graphs, and figures have to be presented as tactile graphics; (2) hearing, so, the visually impaired need to be told the name of colors.

In recent years, several mobile camera phones and computers based colour identifiers have been proposed to recognize colours and tell the names of colours [42]. Currently, DAISY accepts SVG as the standard for digital figures. However, there is not technology developed for recognizing and translating printed figures into SVG format. Krufka and Barner [43], proposed an approach for automating producing tactile graphics from vector graphic images. However, many images are not yet in this format, and it is often not easy to get digital files, such as LaTeX files, from the publishers. Therefore, a printed book often needs to be scanned and recognized before translating it into Braille, tactile graphics or speech. Tactile Graphics Assistant (TGA) is a software aims to translate bitmap images into tactile representations [44]. However, in the workflow of TGA, it needs human intervention and validation between each steps [45]. For example, the software of Photoshop is assumed to be applied for manipulating pure graphics.

1.2.5 Assistance in Object Recognition

In everyday life, we come into contact with a lot of objects in surrounding environment, for instance, we have to find commodities while shopping. Therefore, in order to support the visually impaired to recognize objects, in recent years, a new effort by the computer vision community has been made in recognizing generic objects from images. For example, ShelfScanner [46] is a system developed to assist visually impaired people in selecting commodities at a supermarket. In this system, captured images are recognized to select shopping items based on a known set. LookTel [47] is a software developed to detect and recognize objects such as packaged goods, CD covers, bank notes and more. However, until now, assistive technologies for visual object recognition achieved progress in only a few applications.

Not only above mentioned technologies are developed to assist the visually impaired in navigation, reading, and object recognition, but also there are still many technologies have been proposed to assist the visually impaired in accessing music, game, sport, driving and other. Visually impaired people are very diverse in terms of the degree of visual disabilities and the level of personal abilities, thus for different individuals their needs are very different. Therefore, there are still many tasks need to be done. In the following section, research motivation of this dissertation is proposed and discussed.

1.3 Research Background and Objective

Visual information is widely used for studying, route finding, finding public places and other. However, visually impaired people, especially blind people, are difficult to access visual information unless they are represented non-visually, such as with Braille, tactile graphics, and speech. In this dissertation, we focus on the development of image processing to assist tactile graphic production, and text reading from natural scene images.

1.3.1 Background

1.3.1.1 Tactile Graphic Production

For human beings, it is difficult to study mathematics, physics and chemistry only through text information, so graphs and figures are frequently used to help them to comprehend corresponding knowledge. However, most of these graphs and figures are presented in visual form, they cannot be utilized by many visually impaired students. Through tactile graphics, figures can be understood by the visually impaired. This is because tactile graphics are designed to be represented by raised surfaces that can be felt with fingertips.

At the present time, there are more than 300,000 visually impaired people in Japan. Almost all of the schools for the visually impaired need tactile graphics for teaching and studying. However, in 80% of the blind schools, there is no department for producing tactile graphic teaching materials [48]. To make tactile graphics prior knowledge of their corresponding fields is needed. Thus most of the work for producing tactile graphic

teaching materials is done by teachers while some of the work is done by volunteers. Until now, most tactile graphics are produced using less intelligent computer-aided systems, and some parts have to be made by hand, such as the painstaking work. Therefore, it is necessary to develop a computer-aided system for assisting the production of tactile graphics.

1.3.1.2 Scene Text Reading

Around us, there are a lot of signs and notices that are used for searching route, finding public places and accessing variety of services. However, these information are inaccessible to many visually impaired people. Most visually impaired people, especially the blind hope to walk outdoors independently. But they often lose their ways while first visiting a new place. Therefore, technologies developed for automatically reading scene text play a key role in assisting people with visual impairment, especially low vision people, to find locations.

Up to now, several portable or wearable camera-based devices have been proposed for assisting the visually impaired to read text in scenery. The characteristics of these devices are as follows: since such devices are based on software techniques that extract text information from scene images, installing infrastructures might not be required; we can provide them at a low price; and people with visual impairment can carry them easily when walking outdoors. These devices to read scene text through the following main procedures: capturing a scene image; extracting text from captured image; and translating extracted text into speech. However, each of these proposed devices has significant limitations. For instance, low accuracies for text region detection and assumptions for text layout. Therefore, it is necessary for us to develop a technology for automatically reading text from scene images with high accuracy.

1.3.2 Objectives

Based on the reasons above, this dissertation aims to the contributions as follows: (1) improve the accessibility for textbooks to visually impaired students; (2) support the visually impaired to access text information in the surrounding environment. Therefore,

we are committed to the following efforts: (1) developing a technology to support the translation of graphs and figures in textbooks into tactile graphics; (2) developing a technology for automatically reading text from natural scenery.

1.3.2.1 Translation of Mathematical Graphs into Tactile Graphics

Mathematical graphs are often line drawings, and they usually consist of the rays of the x -axis and the y -axis, straight lines, circles, arcs, ellipses and curves used to represent the mathematical expressions. Therefore, it is possible that computers are able to recognize mathematical graphs from printed materials for producing tactile graphics. Although many methods have been proposed for recognizing line drawings, in their methods, graphs have to satisfy many assumptions. For example, a graph has to be drawn in a box frame [49]; and a graph must not include broken lines or characters [50]. However, many mathematical graphs do not satisfy these assumptions. Almost all of the mathematical graphs have the following characteristics: (1) characters and mathematical formulas may be distributed in and around the graphs; (2) a character string or a mathematical formula may not lie on the horizontal orientation; (3) a graph may contain several types of broken lines. So the methods proposed in the past research cannot be applied to recognize mathematical graphs. Therefore, the first research issue contributes to the recognition of mathematical graphs.

1.3.2.2 Translation of Hand-drawn Figures into Tactile Graphics

We aim to develop a technology for assisting the production of tactile graphic teaching materials. Therefore, we focus on not only the mathematical tactile graphic production, but also the production of tactile graphics corresponding to the figures in physics, chemistry, and biology textbooks. However, we found that, for example, figures in physics textbooks are often not line drawings, they often have the following characteristics. (1) The figures in physics textbooks are often complex. But tactile graphics must be simple for reading and understanding. So original figures have to be simplified before producing tactile graphics. (2) The ability of reading tactile graphics depends deeply on blind people. So tactile graphics should be produced according to a blind person's experience with reading tactile graphics. (3) There are two major types of

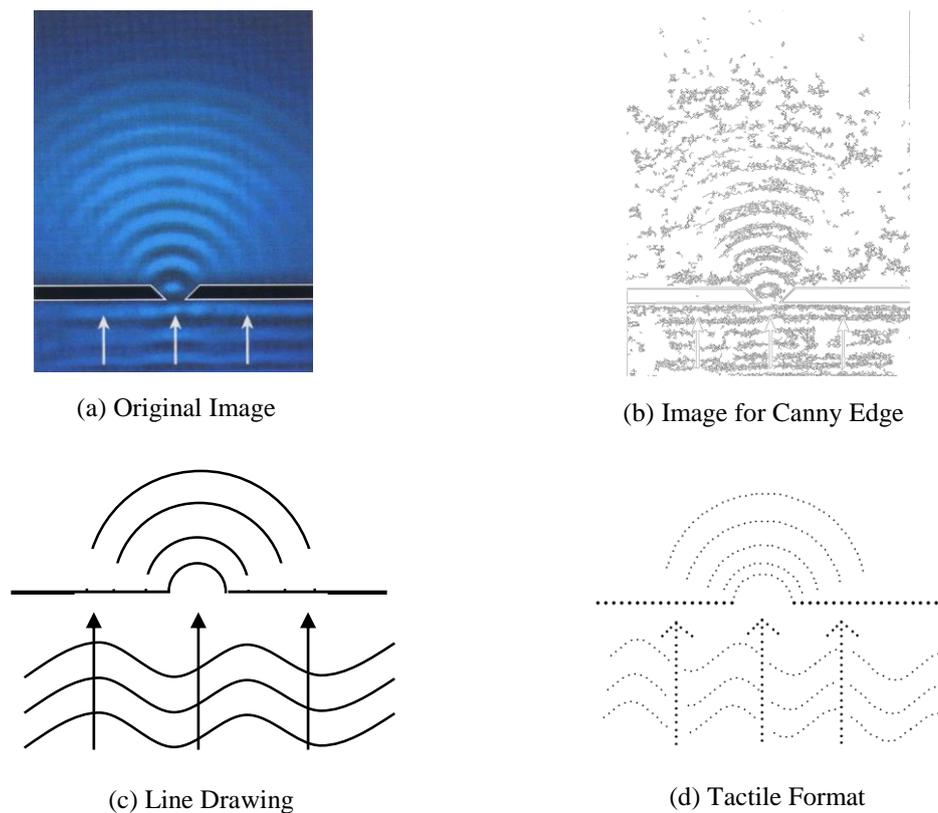


Figure 1.1: An Example for a Figure and Its Edge Image, Line Drawing and Tactile Format

digital images: bitmap images and vector graphics. Vector graphics allow the transformation of scaling and rotation without image distortion, but bitmap images do not allow such transformation. However, figures in physics textbooks are often captured as bitmap images. (4) Figures in physics textbooks are often color images or grayscale images. It is difficult to use image processing to detect object edges where image brightness changes sharply. This is because, in the captured image, there are often not serious contrast at the edge of objects. Figure 1.1 (b) shows an example of detected edges for Figure 1.1 (a), but it cannot be used for producing tactile graphics. Therefore, Figure 1.1 (a) should be first translated into a line drawing such as Figure 1.1 (c), and the line drawing finally should be translated into a tactile graphic such as Figure 1.1 (d). Due to the reasons above, figures in physics textbooks often need to be redrawn before producing tactile graphics.

There are a lot of software systems have been developed for creating figures. Most of

these drawing systems have a menu or a toolbar. Therefore, users can draw a figure by using a mouse and a keyboard to click the buttons on the toolbar in order to choose an object for drawing. Some tablet PC based software systems are also developed with a manual input interface, so users can draw a figure by hand in a natural way. Because most of people who produce tactile graphics, are not familiar with computer operation, they tend to produce tactile graphics using less computer operation. So hand-drawn method might be an effective way to draw a figure.

Based on the reasons above, we choose hand-drawn maps as a preliminary work of our study. Therefore, we aim to develop a system for automatic translation of hand-drawn maps into tactile graphics, and we contribute to do an experiment of usability evaluation in order to show whether hand-drawn method is an effective way for producing tactile graphics. Thus, the second research issue focuses mainly on the discussion of the usability issue, that is, which interface is good for users who are not familiar with computer operation: an interface with a mouse and a keyboard, and an interface with a hand-drawn input method.

1.3.2.3 Extraction of Text from Natural Scene Images

There are several studies related to text extraction from natural scene images. Most research use one of the following three techniques: (1) edge detection algorithms; (2) clustering algorithms for colored pixels [51 - 53]; and (3) image transforms such as Fourier transform and Discrete Cosine Transform (DCT) [54 - 56]. However, each of these methods has significant limitations. For example, character edges detected by the first method often connect to non-character edges; color clustering algorithm is difficult to determine the number of color regions, this is because there can be many different colors in a scene image. DCT is applied to detect character and non-character regions according to the cue features in the DCT coefficients, but for a complex image, it is not always true that the features in the DCT coefficients for character regions can be observed clearly than that for non-character regions [57].

Most methods used in past research first detect text regions in an image, and then segment the image into text and background. However, the performance of these methods

relies on the text detection algorithm and image complexity. Scene text is usually presented on signboards, and the signboard surface usually has uniform color. Therefore, the ideal way for extracting characters from signboard regions is if a given captured image can be segmented into homogeneous regions. In the third research issue, we contribute to extract characters from scene images, the main work is image segmentation.

1.4 Organization

This dissertation consists of 6 chapters, it is designed so that each chapter is self-contained and can be read on its own.

In Chapter 1, a review of assistive technologies for the visually impaired is first described and discussed, and then the research motivations and objectives of this dissertation are introduced. In Chapter 2, tactile graphics and their design principles are depicted and discussed. Chapter 3 discusses a method for automatically translating mathematical graphs into tactile graphics. In this chapter, we mainly contribute to the work of graph recognition. Chapter 4 proposes a system that is developed for automated translation of hand-drawn figures into tactile graphics. In this chapter, we mainly focus on the usability evaluation for hand-drawn method. In Chapter 5, a method for extracting text from scene images is proposed, especially the algorithm for image segmentation. Finally, we conclude this dissertation in Chapter 6, where future works are also postulated.

References

- [1] D. Pascolini and S. P. Mariotti, "Global Estimates of Visual Impairment: 2010", *British Journal Ophthalmology*, Vol.96, No.5, pp.614-618, 2012.
- [2] R. Manduchi and J. Coughlan, "(Computer) Vision without Sight", *Communications of the ACM*, Vol.55, No.1, pp.96-104, 2012.
- [3] http://en.wikipedia.org/wiki/Corrective_lens
- [4] B. B. Blasch, W. R. Wiener and R. L. Welsh, *Foundations of Orientation and Mobility*, AFB Press, Third Edition, 2010.
- [5] D. H. Yen, "Currently Available Electronic Travel Aids for the Blind", <http://www.noogenesis.com/eta/current.html>, 2005.
- [6] A. G. Dodds, D. C. Carter and C. I. Howarth, "The Sonic PathFinder: An Evaluation", *Journal of Visual Impairment and Blindness*, Vol.78, No.5, pp.203-206, 1984.
- [7] A. Heyes, "A Polaroid Ultrasonic Travel Aid for the Blind", *Journal of Visual Impairment and Blindness*, Vol.76, pp.199-201, 1982.
- [8] I. Ulrich and J. Borenstein, "The Guide Cane-Applying Mobile Robot Technologies to Assist the Visually Impaired", *IEEE Transaction on Systems, Man, and Cybernetics-Part A: Systems and Humans*, Vol.31, No.2, pp.131-136, 2001.
- [9] J. Barth and E. Foulhe, "Preview: A Neglected Variable in Orientation and Mobility", *Journal of Visual Impairment and Blindness*, Vol.73, No.2, pp.41-48, 1979.
- [10] S. Shoval, J. Borenstein and Y. Koren, "The NavBelt - A Computerized Travel Aid for the Blind Based on Mobile Robotics Technology", *IEEE Transactions on Biomedical Engineering*, Vol.45, No.11, pp.1376-1386, 1998.
- [11] L. Kim, S. Park, S. Lee and S. Ha, "An Electronic Traveler Aid for the Blind Using Multiple Range Sensors", *IEICE Electronics Express*, Vol.6, No.11, pp.794-799, 2009.
- [12] C. Gearhart, A. Herold, B. Self, C. Birdsong and L. Slivovsky, "Use of Ultrasonic Sensors in The Development of An Electronic Travel Aid", in *Proceedings of the IEEE International Symposium on Sensors Applications*, pp.275-280, 2009.

- [13] D. Yuan and R. Manduchi, "Dynamic Environment Exploration Using a Virtual White Cane", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol.1, pp.243-249, 2005.
- [14] V. Pradeep, G. Medioni and J. Weiland, "Piecewise Planar Modeling for Step Detection Using Stereo Vision", in *Proceedings of the Workshop on Computer Vision Applications for the Visually Impaired*, 2008.
- [15] V. Pradeep, G. Medioni and J. Weiland, "Robot Vision for the Visually Impaired", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp.15-22, 2010.
- [16] J. Saez, F. Escolano and A. Penalver, "First Steps towards Stereo-based 6DOF SLAM for the Visually Impaired", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, 2005.
- [17] J. Saez and F. Escolano, "Stereo-based Aerial Obstacle Detection for the Visually Impaired", in *Proceedings of the Workshop on Computer Vision Applications for the Visually Impaired*, 2008.
- [18] J. Wilson, B. N. Walker, J. Lindsay, C. Cambias and F. Dellaert, "SWAN: System for Wearable Audio Navigation", in *Proceedings of the IEEE International Symposium on Wearable Computers*, 2007.
- [19] P. Meijer, "An Experimental System for Auditory Image Representations", *IEEE Transactions on Biomedical Engineering*, Vol.39, No.2, pp.112-121, 1991.
- [20] G. Sainarayanan, *On Intelligent Image Processing Methodologies Applied to Navigation Assistance for Visually Impaired*, Ph. D. Thesis, University Malaysia Sabah, 2002.
- [21] G. Balakrishnan, G. Sainarayanan, R. Nagarajan and S. Yaacob, "Wearable Real-Time Stereo Vision for the Visually Impaired", *Engineering Letters*, Vol.14, No.2, 2007.
- [22] G. P. Fajarnes, L. Dunai, V. S. Praderas and I. Dunai, "CASBLiP- A New Cognitive Object Detection and Orientation System for Impaired People", in *Proceedings of the 4th International Conference on Cognitive Systems*, 2010.
- [23] A. Rodriguez, J. Javier Yebes, P. F. Alcantarilla, L. M. Bergasa, J. Almazan and A. Cela, "Assisting the Visually Impaired: Obstacle Detection and Warning System by Acoustic Feedback", *Sensors* 2012, 12, pp.17476-17496, 2012.

- [24] V. Kulyukin, C. Gharpure and J. Nicholson, "RFID in Robot-Assisted Indoor Navigation for the Visually Impaired", in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.1979-1984, 2004.
- [25] A. M. Ladd, K. E. Bekris, A. P. Rudys, D. S. Wallach and L. E. Kavraki, "On the Feasibility of Using Wireless Ethernet for Indoor Localization", *IEEE Transaction On Robotics and Automation*, Vol.20, No.3, pp.555-559, 2004.
- [26] W. Crandall, B. Bentzen, L. Myers and J. Brabyn, "New Orientation and Accessibility Option for Persons with Visual Impairment: Transportation Applications for Remote Infrared Audible Signage". *Clinical and Experimental Optometry*, Vol.84, Issue 3, pp.120-131, 2001.
- [27] M. A. Hersh and M. A. Johnson, *Assistive Technology for Visually Impaired and Blind People*, Publisher: Springer-Verlag London, 2008.
- [28] S. Landau, "Tactile Graphics and Strategies for Non-visual Seeing", *Thresholds*, Vol.19, pp.78-82, 1999.
- [29] N. Tanaka and M. Okudaira, "A Study on Image Processing for Action Assistance of Visually Impaired Humans Using a Camera Equipped Cellular Phone", *Technical Report of the Institute of Image Information and Television Engineers*, Vol.33, Issue 6, pp.173-176, 2009.
- [30] N. Ezaki, K. Kiyota, B. T. Minh, M. Bulacu and L. Schomaker, "Improved Text-Detection Methods for a Camera-based Text Reading System for Blind Persons", in *Proceedings of the 8th International Conference on Document Analysis and Recognition*, pp.257-261, 2005.
- [31] S. M. Hanif and L. Prevost, "Texture Based Text Detection in Natural Scene Images: A Help to Blind and Visually Impaired Persons", in *Proceedings of Conference & Workshop on Assistive Technologies for People with Vision & Hearing Impairments Assistive Technology for All Ages*, M.A. Hersh (Ed.), 2007.
- [32] M. Pazio, M. Niedzwiecki, R. Kowalik and J. Lebiecz, "Text Detection System for The Blind", in *Proceedings of the 15th European Signal Processing Conference*, pp.272-276, 2007.
- [33] U. Kalai selvi and J. Anish Kumar, "Camera based Assistive Text Reading System using Gradient and Stroke Orientation for Blind Person", *International Journal of Latest Trends in Engineering and Technology*, Vol.4, Issue 1, pp.325-330, 2014.
- [34] O. Foong and N. Razali, "Signage Recognition Framework for Visually Impaired

People”, in *Proceedings of International Conference on Computer Communication and Management*, pp.488-492, 2011.

- [35] C. Yi, Y. Tian and A. Arditi, “Portable Camera-based Assistive Text and Product Label Reading from Hand-held Objects for Blind Persons”, *IEEE/ASME Transactions on Mechatronics*, 2014.
- [36] J. C. Bliss, “Reading Machines for The Blind”, *Active Touch - The Mechanism of Recognition of Objects by Manipulation: A Multidisciplinary Approach*, G. Gordon (Ed.), Pergamon Press, Oxford, UK, 1978.
- [37] A. J. Hardwick, “Rendering of Moon Text on Simulated Tactile Diagrams for Blind Computer Users by Force-feedback”, *Touch, Blindness and Neuroscience*, S. Ballesteros Jiménez and M. A. Heller (Eds.), UNED Press, Madrid, Spain, pp.351-358, 2004.
- [38] <http://www.daisy.org/>
- [39] P. Blenkhorn and D. G. Evans, “The Architecture of a Windows Screen Reader”, *Assistive Technology-Added Value to the Quality of Life (AAATE 2001)*, C. Marineek, C. Buhler, H. Knops, and R. Andrich (Eds.), IOS Press, Amsterdam, pp.119-123, 2001.
- [40] P. Blenkhorn, D. G. Evans and A. Baude, “Full Screen Magnification Using Direct X Overlays”, *IEEE Transaction on Neural Systems and Rehabilitation Engineering*, Vol.10, No.4, pp.225-232, 2002.
- [41] D. G. Evans and P. Blenkhorn, “Architectures of Assistive Software Applications for Windows-based Computers”, *Journal of Network and Computer Applications*, Vol.26, Issue 2, pp.213-228, 2003.
- [42] H. P. Lee, J. Huang, C. Chen, and T. Sheu, “Building a Color Recognizer System on the Smart Mobile Device for the Visually Impaired People”, in *Proceedings of the 6th International Multi-Conference on Computing in the Global Information Technology*, pp.95-98, 2011.
- [43] S. E. Krufka and K. E. Barner, “Automatic Production of Tactile Graphics from Scalable Vector Graphics”, in *Proceedings of The 7th International ACM SIGACCESS Conference on Computers and Accessibility*, pp.166-172, 2005.
- [44] R. E. Ladner, M. Y. Ivory, R. Rao, S. Burgstahler, D. Comden, S. Hahn, M. Renzelmann, S. Krisnandi, M. Ramasamy, B. Slabosky, A. Martin, A. Licenski, S. Olsen and D. Groce “Automating Tactile Graphics Translation”, in *Proceedings of the 7th International ACM SIGACCESS Conference on Computer and Accessibility*,

pp.150-157, 2005.

- [45] C. Jayant, M. Renzelmann, D. Wen, S. Krisnandi, R. Ladner and D. Comden, "Automated Tactile Graphics Translation: In the Field", in *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pp.75-82, 2007.
- [46] T. Winlock, E. Christiansen and S. Belongie, "Toward Real-time Grocery Detection for the Visually Impaired", in *Proceedings of the Workshop on Computer Vision Applications for the Visually Impaired*, 2010.
- [47] J. Sudol, O. Dialameh, C. Blanchard and T. Dorsey, "Looktel: A Comprehensive Platform for Computer-aided Visual Assistance", in *Proceedings of Workshop on Computer Vision Applications for the Visually Impaired*, 2010.
- [48] S. Oouchi, M. Sawada, T. Kaneko and K. Chida, "A Survey on Making and Using Tactile Educational Materials in Schools for the Blind", *Bulletin of National Institute of Special Needs Education*, Vol.31, pp.113-125, 2004.
- [49] T. Fuda, S. Omachi and H. Aso, "Recognition of Line Graph Images in Documents by Tracing Connected Components", *IEICE Transactions on Information and Systems*, Vol.J86-D- II, No.6, pp.825-835, 2003.
- [50] K. Shingai and R. Fukuda, "Recognition of the Line Images in Science Documents", *Technical Report of IEICE*, PRMU, Vol.101, No.713, pp.69-74, 2002.
- [51] Y. Matsuda, S. Omachi and H. Aso, "String Detection from Scene Images by Binarization and Edge Detection", *IEICE*, D, Vol.J93-D, No.3, pp.336-344, 2010.
- [52] H. Hase, M. Yoneda, M. Sakai and H. Maruyama, "Consideration of Color Segmentation to Extract Character Areas from Color Document Images", *IEICE*, D-II, Vol.J83-D-II, No.5, pp.1294-1304, 2000.
- [53] K. Ashida, H. Nagai, M. Okamoto, H. Miyao and H. Yamamoto, "Extraction of Characters from Scene Images", *IEICE*, D-II, Vol.J88-D-II, No.9, pp.1817-1824, 2005.
- [54] Y. Liu, T. Yamamura, N. Ohnishi and N. Sugie, "Extraction of Character String Regions from a Scene Image", *IEICE*, D-II, Vol.J81-D-II, No.4, pp.641-650, 1998.
- [55] S. Saitoh, H. Goto and H. Kobayashi, "Analysis and Comparison of Frequency Features for Scene Text Detection", *Technical Report of IEICE*, PRMU2004-128, pp.31-36, 2004.

- [56]D. Crandall, S. Antani and R. Kasturi, “Extraction of Special Effects Caption Text Events from Digital Video”, *International Journal on Document Analysis and Recognition*, Springer-Verlag, Vol.5, pp.138-157, 2003.
- [57]M. Inami, *A Character String Extraction Method from Scene Images Using Edge Detection and Color Clustering*, Master Thesis, Toyama Prefectural University, Japan, 2012.

Chapter 2

Tactile Graphics

In this chapter, the definition and the necessity for tactile graphics are first described. Then, the types and forms of tactile graphics are reviewed. Lastly, the principles for designing tactile graphics are summarized.

2.1 Introduction

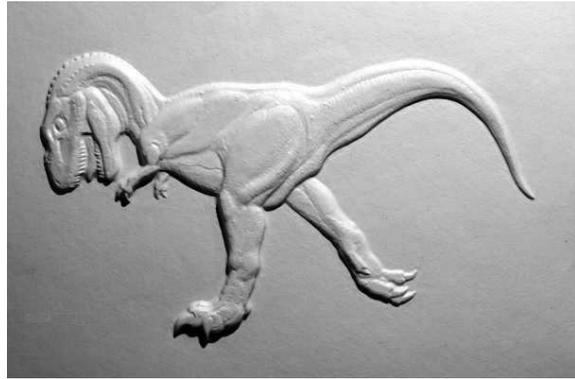
Our society is so geared toward visual impressions. For human beings, they obtain information mainly through eyes. There are more and more descriptive and instructive materials which are presented as pictorial form, with or without additional text words. However, most of these pictorials are in visual form, they cannot be utilized by many visually impaired people.

Images can be accepted by visually impaired people in various ways, for example, verbal descriptions, phonetic explaining, embossed patterns and other. One of the most common methods for the visually impaired is tactile graphics. This is because tactile graphics are designed to be represented by a raised surface, so visually impaired people can touch them with their fingertips. Therefore, the use of tactile graphics is to convey non-textual information. Figure 2.1 shows some examples of tactile graphics that are used to help people with visual impairment.

Because of the accessibility for tactile graphics, in daily life, tactile graphics are widely used by blind people and partially-sighted people. In the following section, main types and forms of existing tactile graphics are introduced.



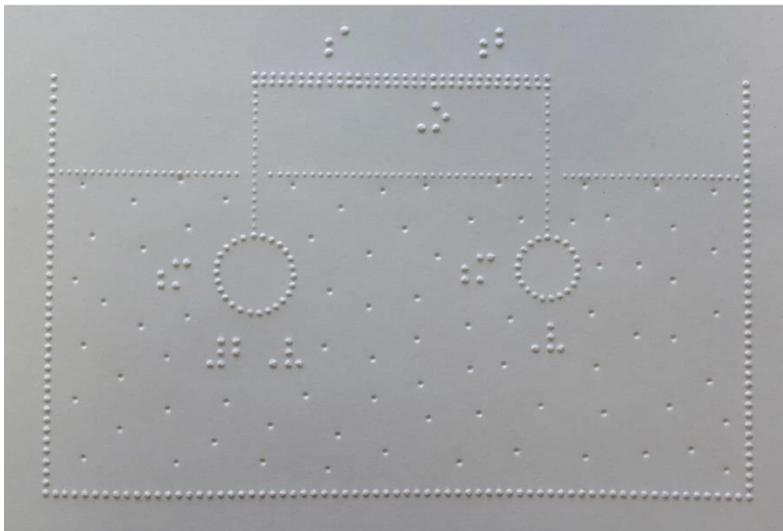
(a) Craft Graphic



(b) Thermoformed Graphic



(c) Swell-form Graphic



(d) Braille-form Graphic

Figure 2.1: Examples for Tactile Graphics

2.2 Types and Forms of Tactile Graphics

(1) Pasted Tactile Graphics

This tactile graphic is produced by simply attaching objects (e.g., strings, cloths and other materials) to a substrate in order to represent items and symbols. However, this is a time consuming work, as well as an expensive method, because it needs to manually paste all materials onto each substrate [1]. Figure 2.1 (a) shows an example for craft graphic.

(2) Thermoformed Tactile Graphics

Thermoform (also called vacuum forming) is one of the most common methods for producing tactile graphics. Thermoformed graphics are created through a process as follows: a sheet of plastic is first put onto a model; the plastic is then heated while the air between the sheet and the model is removed; finally a tactile graphic is formed [1]. As shown in Figure 2.1, (b) is a thermoformed graphic. However, this process is time consuming, because it needs to create a model before heating and forming a tactile graphic.

(3) Tactile Graphics using Swell/Capsule Paper

Swell paper (often referred to capsule paper, microcapsule paper) is a special paper that has a special coating of heat reactive chemicals. This process for producing tactile graphics is as follows: 1) printing a graphic onto a swell paper using a standard printer or copy machine, where black ink provides control over the raised surface area; 2) running the paper through a thermoform machine, and the heat reacts with the black ink and causes it to swell, the rest of the paper and other colors will remain flat [2]. Figure 2.2 shows an example of machine used for heating sell paper, and Figure 2.1 (c) is a swell-from graphic. It is a simple, fast and low cost method for creating tactile graphics by using swell papers.

(4) Braille Graphics

Braille embossers also can be used to produce tactile graphics. Figure 2.3 shows an example of Braille embosser, and Figure 2.1 (d) is a Braille graphic. Braille embosser is a special printer controlled by computer, it embosses a graphic onto paper through the use of solenoids that control embossing pins. But, these embossers are often expensive.



Figure 2.2: Swell-Form Graphics Machine for Producing Tactile Graphics



Figure 2.3: Braille Embosser for Producing Tactile Graphics

2.3 Guidelines for the Design of Tactile Graphics

Tactile graphics are not merely raised pictures, they are often used to interpret non-textual information. We have to take into account some design principles, [3 - 10], while producing tactile graphics. If users can use tactile graphics well, they will be free to take individual actions more freely and safely.

2.3.1 Characteristics for Tactile Graphics

- (1) Tactile graphics can be read and understood easily.
- (2) Tactile graphics can accurately interpret what is to be conveyed.
- (3) Tactile graphics can be made individually according to user's experience with reading tactile graphics.
- (4) Graphics are presented in high tactual quality.

2.3.2 Basic Principles for the Design of Tactile Graphics

- (1) Tactile graphics should be designed as simple as possible, that is, a tactile graphic must include the minimum amount of information. Therefore, some parts of a diagram or sections of a map should be omitted if they does not convey essential contents, only relevant and referenced information is contained.
- (2) Tactile graphics should be made as tactually clear as possible, this means that the information represented in tactile graphics should be easy to read and understand. Therefore, they should use various height levels of tactile elevations and keep differences in the width of the lines.
- (3) It becomes hard to distinguish different symbols of objects if they are so similar or too close together. Therefore, space is required to keep the contrast in texture and lines. The space between symbols and lines should be larger than 1/4 inch, and the space between shapes with sides should be larger than 1/2 inch.
- (4) Because the ability for reading tactile graphics depends deeply on the user's age

level, knowledge level, experience with reading tactile graphics and more. The size and types of tactile graphics should be different for different readers, as well as appropriate language. Braille or embossing words can be used to instead of a key symbol or part of a graphic.

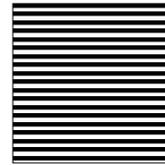
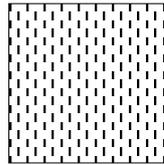
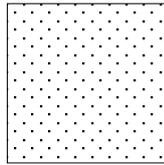
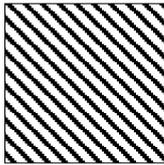
- (5) In order to improve the accessibility for tactile graphics, a complicated graphic should be represented by separate graphics showing partial information, or graphics for general and detailed information.
- (6) Tactile graphics should be created in 2 dimensions.

2.3.3 Tactile Graphic Symbols

For a tactile graphic, it needs to use different symbols to represent different components and objects. Due to the user's skill level, it is necessary to limit the number of key symbols for drawing points, lines, and areas. A tactile graphic is commonly composed of four types of symbols: areas (texture), lines, point symbols and Braille labels.

- (1) Areas are used to represent regions of extent, for example, water or continents in a map. They have the following characteristics: variations in height; differences in the density or texture of the patterns; a line or white space between areas. Figure 2.4 (a) shows some examples for texture.
- (2) Lines are used to represent linear information, for example, rivers, routes, and outlines. The types of lines include solid lines, dashed and dotted lines. Figure 2.4 (b) shows some examples for line.
- (3) Point symbols are used to represent specific locations, for example, a city, point in a line graph. Figure 2.4 (c) shows some examples for point symbols.
- (4) Braille labels are used to explain and define all graphic symbols, either may be used to represent an area or point. They are often placed in the most appropriate locations: in the symbol; next to the symbol; or near the symbol and connected by lead line.

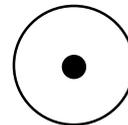
An arrow is a linear symbol with direction, so it is often used as lead line, Figure 2.4 (d) shows some examples of arrow symbols. An arrow is commonly used to indicate a movement, process, or direction. The space between symbols for lines, points, and Braille must be at least 1/8 inch.



(a) Examples for Texture Symbols



(b) Examples for Line Symbols



(c) Examples for Points Symbols



(d) Examples for Arrows Symbols

Figure 2.4: Examples for Symbols Used in Tactile Graphics

References

- [1] M. A. Hersh and M. A. Johnson, *Assistive Technology for Visually Impaired and Blind People*, Publisher: Springer-Verlag London, pp.143-145, 2008.
- [2] P. K. Edman, *Tactile graphics*, American Foundation for the Blind, New York, 1992.
- [3] M. G. Kwok, "Guideline for Tactile Figures and Maps", in *Proceedings of Guidelines on Tactile and Haptic Interactions*, pp.43-47, 2005.
- [4] Braille Authority of North America, Guidelines and Standards for Tactile Graphics, 2010, Web Version, Retrieved from <http://www.brailleauthority.org/tg/web-manual/index.html>, 2011.
- [5] American Printing House for the Blind, Guidelines for Design of Tactile Graphics, <http://www.aph.org/>, 1997.
- [6] Rocky Mountain Braille Associates, Design Principles for Tactile Graphics, <http://www.tactilegraphics.org/>
- [7] American Printing House for the Blind, Guide to Designing Tactile Illustrations for Children's Books, <http://www.aph.org>, 2008.
- [8] T. Kaneko and S. Oouchi, "Practical Guidelines for Converting to Braille Dots Drawings and Characteristics of Tactile Drawings", *FY2002 research report of Department of Visual Impairment Education: Study on Tactile- and Auditory-Perceptible Teaching Materials Creation Systems for Individual Child with Visual Disability*, NISE, pp.6-15, 2003.
- [9] T. Kaneko and S. Oouchi, "Tactile Graphics in Braille Textbooks: The Development of a Tactile Graphics Creation Manual", *NISE Bulletin*, Vol.10, pp.13-28, 2010.
- [10] H. Jurgensen and C. Power, "Information Access for the Blind - Graphics, Modes, Interaction", in *Proceedings of Guidelines On Tactile and Haptic Interactions*, pp.13-25, 2005.

Chapter 3

Automatic Translation of Mathematical Graphs into Tactile Graphics

This chapter discusses a method for automated translation of mathematical graphs into tactile graphics. First of all, some related research are reviewed. Second, a brief description for a method is described, i.e. separating a mathematical graph into solid line elements, broken line elements and character elements. Third, an algorithm for extraction of broken line graph elements is introduced. Fourth, a method for extraction of solid line graph elements is discussed. Lastly, the results of our computer experiments are discussed.

3.1 Introduction

For sighted students, there are all kinds of reading materials. However, for visually impaired students, reading materials are mainly limited to textbooks. In Japan, there are 70 schools for the visually impaired students. But in 80% of these schools, there is no department for producing tactile teaching materials [1]. Most of the work for producing tactile materials is done by teachers, and some work is done by volunteers. This is because making tactile teaching materials need advanced knowledge of its corresponding field. For example, to make a tactile graphic for teaching mathematics, professional mathematical knowledge is needed. Until now, most tactile graphics are produced using less intelligent computer-aided systems. Therefore, a better computer-aided system for automating translation of printed books into tactile graphics is needed. As shown in Figure 3.1, the textbooks of mathematics and science often include texts, mathematical

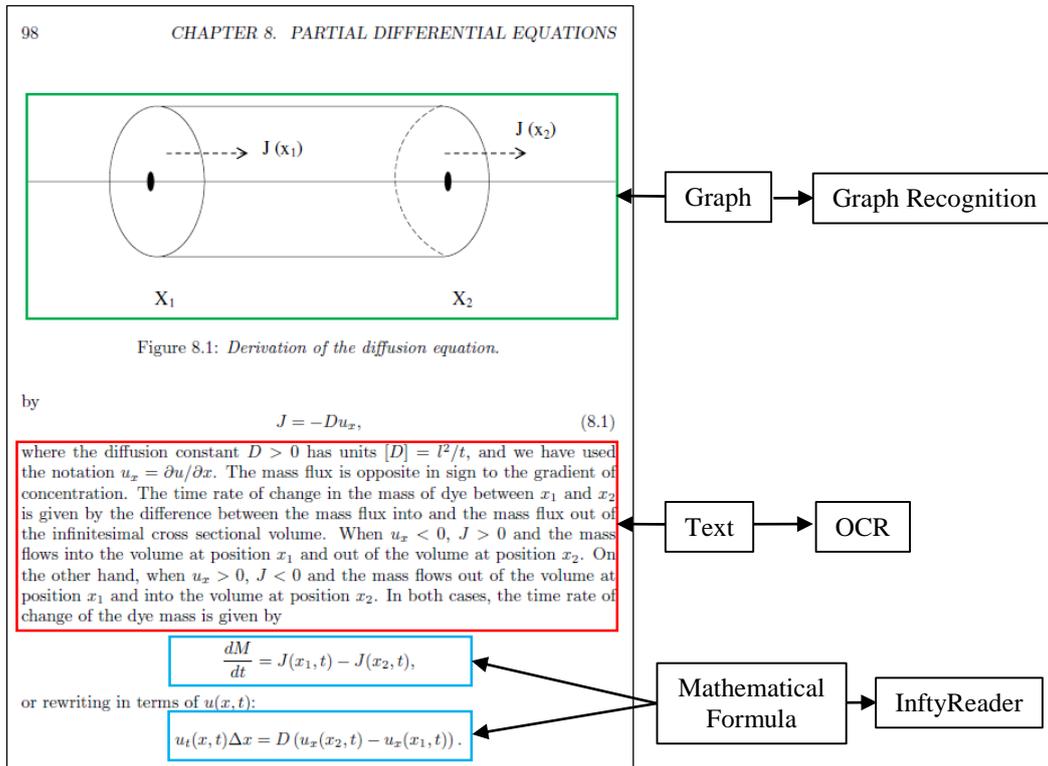


Figure 3.1: A Page of Mathematics Book

formulas/expressions and graphs, etc.

To develop such a system for automatic translation of printed (must be scanned into digital format) or digital documents into tactile format, segmentation for document layout is required. This work had been done by Hirayama [2], Lee [3], and Ishitani [4]. If the document has been segmented into text regions, mathematical expression regions and figure/graph regions, then the software of OCR and InftyReader can be applied to recognize the texts and expressions respectively, finally the texts and expressions are converted into Braille format.

The software of InftyReader is developed by InftyProject [5] which is a volunteer organization for helping people with visual impairment in scientific fields. One of the purposes of InftyProject is to digitize scientific documents such as mathematics journals and books, and it provides their tactile materials to the visually impaired. InftyReader can translate printed mathematical expressions into digital expressions such as LaTeX and MathML. However, because there is no intelligent system to translate printed

mathematical graphs into SVG (Scalable Vector Graphics) [6] images, the efficiency of work related to the translations is disturbed. Note that the DAISY (Digital Accessible Information System) consortium [7], which develops, maintains, and promotes the international standard of digital books for people with visual impairment, recently adopted SVG as the standard for digital figures.

The work, reported by Krufka et al. [8], is done on automatically producing tactile graphics from scalable vector graphics. However, many images are not yet in SVG format. The Tactile Graphics Assistant (TGA) is a software to assist the translation of bitmap images into tactile representations [9]. However, human intervention and validation is necessary between each step of the workflow of TGA [10]. Therefore, a technology is needed for automated translation of figures in mathematics and science textbooks into tactile graphics.

Based on the reasons above, the goal of this chapter is to develop a system to automatically translate mathematical graphs into tactile graphics. Therefore, techniques for mathematical graph recognition are needed in our system. So far, many graph recognition methods have been developed. Fuda et al. [11] studied a graph recognition method requiring that graphs must satisfy many assumptions; a graph, for example, has to be drawn inside a rectangular area that is specified by the x -axis and the y -axis. The graph recognition methods introduced by the literatures [12], [13], and [14] must also satisfy assumptions about graphs. However, many mathematical graphs do not satisfy all of the aforementioned assumptions.

To facilitate the production of tactile graphics, we are developing a computer-aided system [15] for automating translation of printed mathematical graphs into SVG images. The following descriptions summarize the characteristics of the mathematical graphs we focus on.

1. Characters and mathematical formulas may be distributed in and around the graph.
2. A character string or a mathematical formula may not lie on the correct orientation (i.e., on the horizontal orientation).
3. Graphs may contain several types of broken lines.

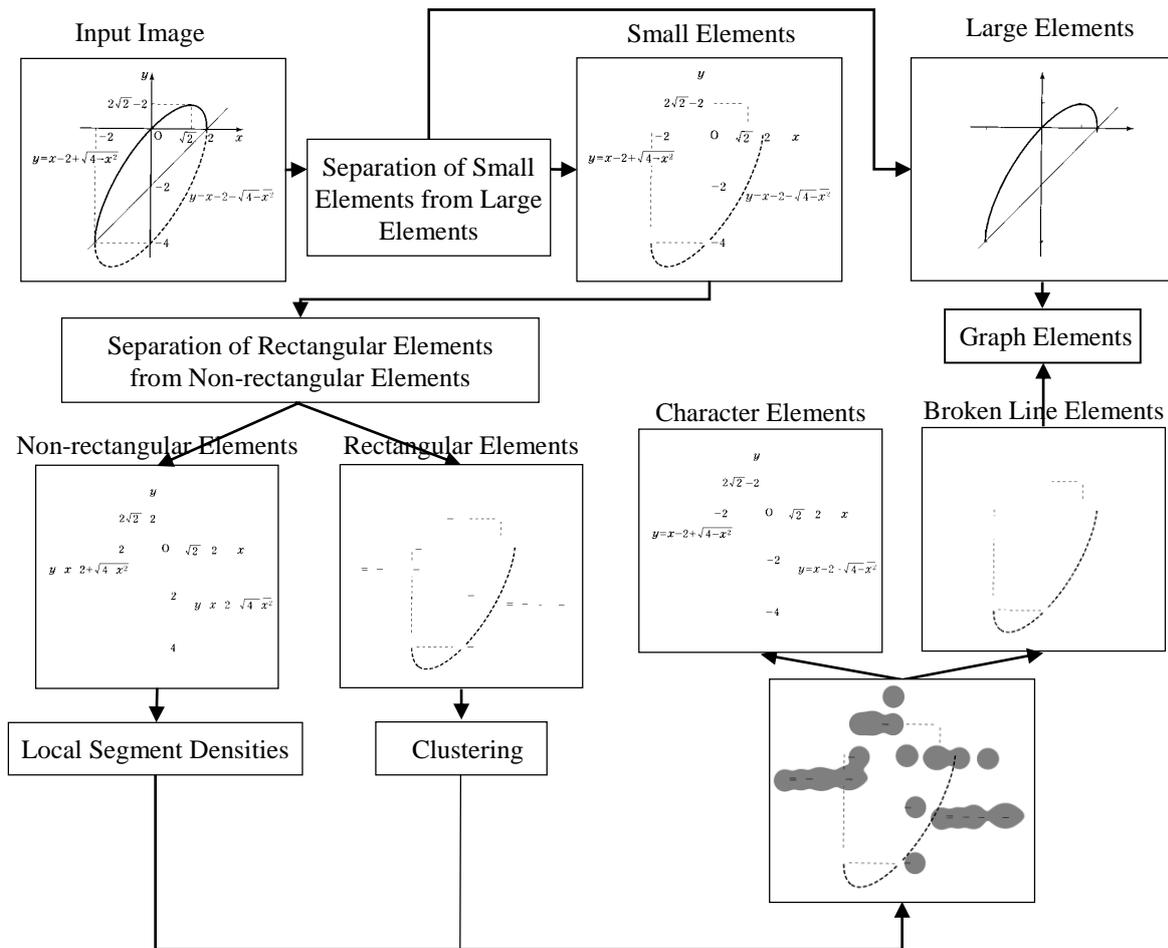


Figure 3.2: Outline of Separation

A mathematical graph includes the following three elements: (1) solid line graph elements, (2) broken line graph elements, and (3) elements from character strings and mathematical formulas. This chapter focuses on mathematical graph recognition.

3.2 Separation of Mathematical Graphs

We have proposed a method [15] to separate a bitmap image of a mathematical graph into three parts: solid line elements, broken line elements, and character elements. Figure 3.2 shows the outline of the separation method. A clustering method is applied to the set of small elements, and as the result finds clusters so that each of them includes only broken line elements. The characteristics of these clusters are shown below.

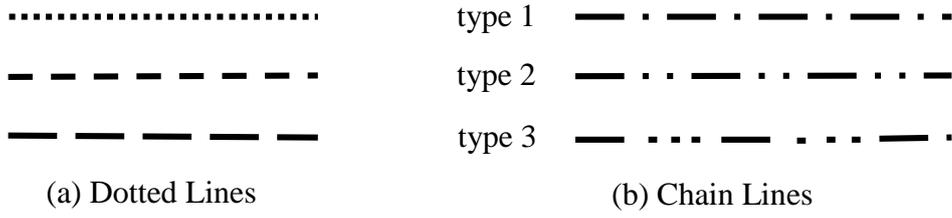


Figure 3.3: Broken Lines

1. For each of the clusters, G , almost all elements in G are elements of the same broken line.
2. The elements of a single broken line are often divided into different clusters.

So, the problem is how to find a cluster which includes all elements of a single broken line. We will discuss this issue in the following sections.

3.3 Extraction of Broken Line Graph Elements

Figure 3.3 shows examples of broken lines which we focus on. There are dotted lines and chain lines. Chain lines are further broken down into types 1, 2, and 3. In this section, we describe methods that classify a cluster from the previous section into a dotted line or a chain line.

A dotted line consists of homogeneous rectangular elements, while a chain line is composed by two different kinds of rectangular elements: short elements and long elements. Therefore, elements of a dotted line can be classified into only one group, homogeneous rectangular elements. Similarly, elements of a chain line are classified into two different groups: one is for short elements, and the other is for long elements. If we can evaluate the number of homogeneous groups in a cluster, it enables us to classify the cluster into a dotted line or a chain line. To evaluate the optimal number of homogeneous groups (i.e., clusters), we measure by two cluster validities, $V_{DB}(\cdot)$ and $V_D(\cdot)$ [16], whose definitions are given below.

Given a set of $k(\geq 2)$ clusters, $\Gamma = \{G_1, \dots, G_k\}$,

$$V_{DB}(\Gamma) = \left(\frac{1}{k} \right) \sum_{i=1}^k \left[\max_{j(j \neq i)} \left\{ \frac{\alpha_i + \alpha_j}{\|\bar{v}_i - \bar{v}_j\|} \right\} \right] \quad (3.1)$$

where for $i = 1, \dots, k$, $\bar{v}_i = \sum_{x \in G_i} \frac{x}{|G_i|}$ and $\alpha_i = \sum_{x \in G_i} \frac{\|x - \bar{v}_i\|}{|G_i|}$

$$V_D(\Gamma) = \min_{1 \leq i \leq k} \left[\min_{1 \leq j \leq k (j \neq i)} \left\{ \frac{\hat{\delta}(G_i, G_j)}{\max_{1 \leq t \leq k} \{\Delta(G_t)\}} \right\} \right] \quad (3.2)$$

Here, for any clusters S and T , $\Delta(S) = \max_{x, y \in S} \{\delta(x, y)\}$ and $\hat{\delta}(S, T) = \min_{x \in S, y \in T} \{\delta(x, y)\}$, and $\delta(x, y)$ is the distance between x and y .

Note that the larger the value of $V_{DB}(\cdot)$, the better the clustering result. Similarly, the smaller the value of $V_D(\cdot)$, the better the clustering result. These two cluster validities are not defined when $k = 1$, therefore, we introduce a fuzzy inference system to avoid this disadvantage.

3.3.1 Dotted Line Classification

In this subsection, we describe the dotted line classification method. The following description is the procedure for the dotted line classification method.

Input: A cluster, G , of broken line elements.

Output: If G is a dotted line, return Yes, otherwise return No.

Step 1: Single-linkage clustering is applied to cluster G , and let Γ_k be the result when the number of clusters is k ($k = 1, 2, 3, 4, 5$). Calculate $V_{DB}(\Gamma_k)$ and $V_D(\Gamma_k)$ for every k .

Step 2: Apply a fuzzy inference system to $V_{DB}(\Gamma_k)$ and $V_D(\Gamma_k)$. If G is classified as a dotted line by the fuzzy inference system, then return Yes, otherwise return No.

In the single-linkage clustering of Step 1, every element, e , in G is represented by two characteristics: the number of the pixels of e , and the length of the long side of e .

The calculation scheme of the fuzzy inference system is based on Mamdani's fuzzy inference method [17], but the minimum operator is exchanged with the product operator. The fuzzy inference system has four arguments, x_1 , x_2 , x_3 , and x_4 , which are defined as follows. Let $G = \{e_1, \dots, e_n\}$, a set of broken line elements, and let Γ_k be a set of clusters of set G ; and then, $x_1 = \max_{2 \leq k \leq 5} V_{DB}(\Gamma_k)$, $x_2 = \min_{2 \leq k \leq 5} V_D(\Gamma_k)$, $x_3 = \frac{\min_{1 \leq i \leq n} p(e_i)}{\max_{1 \leq i \leq n} p(e_i)}$, and $x_4 = \frac{\min_{1 \leq i \leq n} \ell(e_i)}{\max_{1 \leq i \leq n} \ell(e_i)}$, where $p(e)$ is the number of the pixels of element e and $\ell(e)$ is the length of the long side of element e .

Fuzzy if-then rules are given below, and the membership functions are shown in Figure 3.4.

- Rule 1:** If x_1 is large, x_3 is large, and x_4 is large, then G is probably a dotted line.
- Rule 2:** If x_2 is small, x_3 is large, and x_4 is large, then G is probably a dotted line.
- Rule 3:** If x_1 is small, then G is probably not a dotted line.
- Rule 4:** If x_2 is large, then G is probably not a dotted line.
- Rule 5:** If x_3 is small, then G is probably not a dotted line.
- Rule 6:** If x_4 is small, then G is probably not a dotted line.

A cluster is classified as a dotted line if the output value of the fuzzy inference systems is more than or equal to 0.5.

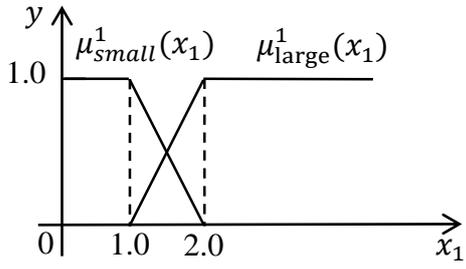
3.3.2 Chain Line Classification

Next, we will discuss a chain line classification method, which distinguishes clusters of elements of chain lines. If a cluster is classified as a chain line, the chain line classification method also gives its type. The following description is the procedure.

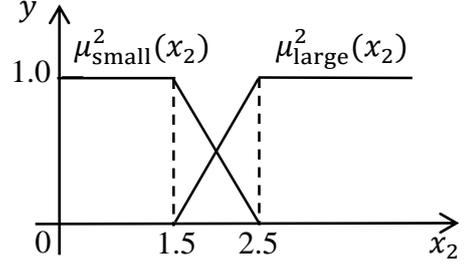
Input: A cluster, G , of broken line elements.

Output: If G is classified as a chain line, return the type of G , otherwise return No.

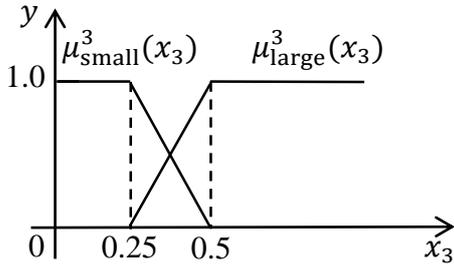
Step 1: Apply single-linkage clustering to G by setting the number of clusters to 2, and then divide G into two groups. Assign label 'a' to elements of one group, and label 'b' to elements of the other group. We then have a sequence of labels for G .



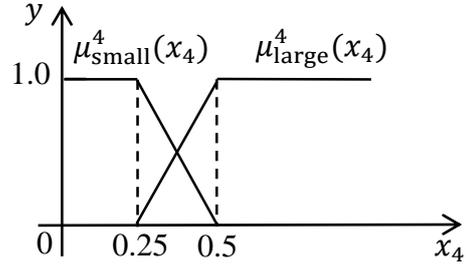
(a) Membership Functions for x_1



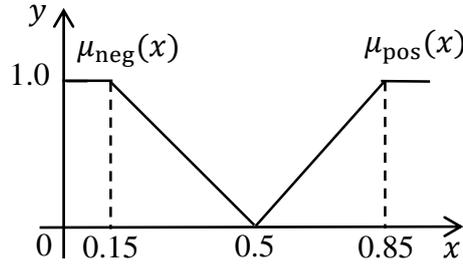
(b) Membership Functions for x_2



(c) Membership Functions for x_3



(d) Membership Functions for x_4



(e) Membership Functions for Consequence

Figure 3.4: Membership Functions

Step 2: Calculate similarity, $S_p(G)$ ($p = 1, 2, 3$), between the sequence and the template of a type p chain line. Here, the template of type 1 chain lines is $ababab \dots$. Similarly, those of types 2 and 3 are $abbabbabb \dots$ and $abbbabbbb \dots$, respectively.

Step 3: If similarity $S_p(G)$ is equal to the number of elements of G , then classify G as a type p chain line and return p , otherwise return No.

For the sequence obtained by Step 1 and the template of a type p chain line,

similarity $S_p(G)$ is defined as the largest number of successively matching labels.

3.3.3 Broken Line Classification

Lastly, the procedure for the broken line classification method is described below.

Input: A set of clusters, $\Gamma = \{G_1, \dots, G_t\}$, from Section 3.2.

Output: A set of dotted line clusters, Δ , sets of type p chain line clusters, X_p ($p = 1, 2, 3$), and a set of clusters, Φ .

Step 1: Set $\Delta \leftarrow \emptyset$, $X_p \leftarrow \emptyset$ ($p = 1, 2, 3$), and $\Phi \leftarrow \emptyset$.

Step 2: If Γ is empty, output Δ , X_p ($p = 1, 2, 3$), and Φ , and stop the procedure.

Step 3: Select G_i from Γ , and set $\Gamma \leftarrow \Gamma - \{G_i\}$. If G_i includes only one element, set $\Phi \leftarrow \Phi \cup \{G_i\}$ and go to Step 2.

Step 4: Apply the dotted line classification method to G_i . If G_i is classified as a dotted line, set $\Delta \leftarrow \Delta \cup \{G_i\}$ and go to Step 2.

Step 5: Apply the chain line classification method to G_i . If G_i is classified as a type p chain line, set $X_p \leftarrow X_p \cup \{G_i\}$ and go to Step 2.

Step 6: Divide set G_i into three groups, G'_i , G''_i , and G'''_i , in the following way. Suppose $S_p(G_i)$ is the greatest among the three similarities, $S_1(G_i)$, $S_2(G_i)$, and $S_3(G_i)$. Then, separate the elements of the sequence for G_i which gives similarity $S_p(G_i)$, and let G'_i denote the set of these elements. Furthermore, let G''_i and G'''_i denote the set of elements in G_i which are located on the left and the right side of G'_i , respectively.

Step 7: G'_i is classified as a type p chain line. Add G'_i to set X_p , and also add G''_i and G'''_i to set Γ . Go to Step 2.

3.3.4 Merging Clusters

Since the broken line classification method divides a cluster into several groups until every cluster is classified into one of the four types of broken lines, we need a merging process that combines clusters consisting of elements from the same broken line into a single cluster. If two clusters, G_1 and G_2 , satisfy the following two geometric characteristics, it is plausible that G_1 and G_2 are merged into a single cluster: (1) the

gradient for the broken lines corresponding to G_1 is almost equal to the gradient for the broken line corresponding to G_2 , and the two broken lines are located closely to each other, and (2) broken lines G_1 and G_2 are separated by some obstacles such as character strings. So our merging process is based on the following two criteria:

1. Similar geometric description.
2. Existence of interference.

Our merging process thus consists of two different merging processes.

3.3.4.1 Disposition Criterion

The following description explains the workflow of the merging process based on the first criterion.

Input: A set, Γ , of clusters from subsection 3.3.3.

Output: A set of clusters after merging.

Step 1: Select a pair of clusters, (G_i, G_j) , from Γ which is not tested yet. If there exists no such pair, output Γ and stop this procedure.

Step 2: If the types of broken lines G_i and G_j are different, go to Step 1.

Step 3: Apply pair (G_i, G_j) to a fuzzy inference system. If the result from the fuzzy inference system is positive (i.e., G_i and G_j are part of the same broken line), then merge G_i and G_j into a single cluster and go to the next step, otherwise go to Step 1.

Step 4: Let G be the cluster from the previous step. Update Γ by setting $\Gamma \leftarrow (\Gamma - \{G_i, G_j\}) \cup \{G\}$, and go to Step 1.

Let G_1 and G_2 be clusters corresponding to dotted lines. If these two clusters satisfy the following four geometric characteristics, then it is plausible that these two clusters are part of the same broken line.

1. The number of pixels of any element in G_1 is almost equal to the number of pixels of any element in G_2 .
2. The length of the long side of any element in G_1 is almost equal to the length of the long side of any element in G_2 .

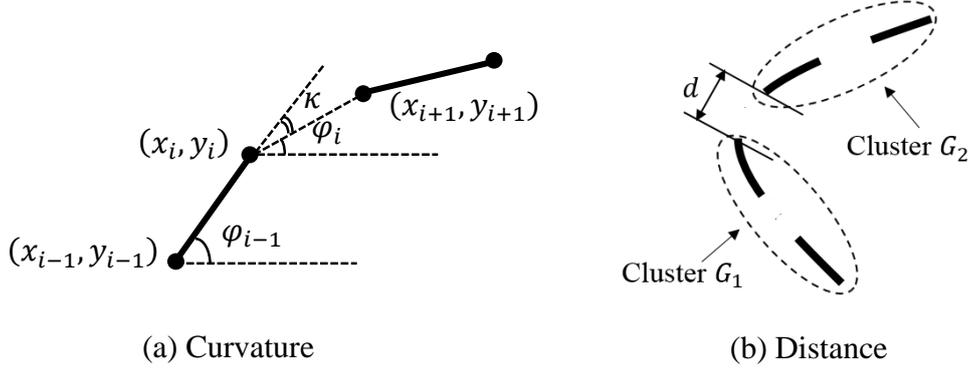


Figure 3.5: Curvature and Distance between Two Clusters

3. The distance between G_1 and G_2 is short.
4. The curvature at the intersection determined by lengthening dotted lines G_1 and G_2 is small.

The curvature for a digital curve at a point is calculated by the method introduced by literature [18]; that is, the curvature, κ , at point (x_i, y_i) in Figure 3.5 (a) is defined as $\kappa = |\varphi_i - \varphi_{i-1}|$. Fuzzy inference is applied to evaluate degree, which represents how true it is that two clusters G_1 and G_2 are part of the same broken line. The fuzzy inference system has four arguments, x_1 , x_2 , x_3 , and x_4 ; and they have the following measurements concerning the four geometric characteristics for dotted lines.

1.
$$x_1 = \left| \frac{1}{|G_1|} \sum_{e \in G_1} p(e) - \frac{1}{|G_2|} \sum_{e \in G_2} p(e) \right| \quad (3.3)$$

where $p(e)$ is the number of the pixels of element e .

2.
$$x_2 = \left| \frac{1}{|G_1|} \sum_{e \in G_1} \ell(e) - \frac{1}{|G_2|} \sum_{e \in G_2} \ell(e) \right| \quad (3.4)$$

where $\ell(e)$ is the length of the long side of element e .

3. x_3 is the shortest distance between G_1 and G_2 ; the shortest distance is determined by two endpoints of G_1 and G_2 (see Figure 3.5 (b)).
4. x_4 is the curvature at the intersection determined by lengthening dotted lines G_1 and

G_2 (see Figure 3.5 (a)).

The fuzzy if-then rules of the fuzzy inference system is shown below; in the following rules, G_1 and G_2 stand for clusters of elements from dotted lines.

Rule 1: If x_1 is small, x_2 is small, x_3 is short, and x_4 is small, then G_1 and G_2 are probably part of the same dotted line.

Rule 2: If x_1 is large, then G_1 and G_2 are probably not part of the same dotted line.

Rule 3: If x_2 is large, then G_1 and G_2 are probably not part of the same dotted line.

Rule 4: If x_3 is long, then G_1 and G_2 are probably not part of the same dotted line.

Rule 5: If x_4 is large, then G_1 and G_2 are probably not part of the same dotted line.

Note that if G_1 and G_2 are chain lines of the same type, then two variables regarding the average of the numbers of pixels are needed: one is for short segments, the other is for long segments. Similarly, we need two variables regarding the average of the lengths. Therefore, when G_1 and G_2 are chain lines of the same type, the fuzzy inference system has 7 fuzzy if-then rules. Due to the limited space, definitions for the membership functions of the fuzzy sets are omitted.

3.3.4.2 Interference Criterion

Almost all of the broken line elements are in rectangular shapes; however, some of them are not. Non-rectangular elements are not classified as broken line elements. Furthermore, a broken line element overlapped with a graph element is not classified as a broken line element either. For this reason, a single broken line is sometimes divided into two or more parts; and therefore we need to introduce a merging process for the second criterion. This merging process is based on spline interpolations.

First, we select a pair of clusters, G_1 and G_2 , from the previous section. Then, G_1 and G_2 are examined by a fuzzy inference system whose output means a similarity between G_1 and G_2 . Here, the higher the similarity between G_1 and G_2 , the more plausible the fact that G_1 and G_2 are part of the same broken line. When the similarity between G_1 and G_2 is high, a natural cubic spline function, $y = s(x)$, is calculated using G_1 and G_2 ; the spline function is derived from the knots, which are endpoints of

the elements in G_1 and G_2 . We then count the number of pixels, (x, y) , satisfying the following conditions.

1. (x, y) satisfies condition $y = s(x)$, and is located between G_1 and G_2 .
2. (x, y) is a black pixel in the original image.

G_1 and G_2 are finally determined to be part of the same broken line if this number is large enough, and then merged into a single cluster. In this method, g is used to denote the path located between G_1 and G_2 , which is determined by the natural cubic spline function, $y = s(x)$. So, if the number of the pixels exceeds half of the length of g , then the two clusters are merged into a single cluster.

Let G_1 and G_2 be two clusters of elements of dotted lines. Fuzzy if-then rules of the fuzzy inference system that evaluates similarities are then shown below.

Rule 1: If x_1 is small and x_2 is small, then G_1 and G_2 are probably part of the same dotted line.

Rule 2: If x_1 is large, then G_1 and G_2 are probably not part of the same dotted line.

Rule 3: If x_2 is large, then G_1 and G_2 are probably not part of the same dotted line.

In these fuzzy if-then rules, the value of x_1 and x_2 are calculated by equations (3.3) and (3.4), respectively. Definitions of membership functions of the two fuzzy sets are omitted due to the limitations of space. Note that if G_1 and G_2 are chain lines of the same type, then two variables regarding the average of numbers of pixels are needed: one is for short segments and the other is for long segments. Similarly, we need two variables regarding the average of lengths. Therefore, when G_1 and G_2 are chain lines of the same type, the fuzzy inference systems has 5 fuzzy if-then rules.

3.4 Extraction of Solid Line Graph Elements

From Section 3.2, large elements (i.e., solid line graph elements) are separated from small elements. A large element is often formed by overlapping two or more graph elements such as rays of the x -axis and the y -axis and so on. The method for extraction of

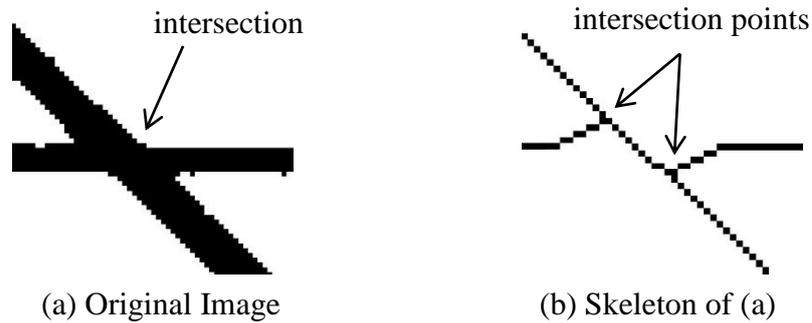


Figure 3.6: Intersection and Intersection Point

solid line graph elements is based on two steps: segmentation and merging.

3.4.1 Segmentation of Large Elements

We introduce the following five procedures for splitting large elements: (1) thinning, (2) removing short branches, (3) finding intersections, (4) splitting, and (5) cutting hooklets.

- (1) **Thinning:** We first apply a thinning procedure [19] to large elements, giving us skeletons of those elements.
- (2) **Removing Short Branches:** A skeleton often includes many undesirable short branches, and therefore every branch whose length is less than a threshold value is removed from the skeleton.
- (3) **Finding Intersections:** For every point, p , in a skeleton, using 3×3 spatial filters we identify whether p is an intersection point. The width of an original graph element is more than one pixel, and so not every intersection in the original image is a point. Therefore, an intersection includes more than one intersection point in the skeleton for a large element (see Figure 3.6). Thus, to group intersection points located in the same intersection, we apply single-linkage clustering to the set of intersection points.
- (4) **Splitting:** By removing all intersection points in a skeleton, the skeleton is divided into small fragments. For a fragment, e , if intersection points adjacent to the two

endpoints of e are members of the same cluster, then we remove e from the set of fragments. Furthermore, a fragment is also removed if its length is less than a threshold value. The remaining fragments are called primitive elements.

- (5) Cutting Hooklets: Due to the thinning algorithm, a fragment often has a hooklet at the extreme tip (see Figure 3.6 (b)). To facilitate the merging process in the following section, we remove the hooklets of fragments. We use Wall and Danielsson's method [20] to check if a hooklet exists.

3.4.2 Merging Primitive Elements

We apply fuzzy inference to merge primitive elements, so that the merging result forms a graph element. In this subsection, we apply fuzzy inference twice. The first fuzzy inference system is applied to examine how geometrically similar two primitive elements are. The second fuzzy inference system is then applied to merge primitive elements selected by the first fuzzy inference system.

If two primitive elements, e_1 and e_2 , satisfy the following three geometric characteristics, it is plausible that e_1 and e_2 are part of the same graph element.

- (1) e_1 and e_2 have been connected at the same intersection before applying the segmentation process from subsection 3.4.1.
- (2) Let C be a curve which is part of a skeleton from subsection 3.4.1, and suppose C includes e_1 and e_2 . A curvature of C at a point p is then low, where p is the intersection point that divides e_1 and e_2 .
- (3) The width of the original graph element corresponding to e_1 is almost equal to the width of the original graph element corresponding to e_2 .

The first fuzzy inference system has two arguments: curvature, x_1 , and width, x_2 . In the case where two primitive elements, e_1 and e_2 , are connected to the same intersection, e_1 and e_2 are applied to the first fuzzy inference system. The fuzzy if-then rules of the first fuzzy inference system are shown below.

Rule 1: If x_1 is low and x_2 is small, then the similarity between the two primitive

elements e_1 and e_2 is high.

Rule 2: If x_1 is high, then the similarity between the two primitive elements e_1 and e_2 is low.

Rule 3: If x_2 is large, then the similarity between the two primitive elements e_1 and e_2 is low.

The first fuzzy inference system checks the geometrical similarity between any pair of primitive elements. We then select primitive elements if the similarity exceeds a threshold value, and let E be the set of such primitive elements. Next, we apply the second fuzzy inference system to any pair of primitive elements in E . The second fuzzy inference system has three arguments: x_1 , x_2 , and x_3 ; x_1 and x_2 are same as the two arguments of the first one, while the third argument, x_3 , takes the shortest distance between two primitive elements. The fuzzy if-then rules of the second fuzzy inference system are shown below.

Rule 1: If x_1 is low, x_2 is small, and x_3 is short, then the two primitive elements e_1 and e_2 are probably part of the same graph element.

Rule 2: If x_1 is high, then the two primitive elements e_1 and e_2 are probably not part of the same graph element.

Rule 3: If x_2 is large, then the two primitive elements e_1 and e_2 are probably not part of the same graph element.

Rule 4: If x_3 is long, then the two primitive elements e_1 and e_2 are probably not part of the same graph element.

Two primitive elements are merged into one graph element if the value from the second fuzzy inference system exceeds a threshold value.

3.5 Fitting and Classifying Graph Element

Straight lines, circles, ellipses, circular arcs, et cetera, are expressed as basic shapes in the SVG specification; the other curves are expressed by piecewise cubic Bézier curves [6].

Given the finite set of points of an element, denoted by $P = \{(x_i, y_i) | 0 \leq i \leq n\}$, the method of least squares is applied to P to determine a model: a straight line, a circle, an ellipse, or a circular arc. Here, we minimize the sum of the squared distances between the points and the corresponding points on the model. If the sum is smaller than a threshold value, the set of points is classified as the model.

3.5.1 Straight Line, Circle and Arc Classification

To express a straight line as a SVG code, the two endpoints of the straight line are needed. These endpoints are determined as the two points on the model which are the nearest to the two terminal points of P .

The SVG specification requires the center coordinate and the radius to express a circle. First, we calculate the arithmetic mean \bar{x} of the x -coordinates, x_i 's, and also the arithmetic mean \bar{y} of the y -coordinates, y_i 's. Let $\mu_i = x_i - \bar{x}$ and let $\vartheta_i = y_i - \bar{y}$ for $i = 0, 1, \dots, n$. We solve the problem first in (μ, ϑ) coordinates, and then transform back to (x, y) coordinates. Let the circle have the center (μ_c, ϑ_c) and radius r . We minimize

$$\sum_{i=0}^n \left((u_i - u_c)^2 + (v_i - v_c)^2 - r^2 \right)^2$$

Solving this problem by the method of least squares, we have the center coordinate, (μ_c, ϑ_c) , and the radius, $r = \sqrt{\mu_c^2 + \vartheta_c^2 + (S_\mu + S_\vartheta)/n}$, where $S_\mu = \sum_{i=0}^n \mu_i^2$ and $S_\vartheta = \sum_{i=0}^n \vartheta_i^2$; the center (x_c, y_c) of the circle in the original coordinates system is $(x_c, y_c) = (\mu_c, \vartheta_c) + (\bar{x}, \bar{y})$.

To find the model of a circular arc, we first find the circle which best fits to P , and the endpoints of the circular arc are then determined as the two points on the circle which are the nearest to the terminal points of P .

3.5.2 Ellipse Classification

When we write the SVG code of a general ellipse, SVG requires the center coordinate, (x_c, y_c) , the gradient, α , of the major axis, and the standard equation, $(x/a)^2 +$

$(y/b)^2 = 1$, which is given by applying the two transformations to the general ellipse: the translation which moves the origin $(0, 0)$ to the point $(-x_c, -y_c)$ and the rotation whose angle is $\tan^{-1}\alpha$. Let \bar{x} and \bar{y} be the arithmetic means of x -coordinates and y -coordinates, respectively. We then put the arithmetic mean (\bar{x}, \bar{y}) to the center coordinate, (x_c, y_c) . Next, Principal Component Analysis (PCA) [21] is applied to P , and the gradient of the first principal component is set to α the gradient of the major axis. After that, the two transformations mentioned above are applied to P . Let (μ_i, ϑ_i) be the point transformed from (x_i, y_i) by the two transformations. We then solve the least squares problem:

$$\sum_{i=0}^n \left((b^2 u_i^2 + a^2 v_i^2 - a^2 b^2) \right)^2 \rightarrow \text{minimize}$$

3.5.3 Cubic Bézier Curve Fitting

If a graph element is not classified as one of the models above, we then apply an algorithm of piecewise cubic Bézier curve fitting, introduced by Schneider [22], to P , the set of points.

A Bézier curve of degree n is defined as

$$Q(t) = \sum_{i=0}^n V_i B_i^n(t), \quad t \in [0, 1] \quad (3.5)$$

where the V_i 's are the control points, and the $B_i^n(t)$'s are the Bernstein polynomials,

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, \dots, n \quad (3.6)$$

where $\binom{n}{i}$ is the binomial coefficient. Equation (3.5) is called a cubic Bézier curve when $n = 3$. Figure 3.7 shows an example of cubic Bézier curves; the four points, $V_0, V_1, V_2,$ and V_3 , are the control points, the thick curve is the cubic Bézier curve, and the polygon expresses the one constructed by the four control points.

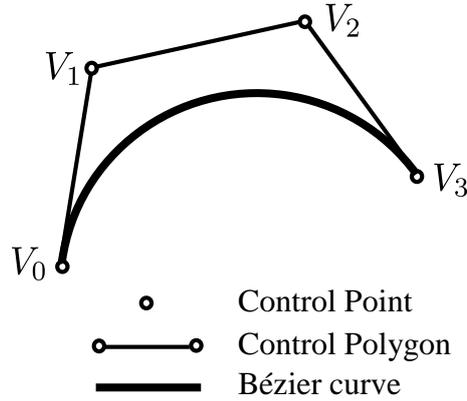


Figure 3.7: A Single Cubic Bézier Segment

The problem of finding a cubic Bézier curve which best fit the set of points P is stated in this manner: given a set of points, find a cubic Bézier curve that fits these points within some given tolerance. The fitting criterion here is to minimize the sum of the squared distances from the points to their corresponding points of the curve. Formally, we minimize the function, S , defined by

$$S = \sum_{i=0}^n (p_i - Q(q_i))^2 \quad (3.7)$$

where p_i is the (x_i, y_i) coordinates of P and q_i is the parameter value associated with p_i . To solve this least squares problem, the following conditions are considered.

- (1) V_0 and V_3 , the first and last control points, are given; they are set to be the first and last points of P .
- (2) Let \vec{t}_1 and \vec{t}_2 be the unit tangent vectors at V_0 and V_3 , respectively. Then, $V_1 = \alpha_1 \vec{t}_1 + V_0$ and $V_2 = \alpha_2 \vec{t}_2 + V_3$ hold; that is, the two inner control points, V_1 and V_2 , are each some distance from the nearest end control point, in the tangent vector direction.

Then, our problem can be defined as finding α_1 and α_2 to minimize S ; that is, we solve the following two equations for α_1 and α_2 to determine the inner control points V_1 and V_2

$$\frac{\partial S}{\partial \alpha_1} = 0 \quad \text{and} \quad \frac{\partial S}{\partial \alpha_2} = 0 \quad (3.8)$$

Let p be the point that is located at the farthest distance from the cubic Bézier curve, which is given by Equation (3.7). If the distance from p to the curve exceeds a threshold value, then the set of points P is divided at point p . Then, equation (3.7) is applied to both of the two parts until the farthest distance does not exceed the threshold.

3.6 Experimental Results

Figure 3.8 shows some sample images of mathematical graphs which are selected from mathematics and science textbooks. Then, these graphs were captured by using an image scanner, whose resolution was set at 600 dpi. These electronic images were saved in 24-bit bitmap format. The size of images is about 1,500×1,500 pixels. All the graphs are processed in a binary format; we can assume that the images dealt with in this paper are monochromatic (i.e., black for foreground and white for background). We applied these mathematical graphs to our method and examined the effectiveness of the method.

3.6.1 Results of Broken Line Extraction

Figure 3.9 shows the experimental results of Figure 3.8. In Figure 3.9, broken line elements of the same cluster are represented by the same color. For the mathematical graphs of Figure 3.8 (a) ~ (e) we have correct results, i.e., every broken line of these 5 mathematical graphs has been correctly extracted. On the other hand, we do not have correct results from the mathematical graphs of Figure 3.8 (f) ~ (h). We found three major reasons why we have incorrect results, and discuss them below.

- (1) Let us consider the case for Sample 6. There are two different broken lines in this figure, however our method merged them into a single broke line. This merging is done by the process based on spline functions because there were too many pixels between them.
- (2) Let us consider the case for Sample 7. This is the opposite case of (1). In Figure 3.8

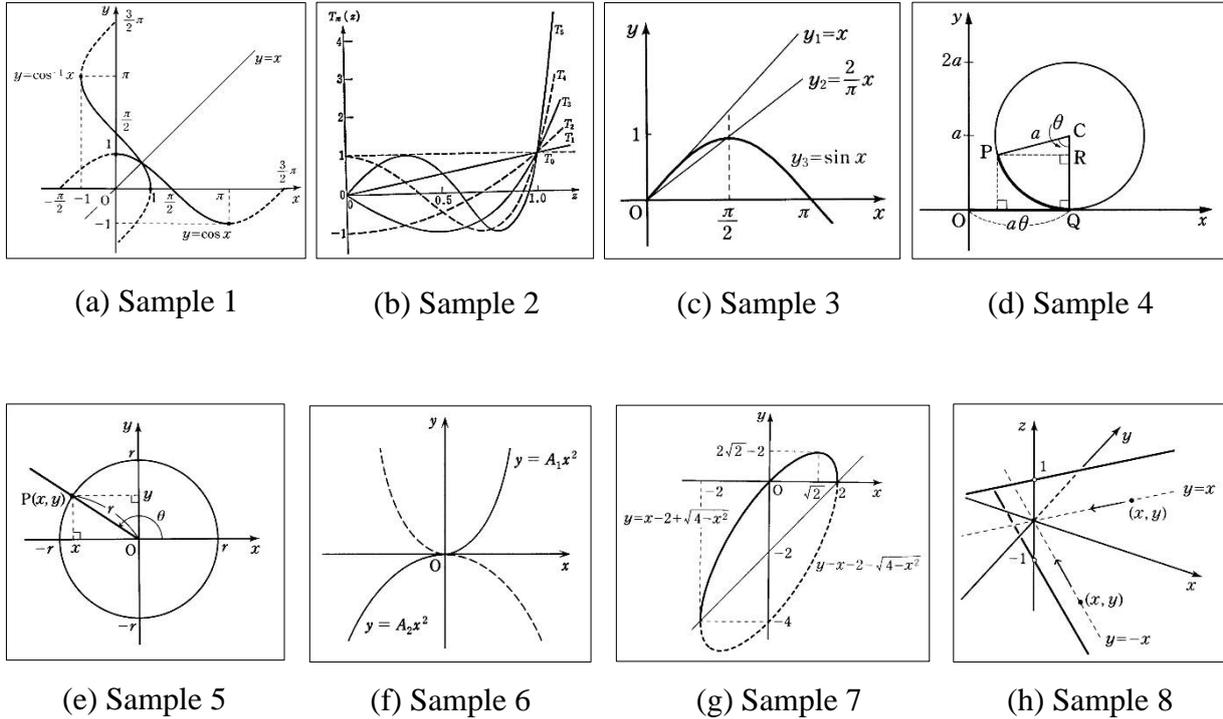


Figure 3.8: Examples of Mathematical Graphs

(g), the broken line, which is part of the ellipse, was classified as the two different broken lines (the blue and the green). This is because the number of the pixels between the two broken lines is not sufficient for the process based on spline functions to merge them into a single broken line.

(3) Let us consider the case for Sample 8. In this figure, there are two black dots on the two broken lines to emphasize two points (x, y) s. In Figure 3.9 (h), the red broken line, which includes the upper right circle, was classified as a type 3 chain line. However, the broken line elements, which are next to the red one, were classified as a dotted line. So, our method was not able to find the broken line including the black circle.

We have applied 33 mathematical graphs which include broken lines to our method, and examined the effectiveness of the method. The mathematical graph of Sample 6 is the only graph that two different broken lines are merged into a single broken line. Most incorrect results come from the 2nd case, i.e., two clusters are not merged into a single cluster by the merging process based on spline functions since there are too few pixels

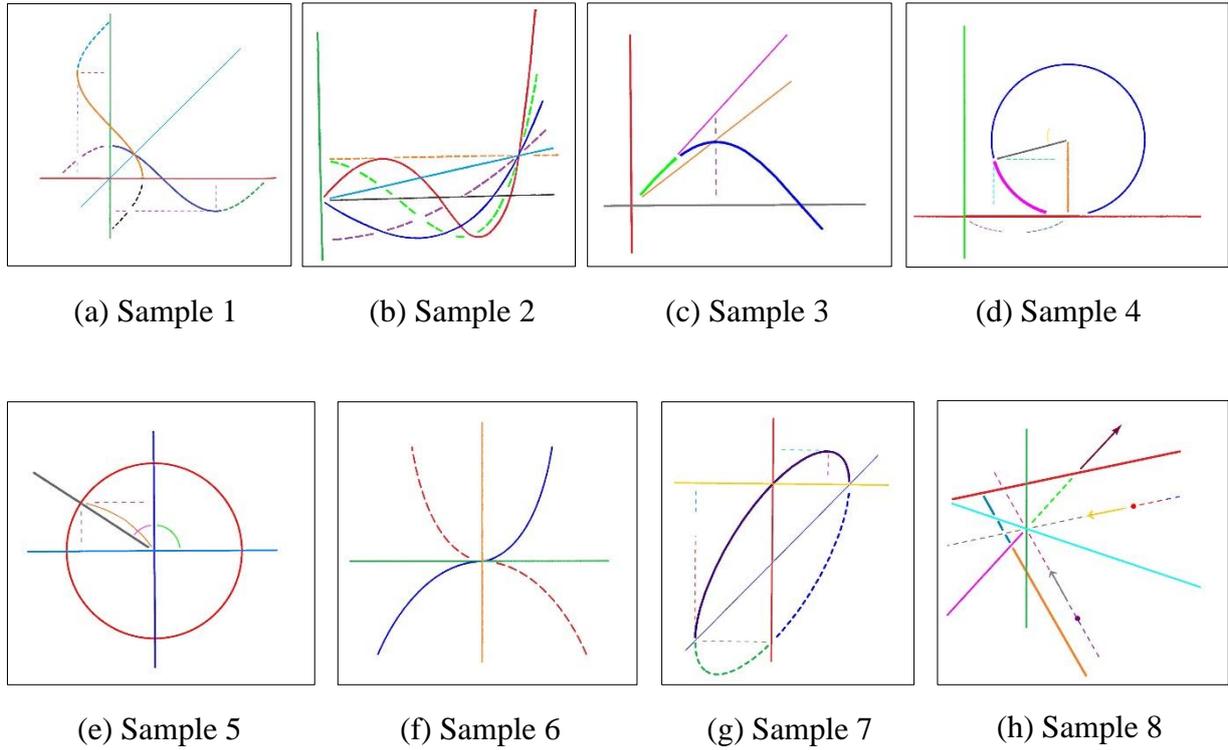


Figure 3.9: Experimental Results of Figure 3.8

between the two clusters.

We use recall, precision, and f-measure to evaluate the accuracy of our method for broken line elements extraction. Table 3.1 shows the accuracy for broken line elements extraction. C is used to denote the number of all actual elements in experimental graphs which is calculated by observation. N is used to denote the number of extracted elements including positive results and negative results. R is used to denote the number of correctly extracted results.

Table 3.1: Accuracy of Extraction for Broken Line Elements

	C	N	R	Recall	Precision	F-measure
Broken Line	116	135	83	0.72	0.61	0.66

where, $\text{Recall} = \frac{R}{C}$, $\text{Precision} = \frac{R}{N}$, and $\text{F-measure} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}$.

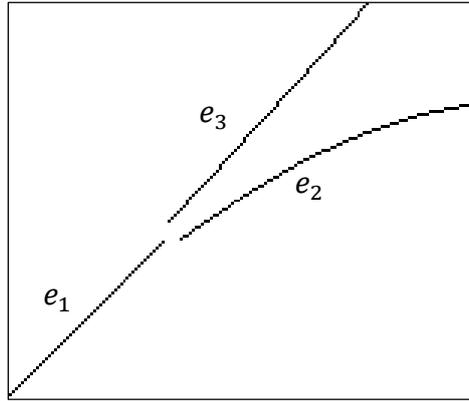


Figure 3.10: Skeleton Image for Sample 3

3.6.2 Results of Solid Line Extraction

For the mathematical graphs of Figure 3.8 (a), (b), (f) ~ (h), we have correct results. However, we do not have correct results for Figure 3.8 (c) ~ (e). The reasons why we have incorrect results are categorized into two groups: one is the thickness of a curve, and the other is the curvature at an intersection. We discuss the reasons why we have the incorrect results below.

- (1) There is a small arc inside the circle of Figure 3.8 (e). This arc was divided into two parts at the intersection between the arc and the y -axis (see Figure 3.9 (e)). Since the curvature of the arc at the intersection is large, our method did not merge the two fragments into a single arc.
- (2) In the original image, some part of a graph element is emphasized using a thick line (see Figure 3.8 (d)). By the division process from Section 3.4, we have the two primitive elements: the pink one and the blue one in Figure 3.9 (d). Although they are part of the same graph element, the pink line is much thicker than the blue line, so they were not merged into a single graph element.
- (3) In Figure 3.8 (c), the line, $y_1 = x$, is a tangent to the curve, $y_3 = \sin x$, at the origin. Figure 3.10 shows the skeleton of the two graphs around the intersection point of Figure 3.9 (c). In Figure 3.10, e_1 and e_2 are part of the curve, and e_1 and e_3 are also part of the line. However, in the skeleton (i.e., Figure 3.10), the smoothness between e_1 and e_2 is degraded, and the smoothness between e_1 and

e_3 is also degraded. So, the curvature between e_1 and e_2 is not small enough to merge them into a single graph element. We see the same degradation in the line between e_1 and e_3 . Therefore, the tangent and the curve were divided into the three parts, and they did not merge together.

After applying 38 mathematical graphs which include solid line elements to our method, 6 graphs produced wrong results. Most wrong results comes from the 3rd case. Table 3.2 shows the accuracy for solid line elements extraction.

Table 3.2: Accuracy of Extraction for Solid Line Elements

	<i>C</i>	<i>N</i>	<i>R</i>	Recall	Precision	F-measure
Solid Line	224	222	183	0.817	0.824	0.82

3.6.3 Results of Graph Element Fitting

We selected more than 30 mathematical graphs. After performing our method for extracting graph elements, we applied the fitting procedures to the graph elements to find their models. All the models were fitted correctly to the graph elements by visual observation.

3.6.4 SVG and Edel Document Production

The final goal of our system is to output SVG documents and Edel files. SVG has the formats to express the four shapes: straight line, circle, circular arc, and curve. Therefore, we can create an SVG document for a mathematical graph directly once we have detected the shapes for all the elements of the graph. Edel [23] is a software system to create digital files available to produce tactile graphics by Braille embossers, and this system is used widely in Japanese blind schools. An Edel document is a collection of embossed dots, that is, an Edel document is similar with the expression for raster graphics. So, it is easy to create an Edel document once we have detected the shapes for all the elements of a mathematical graph.

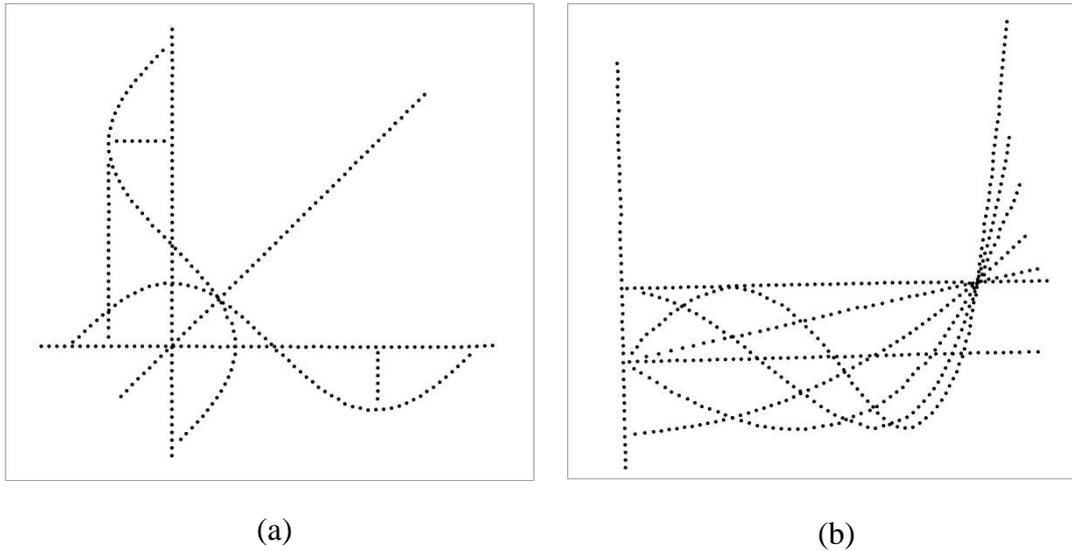


Figure 3.11: Tactile Graphics

Figure 3.11 (a), (b) are tactile graphic images of Sample 1 and 2 in Figure 3.9. As Figure 3.11 shows, it is good enough for people with visual impairment to understand the mathematical graphs if the bitmap image was the actual tactile graphic. However, we need to ensure our opinion, that is, we need to check if the visually impaired are able to understand the tactile graphics which are produced by our method. This is one of our future concerns.

3.7 Summary

In this chapter, we proposed a method of mathematical graph recognition for assisting tactile graphic production. The main procedures of our method are: (1) separating a graph into solid line elements, broken line elements, and character elements. More than 30 mathematical graphs were applied to do experimental result, and the effectiveness of our method was examined. Although our method works well for many examples, we have unsuccessful results. Fixing these drawbacks is one of our future endeavors.

The final goal of our research is to convert bitmap images into SVG descriptions. Therefore, in the near future, SVG codes will be created to express mathematical graphs.

References

- [1] S. Oouchi, M. Sawada, T. Kaneko and K. Chida, “A Survey on Making and Using Tactile Educational Materials in Schools for the Blind”, *Bulletin of National Institute of Special Needs Education*, Vol.31, pp.113-125, 2004.
- [2] Y. Hirayama, “A Block Segmentation Method for Document Images with Complicated Column Structures”, in *Proceedings of the 2nd International Conference on Document Analysis and Recognition*, pp.91-94, 1993.
- [3] S. Lee and D. Ryu, “Parameter-Free Geometric Document Layout Analysis”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.23, No.11, pp.1240-1256, 2001.
- [4] Y. Ishitani, “Document Layout Analysis by Interaction between Data-driven Processing and Concept-driven Processing”, *Information Processing Society of Japan*, Vol.42, No.11, pp.2711-2723, 2001.
- [5] <http://www.inftyproject.org/jp/>
- [6] J. David Eisenberg, *SVG Essentials*, O’Reilly, 2002.
- [7] <http://www.daisy.org/>
- [8] S. E. Krufka and K. E. Barner, “Automatic Production of Tactile Graphics from Scalable Vector Graphics”, in *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*, pp.166-172, 2005.
- [9] R. E. Ladner, M. Y. Ivory, R. Rao, S. Burgstahler, D. Comden, S. Hahn, M. Renzelmann, S. Krisnandi, M. Ramasamy, B. Slabosky, A. Martin, A. Licenski, S. Olsen and D. Groce, “Automating Tactile Graphics Translation”, in *Proceeding of the 7th International. ACM SIGACCESS Conference on Computer and Accessibility*, pp.150-157, 2005.
- [10] C. Jayant, M. Renzelmann, D. Wen, S. Krisnandi, R. Ladner and D. Comden, “Automated Tactile Graphics Translation: In the Field”, in *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pp.75-82, 2007.
- [11] T. Fuda, S. Omachi and H. Aso, “Recognition of Line Graph Images in Documents by Tracing Connected Components”, *IEICE Transactions on Information and Systems*, Vol.J86-D-II, No.6, pp.825-835, 2003.

- [12]S. Shimada, S. Kakumoto and M. Ejiri, “A Recognition Algorithm of Dashed and Chained Lines for Automatic Inputting of Drawings”, *IEICE Transactions on Information and Systems*, Vol.J69-D, No.5, pp.759-770, 1986.
- [13]N. Yokokura and T. Watanabe, “Recognition of Business Bar-graphs Using Layout Structure Knowledge”, *Information Processing Society of Japan*, Vol.40, No.7, pp.2954-2966, 1999.
- [14]M. Yo, T. Kawahara and R. Fukuda, “Handwriting Graphics Input System for High School Mathematics”, *Technical Report of IEICE*, PRMU2003-291, pp.43-48, 2004.
- [15]N. Takagi, “Pattern Recognition in Computer-Aided Systems for Transformation of Mathematical Figures into Tactile Graphics”, *International Journal of Intelligent Computing in Medical Sciences and Image Processing*, Vol.3, No.1, pp.43-56, 2009.
- [16]J. C. Bezdek, J. Keller, R. Krisnapuram and N. R. Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Kluwer Academic Publishers, 1999.
- [17]R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, 2nd Edition, A Wiley-Interscience Publication, 2001.
- [18]N. Ono and R. Takiyama, “On Calculations of Curvature of Sampled Curves”, *Technical Report of IEICE*, Vol.IE93-74, pp.7-14, 1993.
- [19]C. J. Hilditch, “Linear Skeletons from Square Cupboards”, in *Machine Intelligence*, Vol.IV, B. Meltzer and D. Michie, (Eds.), Elsevier, New York, pp.403-420, 1969.
- [20]K. Wall and P. Danielsson, “A Fast Sequential Method for Polygonal Approximation of Digitized Curves”, *Computer Vision, Graphics, and Image Processing*, Vol.28, Issue 2, pp.220-227, 1984.
- [21]R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*, Pearson Prentice Hall, 2007.
- [22]P. J. Schneider, “An Algorithm for Automatically Fitting Digitized Curves”, *Graphics Gems*, Andrew S. Glassner (Ed.), Academic Press, San Diego, pp.612-626, 1990.
- [23]Edel, <http://www7a.biglobe.ne.jp/EDEL-plus/>

Chapter 4

Automatic Translation of Hand-drawn Figures into Tactile Graphics

This chapter proposes a system which is developed for assisting automatic translation of hand-drawn maps into tactile graphics. In this chapter, firstly, several tactile graphic production systems are reviewed and discussed, and the advantages of our proposed system is depicted by comparing our system with other systems. Secondly, the techniques used in our system are described and discussed. Lastly, this chapter is concluded by evaluating the usability of our system.

4.1 Introduction

Producing tactile maps is an important effort to give blind people a more self-sufficient life. There have been many studies of computer-aided systems in order to assist the production of tactile graphics [1 - 6]. Tactile Map Automated Creation System (TMACS) [2, 3], for example, has been developed to produce tactile maps automatically. It is a web application and produces the digital file for a tactile map from the information about two places: a departure place and destination. TMACS assumes that users can be blind, and so it produces the tactile map automatically from the map database if a user only provides a departure place and destination to the system. However, tactile maps produced by TMACS are sometimes difficult to read for the blind because it is possible to include unnecessary information in the tactile maps. Further, Geospatial Information Authority of Japan has also developed a tactile map production system [4]. This system assumes that users are sighted people, and it is totally provided as a GUI application. Operating this GUI application is not easy for users who are not familiar with computers.

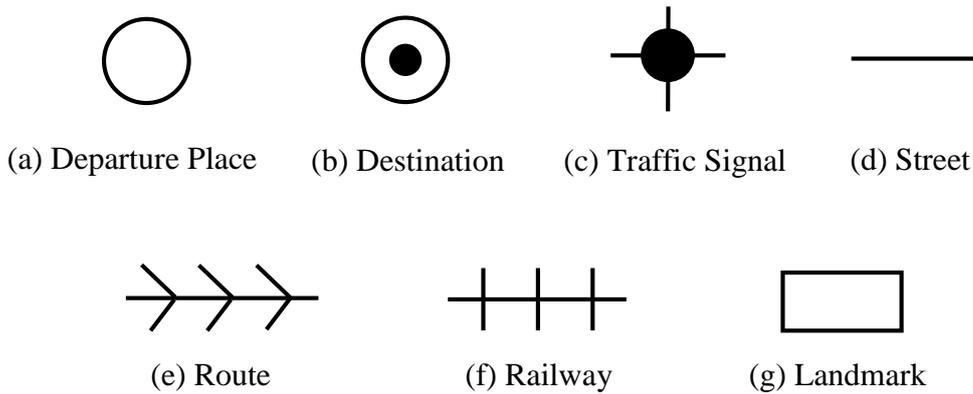


Figure 4.1: Object Types and Symbols

Based on the background above, we are now developing a system for automating production of tactile maps. In the tactile map production method using our system, a sighted user first draws manually a hand-drawn map using a pencil and paper, and the map is then converted to a digital image using an image scanner or a digital camera. Finally, by using our system, the digital image is recognized and translated into digital files which are available to produce the tactile maps. Our system chooses the Edel [7] and Scalable Vector Graphics (SVG) [8] documents as the output file formats. Here, Edel is a software system to create digital files available to produce tactile graphics by Braille embossers, and this system is used widely in Japanese blind schools. The advantages for our system are as follows.

- (1) Sighted users can draw maps by using pencils and papers; no computer operation is needed when drawing maps.
- (2) Sighted users can easily add necessary information and remove unnecessary information. This implies that providing tactile maps individually is not time consuming.

It is important to evaluate the usability for the method using our system. The evaluation was done by comparing with the methods using different software systems.

4.2 Outline For Our Method and Basic Procedures

In order to facilitate hand-drawn map recognition, we assume the following conditions

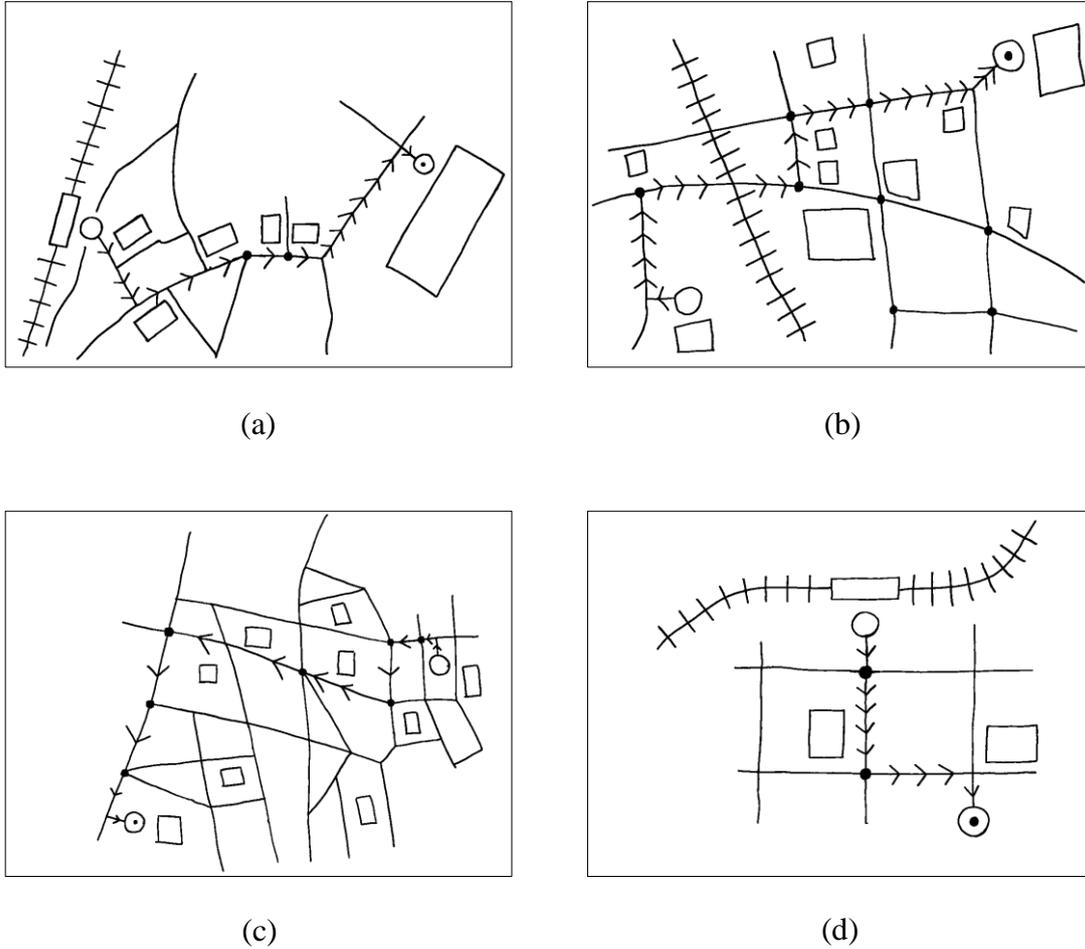


Figure 4.2: Examples for Hand-drawn Maps

for drawing maps.

- (1) A hand-drawn map consists of the following object types: departure place, destination, traffic signal, route, railway, landmark, and street. The symbols for objects are summarized in Figure 4.1.
- (2) A hand-drawn map is a line drawing, except for traffic signals and the bullet for the destination symbol.
- (3) A landmark object is represented by a polygon and forms a connected component. Furthermore, it does not connect to any other objects.

Figure 4.2 shows examples for hand-drawn maps. Inputting the digital image for a hand-drawn map to our system, it outputs two digital files for tactile maps: the SVG and Edel documents. The procedure for hand-drawn map translation is outlined in Figure 4.3.

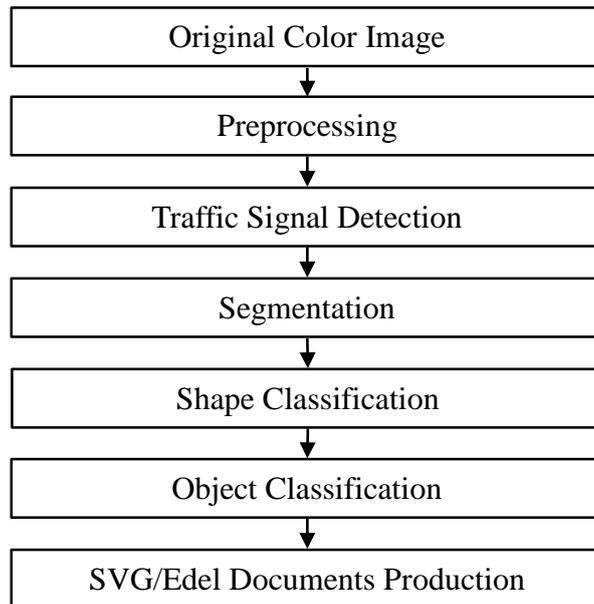


Figure 4.3: Outline for Hand-drawn Map Translation

In our tactile map production method, a map is first drawn manually using a pencil and paper. Then, the hand-drawn map is captured by an image scanner or a digital camera. Finally, our system translate the digital image for the hand-drawn map into SVG and Edel documents, producing the tactile maps.

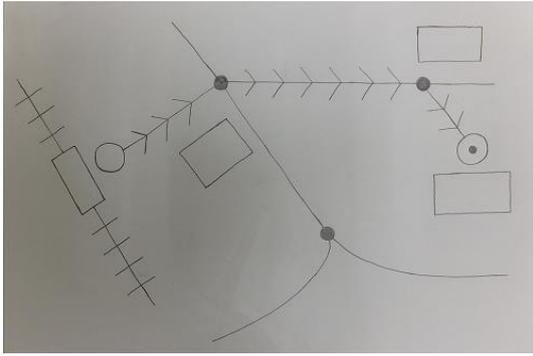
There are various new technologies applied in hand-drawn sketch recognition. For example, Broelemann, et al. [9] developed a method for automatic street graph construction of hand-drawn maps, but this method is restricted to detection of streets. The sketch recognition methods introduced in [10 - 14] cannot be used to recognize the hand-drawn maps due to their limitations. So, a new method was proposed to recognize the hand-drawn maps in this chapter.

In the remainder of this section, preprocessing, traffic signal detection, segmentation, shape classification, and object classification will be explained.

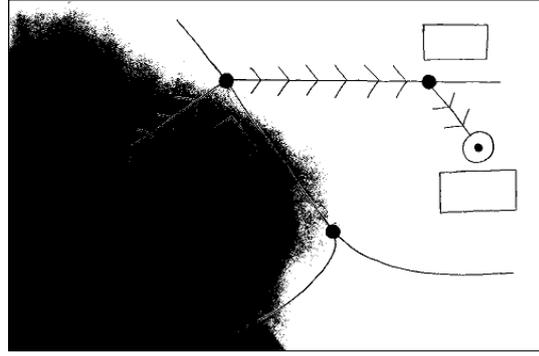
4.2.1 Preprocessing and Traffic Signal Detection

The preprocessing consists of the following three procedures.

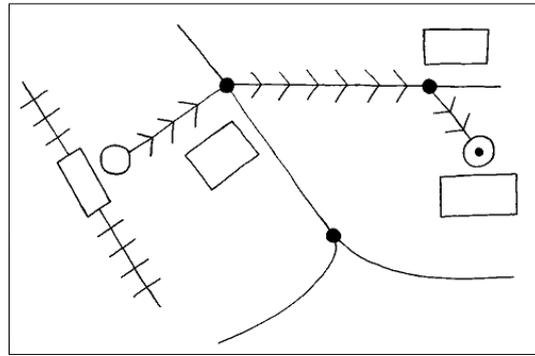
- (1) An original color image is converted to a grayscale image I by the formula



(a) Captured Image



(b) Threshold Image by Otsu's Method



(c) Threshold Top-hat Image

Figure 4.4: Shading Correction by Using Top-hat Transformation

$$I = 0.2989 \cdot R + 0.5866 \cdot G + 0.1145 \cdot B \quad (4.1)$$

where R , G , and B are the intensities for red, green, and blue of the original image.

(2) The two noise reduction methods are applied to the grayscale image.

(a) The median filter with a 3×3 window is first applied to eliminate salt-and-pepper noise, and then

(b) a grayscale morphological top-hat transformation [15] is applied to correct the effects of non-uniform illumination. Here, a disk structuring element of radius 3 pixels is used for the top-hat transformation. The effect for a top-hat transformation is explained in Figure 4.4.

(3) The grayscale image is converted to a binary image using Otsu's method [16].

Mathematical morphology is applied to extract traffic signals from the binary image. First, an erosion operator is performed to the binary image for three times, and a dilation operator is then applied to the eroded image for three times. The structuring element is a disk of radius 3 pixels. The erosion operation eliminates thinner line segments, but it does not remove traffic signals completely. The dilation operation then detects all the traffic signals. From this detection, the bullet for the destination symbol is also extracted.

4.2.2 Segmentation and Shape Classification

After detecting traffic signals, we segment objects into fragments. The four procedures are introduced to divide objects: (1) thinning, (2) short branch removal, (3) feature point detection, and (4) dividing.

- (1) Hilditch's thinning algorithm [17] is first applied to the binary image, and the skeleton image is obtained.
- (2) Short Branch Removal: Every short branch of the skeleton is removed if its length is shorter than a threshold value.
- (3) Feature Point Detection: Intersections and corner points are detected in this stage. A 3×3 sliding window is applied to the skeleton image in order to detect intersections. The method for detecting corner points is explained below.
- (4) Dividing is as follows. By removing all intersections and corner points in the skeleton, the skeleton is divided into fragments. It is characteristic that every fragment has two endpoints or no endpoint. Fragments are called elements in this chapter.

Corner Point Detection. There have been many studies for detecting corner points in digital images [18 - 20]. We have introduced a new method to precisely detect corner points for hand-drawn figures. This method first finds a piecewise linear approximation (PL approximation, for short) for a digital curve; this approximation is detected by Wall and Danielsson's algorithm [21]. Let O , P_1 , and P_2 be the origin and the two different points on the xy plane (see Figure 4.5). If the ratio of the area for the triangle OP_1P_2 to the length of the line segment $\overline{OP_2}$ exceeds a threshold value, then the point P_1 is chosen as a point of the PL approximation; if not, the next point P_2 is examined. Let us consider a digital curve, denoted by a sequence of the points $C = P_0, P_1, \dots, P_{n-1}$. The Wall and Danielsson's algorithm is then presented below.

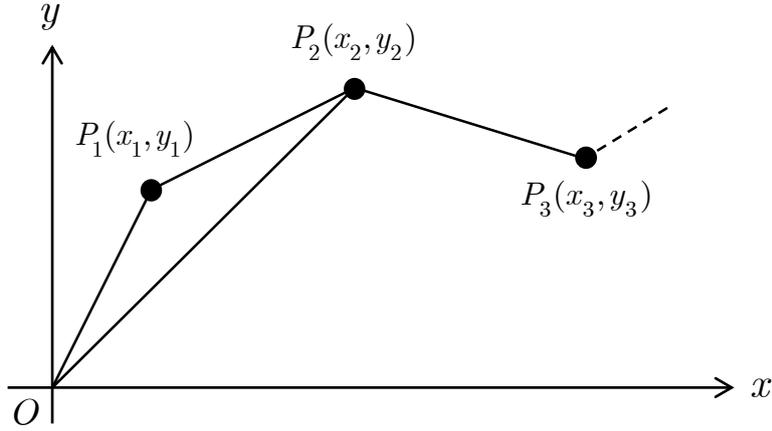


Figure 4.5: Piecewise Linear Approximation

Step 1: Set $i \leftarrow 1$, and choose the point P_0 as the first starting point for the PL approximation.

Step 2: Let O be the point P_{i-1} , and set $e_i \leftarrow 0$.

Step 3: Set $i \leftarrow i + 1$, and calculate e_i and ℓ_i . Consider

$$\begin{aligned} e_i &\leftarrow e_{i-1} + \Delta e_i \\ \ell_i &\leftarrow \sqrt{x_i^2 + y_i^2} \end{aligned} \quad (4.2)$$

where Δe_i is the value $x_{i-1}y_i - x_i y_{i-1}$ which is the signed area for the parallelogram spanned by the two vectors $\overrightarrow{OP_{i-1}}$ and $\overrightarrow{OP_i}$.

Step 4: If $|e_i| \leq t \cdot \ell_i$ holds, then go to Step 3; if not, choose P_{i-1} as a point of the PL approximation, and then go to Step 2. Here, t is a threshold value.

The above algorithm cannot detect corner points correctly. However, if a corner point exists, it must be close to a point of the PL approximation. So, we improve the Wall and Danielsson's algorithm by exchanging Step 4 with the following Step 4' so that the algorithm is able to precisely detect corner points of a hand-drawn digital curve.

So, we introduce two procedures explained below, and our corner detection procedure is the one that is obtained by exchanging Step 4 with the following Step 4'.

Step 4': If $|e_i| \leq t \cdot \ell_i$ holds, then go to Step 3; if not, the point P_{i-1} is detected as a

point of the PL approximation, and perform the following process to test whether a corner point exists near the point P_{i-1} .

- (1) Conduct the following procedure 1 whose input is P_{i-1} . If this procedure returns a point, choose this point as a corner point of the digital curve C , and then go to Step 2.
- (2) Conduct the following procedure 2 whose input is P_{i-1} . If this procedure returns a point, choose this point as a corner point of the digital curve C , and then go to Step 2.

Procedure 1 aims to detect a sharp corner point. This procedure is performed in the following steps.

Step 1: Let P_{i-s} ($s > 1$) be a point of the digital curve C satisfying the following two conditions:

- (1) P_{i-s} has been chosen as a point of the PL approximation;
- (2) P_{i-1} is the next PL approximation point to P_{i-s} .

Step 2: For every point P_{i-s+k} of the digital curve C between P_{i-s} and P_{i-1} ($k = 1, 2, \dots, s-1$), calculate the distance, denoted by ℓ_k , between P_{i-s} and P_{i-s+k} . Then, let L be the sequence $\ell_1, \ell_2, \dots, \ell_{s-1}$.

Step 3: For the sequence L , if the distances for L monotonously increase in the middle and the monotonously decrease to the end, that is, the inequalities $\ell_1 < \ell_2 < \dots < \ell_q > \ell_{q+1} > \dots > \ell_{s-1}$ hold, then the point P_{i-s+k} is determined as a corner point of the digital curve C .

Next, the procedure 2 is explained; it is motivated by the method for the literature [22]. Remember that the point P_{i-1} has been chosen as a point of the PL approximation.

Step 1: For every P_{i-k} ($k = 1, 2, \dots, 21$), calculate the sharpness $s(P_{i-k})$; the procedure to calculate sharpness is explained below.

Step 2: For the sequence of the sharpnesses $s(P_{i-1}), s(P_{i-2}), \dots, s(P_{i-21})$, calculate the simple moving averages $svm(P_{i-1}), \dots, svm(P_{i-21})$.

Step 3: If the maximum value of $svm(P_{i-1}), \dots, svm(P_{i-21})$ exceeds a threshold value, then the point P_{i-k} which gives the maximum value is determined as a corner point of the digital curve C ; if not, we decide that no corner point exists near the point P_{i-1} .

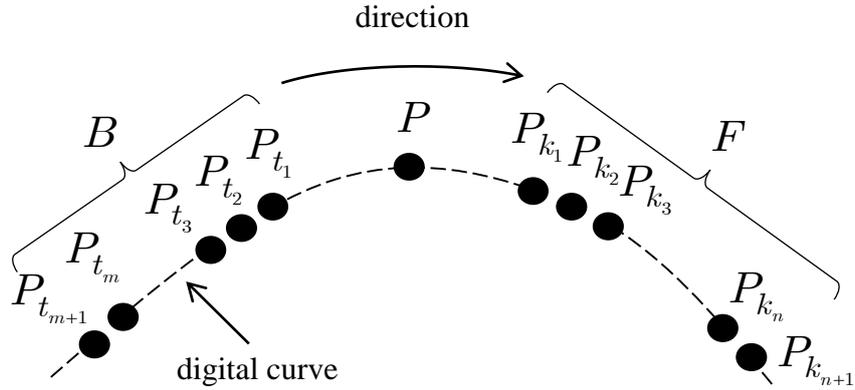


Figure 4.6: Sharpness

The calculation for the sharpness $s(P)$ for a point P is as follows. We first detect two subsequences of points, denoted by $F = P_{k_1}, \dots, P_{k_n}$ and $B = P_{t_1}, \dots, P_{t_m}$, in the following way (see Figure 4.6). We visit the points forward from P and select a point P_{k_j} , the distance between P_{k_j} and P is more than or equal to 2 pixels and less than or equal to 10 pixels; P_{k_1} is the first point satisfying this condition. We select points until the condition breaks; P_{k_n} is the last point satisfying the condition. The subsequence B is obtained similarly, while visiting the points backward from P . The sharpness $s(P)$ is then defined as the following formula:

$$s(P) = \frac{1}{nm} \sum_{r=1}^n \sum_{s=1}^m (\text{the angle of } \angle P_{k_r} P P_{t_s}) \quad (4.3)$$

Next, shape classification is explained. Every element is classified into one of the four shapes: straight line, circle, circular arc, and curve. This classification is done by the method of least squares. That is, we minimize the sum of the squared Euclidean distances between the points of the element and the corresponding points on the model. The element is then classified as the shape for the model if the sum is smaller than a threshold value. A curve is expressed by a piecewise cubic Bézier curve. The detail of the methodology for shape classification is introduced in section 3.5 of Chapter 3.

4.3 Object Classification

In this section, object classification is described. First of all, a landmark object is classified by the following simple procedure: if an object is isolated and closed and has more than 3 corners, the object is classified as a landmark. Except for landmark classification, fuzzy inference is applied to design the classification methods. The fuzzy inference systems introduced in this chapter are constructed by Mamdani's fuzzy inference method [23], but the minimum operator is exchanged with the product operator. To detect route and railway objects, it is necessary to find arrows and crosses. So, elements are first grouped into a single cluster if they are connected to the same intersection. Every cluster is then examined if it is an arrow or a cross. Note that an element can belong to different two clusters when the two endpoints of the element connect to the different intersections.

4.3.1 Arrow Classification

An arrow consists of three or four straight line elements (see Figure 4.7 (a) and (b)). We apply two fuzzy inference systems to calculate the similarity for arrow. The following description is the outline of the first fuzzy inference system.

- (1) We first choose a cluster, E , which includes four straight line elements, denoted by $e_1, e_2, e_3,$ and e_4 . The four angles, $\varphi_1, \varphi_2, \varphi_3,$ and φ_4 are then measured (see

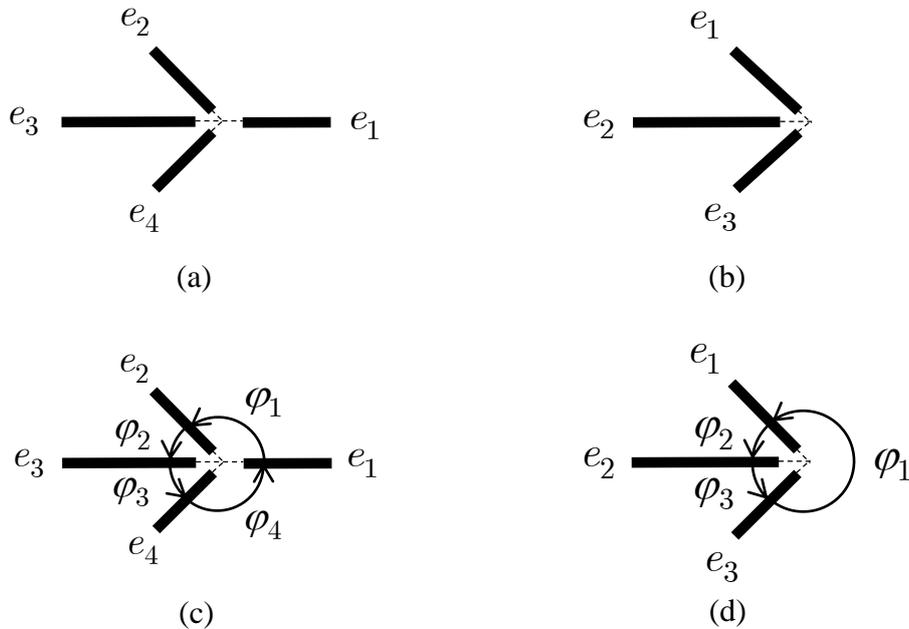


Figure 4.7: Elements for Arrows

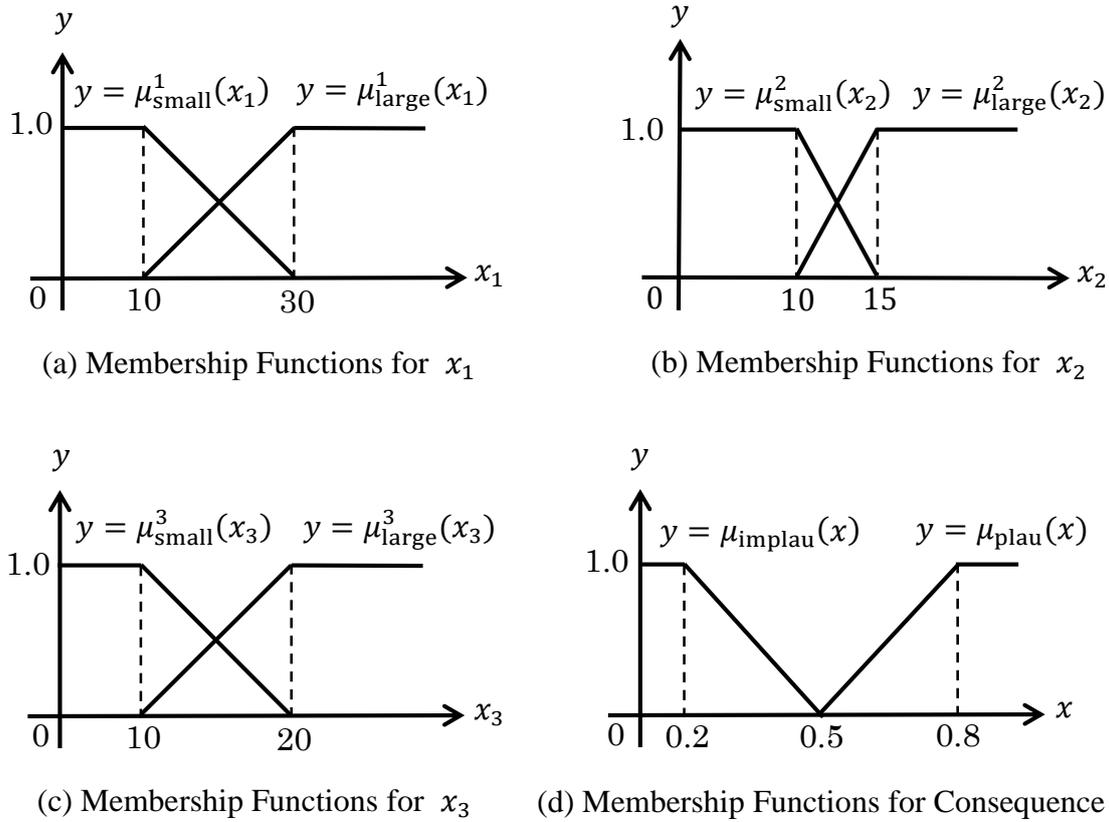


Figure 4.8: Membership Functions

Figure 4.7 (c)).

(2) Next, the following three attribute values are calculated:

- (1) the standard deviation of φ_1 , φ_2 , φ_3 , and φ_4 ; it is denoted by x_1 .
- (2) the average value of $|\varphi_1 - \varphi_4|$ and $|\varphi_2 - \varphi_3|$; it is denoted by x_2 .
- (3) the difference between the two lengths of the elements e_2 and e_4 ; it is denoted by x_3 .

(3) The fuzzy inference system is applied to x_1 , x_2 , and x_3 . The fuzzy if-then rules are denoted below, and the membership functions are shown in Figure 4.8. We then have a value, s , of $[0, 1]$ from the system; s implies the similarity for E .

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

Rule 2: If x_1 is small, then it is implausible that E is an arrow.

Rule 3: If x_2 is large, then it is implausible that E is an arrow.

Rule 4: If x_3 is large, then it is implausible that E is an arrow.

The second fuzzy inference system is applied to a cluster, E , which includes three straight line elements. This system has the following three input attributes:

- (1) the standard deviation for φ_1 , φ_2 , and φ_3 (see Figure 4.7 (d)); it is denoted by x_1 ;
- (2) the difference between φ_2 and φ_3 ; it is denoted by x_2 ;
- (3) the difference between the two lengths of the elements e_1 and e_3 ; it is denoted by x_3 .

The following rules are the fuzzy if-then rules. The membership functions are omitted.

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

Rule 2: If x_1 is small, then it is implausible that E is an arrow.

Rule 3: If x_2 is large, then it is implausible that E is an arrow.

Rule 4: If x_3 is large, then it is implausible that E is an arrow.

4.3.2 Cross Classification

A cross consists of four straight line elements (see Figure 4.9 (a)). A cluster, E , is applied to the fuzzy inference system for cross classification, if E includes four straight line elements, e_1 , e_2 , e_3 , and e_4 . This fuzzy inference system requires the following two input attributes:

- (1) the standard deviation of φ_1 , φ_2 , φ_3 , and φ_4 (see Figure 4.9 (b)); it is denoted by x_1 .
- (2) the standard deviation for $\varphi_1 + \varphi_2$, $\varphi_2 + \varphi_3$, $\varphi_3 + \varphi_4$, and $\varphi_1 + \varphi_4$; it is denoted

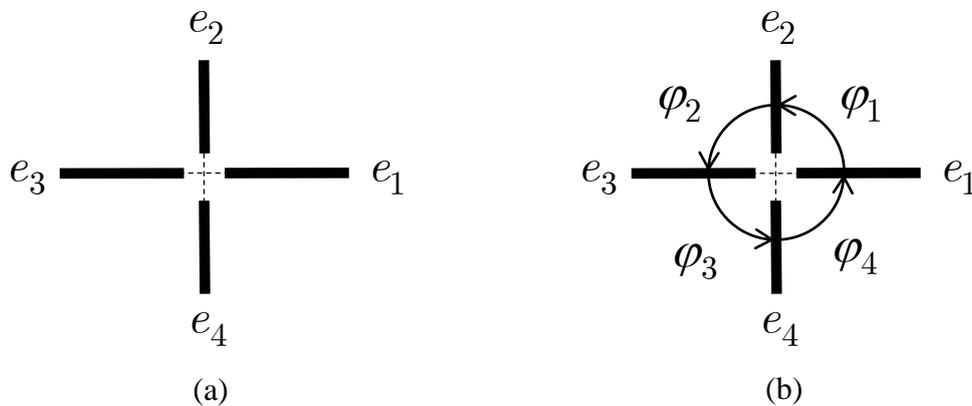


Figure 4.9: Elements for Cross

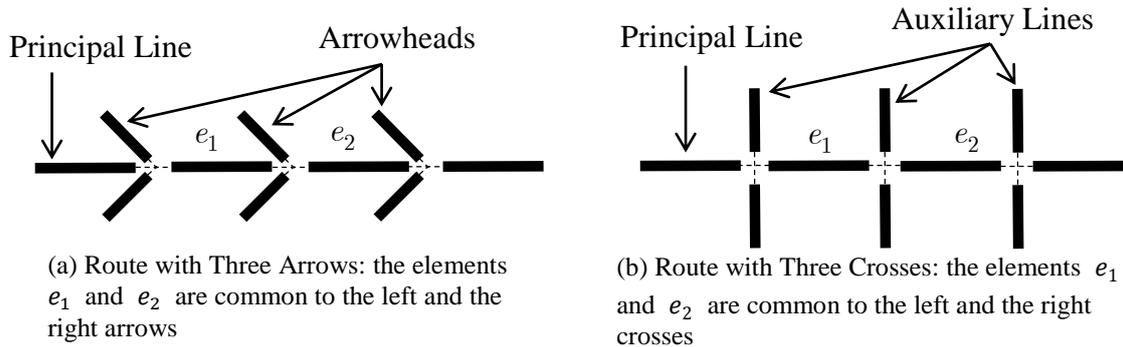


Figure 4.10: Route and Railway Symbols

by x_2 .

The fuzzy if-then rules are given below, while the membership functions are omitted.

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

Rule 2: If x_1 is large, then it is implausible that E is a cross.

Rule 3: If x_2 is large, then it is implausible that E is a cross.

If the similarity for a cluster obtained by the cross classification is larger than a threshold value, the cluster is classified as a cross. Similarly, if the similarity from the arrow classification exceeds a threshold value, the cluster is classified as an arrow.

4.3.3 Route and Railway Classification

Route and railway classifications are conducted by the following way.

- (1) We first detect a sequence, denoted by Q , of clusters such that every two adjacent clusters include a common straight line element (see Figure 4.10).
- (2) If the sequence Q consists of arrows, then the route classification is executed to calculate the similarity, denoted by s_1 , for the sequence Q . If s_1 is larger than a threshold value, the sequence Q is classified as a route.
- (3) If the sequence Q consists of crosses, then the railway classification is executed and we obtain the similarity, denoted by s_2 , for the sequence Q . If s_2 exceeds a threshold value, the sequence Q is classified as a railway.

Fuzzy inference systems are applied to compute the two similarities. To calculate the attribute values for the fuzzy inference systems, we detect the principal line and the arrowheads (or the auxiliary lines) for the sequence Q as follows. If two straight line

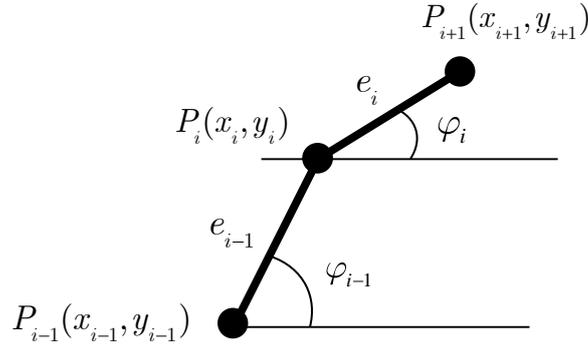


Figure 4.11: Curvature for Digital Curve

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

- (1) The elements e_1 and e_2 are connected at an intersection, P .
- (2) The curvature at the intersection is small.

Let C be a digital curve, and let P_i be a point on C . A curvature κ of C at P_i is then defined as the subtraction $|\varphi_i - \varphi_{i-1}|$ (see Figure 4.11); that is,

$$\kappa = |\varphi_i - \varphi_{i-1}| \quad (4.4)$$

where φ_i is the measure of an angle which is formed between the x -axis and the line segment from P_i to P_{i+1} , and φ_{i-1} is also the measure of an angle which is formed between the x -axis and the line segment from P_{i-1} to P_i [24]. Two adjacent elements e_{i-1} and e_i are merged into a single straight line segment if the curvature between e_{i-1} and e_i is less than a threshold value. After that, we extract the principal line and the arrowheads (or the auxiliary lines) for the sequence Q .

After extracting the principle line and the arrowheads (or the auxiliary lines) for the sequence Q , two fuzzy inference systems are applied to classify the sequence Q as a route or railway. The first fuzzy inference system is for route classification. It has the following two input attributes:

- (1) the standard deviation for the lengths of elements in the principal line; it is denoted by x_1 ;
- (2) the standard deviation for lengths of elements in the arrowheads; it is denoted by

x_2 .

The fuzzy if-then rules for the first fuzzy inference system are denoted below, but the membership functions are omitted.

Rule 1: If x_1 is small and x_2 is small, then it is plausible that Q is a route.

Rule 2: If x_1 is large, then it is implausible that Q is a route.

Rule 3: If x_2 is large, then it is implausible that Q is a route.

The second fuzzy inference system is for railway classification, and the description for the second system is omitted because it is similar to the first system.

Finally, the recognized result for hand-drawn map is saved as SVG and Edel files, the creation method for SVG and Edel documents is the same as subsection 3.6.4 of Chapter 3.

4.4 Experimental Results

This section describes the accuracy for our classification system. Five participants drew 15 maps using pencils and papers. These maps were then captured by using an image scanner; the resolution of the scanner was set to 100 dpi. These digital images were saved as 24-bits bitmap images. The sizes of the images are in 1169×850 pixels. We have measured the accuracy for our classification system using precision, recall, and f -measure. Here, we use N to denote the number of all objects for the symbol, T is used to denote the number of objects correctly classified as the symbol, C denotes the number of objects classified as the symbol, P is used to denote precision, R denotes recall, and F is f -measure. Where,

$$R = \frac{T}{N}, \quad P = \frac{T}{C}, \quad \text{and} \quad F = \frac{2 \cdot R \cdot P}{R + P}.$$

In the 15 maps, there are 50 traffic signals, 15 destinations, 15 departure places, and 40 landmarks. The classification accuracy for our system is summarized in Table 4.1. We can conclude that our system can produce the Edel and SVG documents almost correctly from hand-drawn maps.

Table 4.1: Experimental Results for Shape Recognition

Symbol	<i>N</i>	<i>T</i>	<i>C</i>	<i>P</i>(%)	<i>R</i>(%)	<i>F</i>(%)
Traffic Single	50	50	52	96.2	100	98.1
Destination	15	15	15	100	100	100
Departure Place	15	14	14	100	93.3	96.5
Landmark	40	38	38	100	95.0	97.4
Cross	130	125	125	100	96.2	98.1
Arrow	122	121	121	100	99.2	99.6
Route	15	14	16	87.5	93.3	90.3
Railway	28	28	28	100	100	100

4.5 Usability Evaluation for Our System

4.5.1 Method

We have studied the usability evaluation for our system. Thirteen participants, 12 males and 1 female, aged 20, participated in this investigation; all of them are the third year university students. To evaluate the usability for our system, we selected two common methods for the production of tactile graphics; one is the method to use the software system Edel which assists us to draw diagram images for tactile graphics produced by Braille embossers, and another one is the method using swell papers. In the second method, a diagram image is transferred to a swell paper by a printer and so forth. A swell paper has been coated with thermally foamed microcapsules that respond to irradiation and cause the dark image on the paper to swell, creating the tactile graphic. Microsoft PowerPoint (PPT) was selected to draw a diagram image on a swell paper, because this software system is commonly used in universities in Japan.

Before starting the experiment, we asked the following questions Q1, Q2, and Q3 to all the participants, and the results for the questions Q1 and Q3 are summarized in Figures 4.12 and 4.13. We omit to show the result for the question Q2 because all the participants have no experience in using Edel. All the participants are familiar with using computers and have much experience in using PPT.

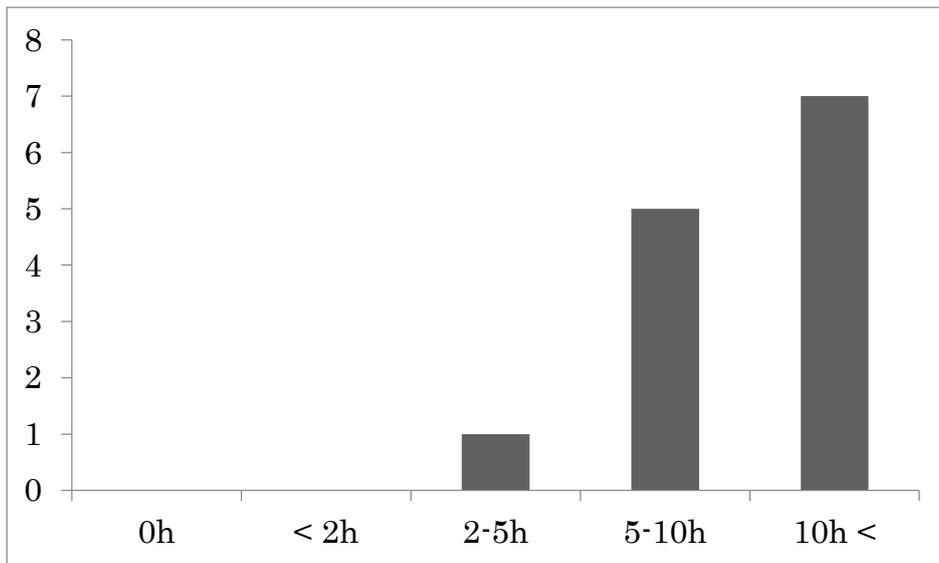


Figure 4.12: Experience in Computers: The Vertical Axis Indicates The Number of Participants.

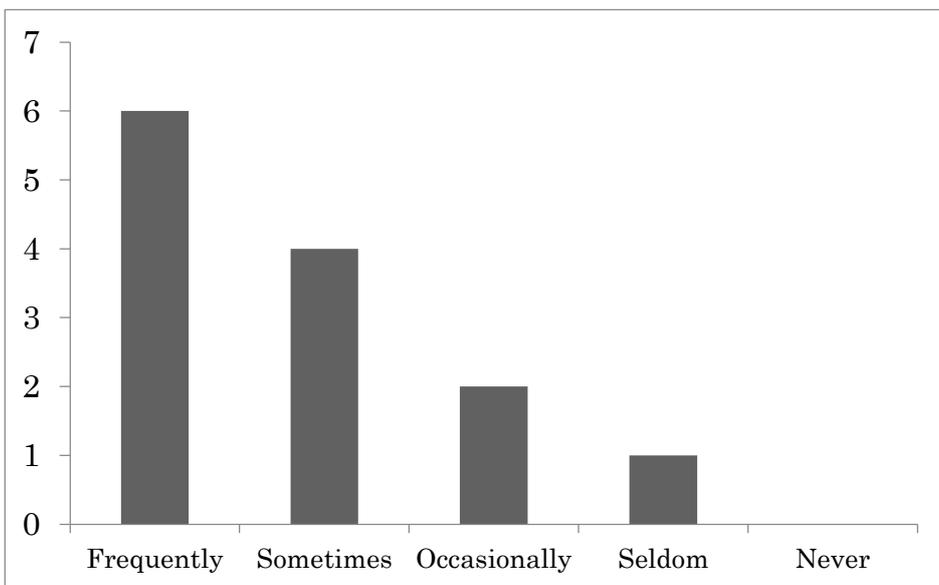


Figure 4.13: Experience in PPT: The Vertical Axis Indicates The Number of Participants.

Q1: How long do you use computers every week?

Q2: Have you ever created figures by using Edel?

Q3: Have you ever created figures by using PPT?

We had two sessions in this investigation. For the first session, a participant created

the map diagram shown in Figure 4.14 (a) - (c); the digital files were produced by using Edel, PPT, and our system. Note that, in our system, a user first draws a map manually using a pencil and paper, then converts the hand-drawn map to a bitmap image using an image scanner, and finally the bitmap image is transformed into the two digital files, the Edel and SVG documents. After conducting all tasks, the participant was asked to answer the following two questions Q4 and Q5.

Q4: Were you able to easily create the digital file(s) for the tactile map?

Q5: Do you want to use this software system to create the digital file for a tactile map?

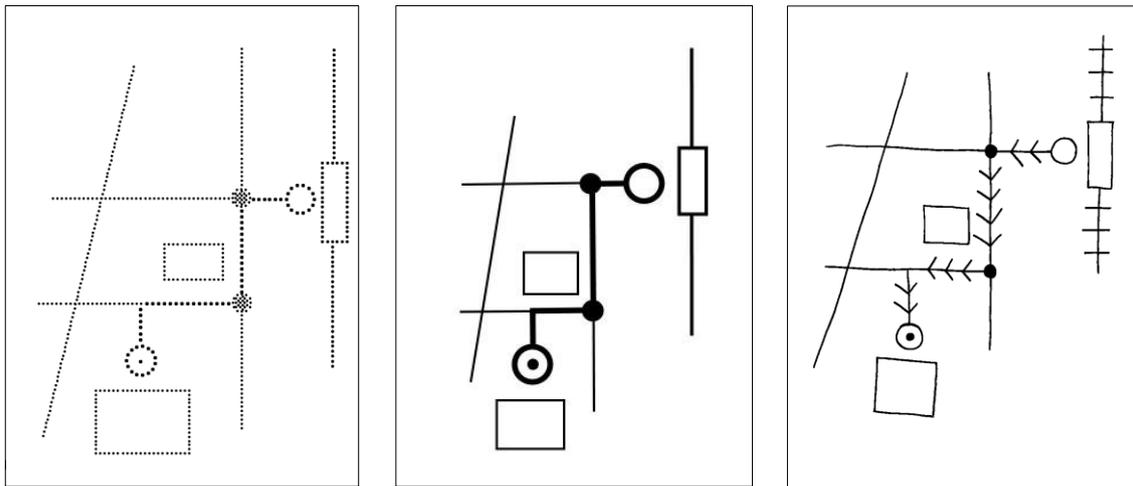
After the first session, the participant had 5 minutes for learning the software systems by himself/herself. The second session followed this learning session. In the second session, the participant conducted similar tasks to the first session, but the maps were those shown in Figure 4.14 (d) - (f). After creating all digital files, the participant was asked to answer two questions Q4 and Q5 again and was also asked to answer the following question Q6.

Q6: Were you able to draw the map as you like?

The orders of the software systems were determined randomly in both the first and the second sessions.

4.5.2 Results

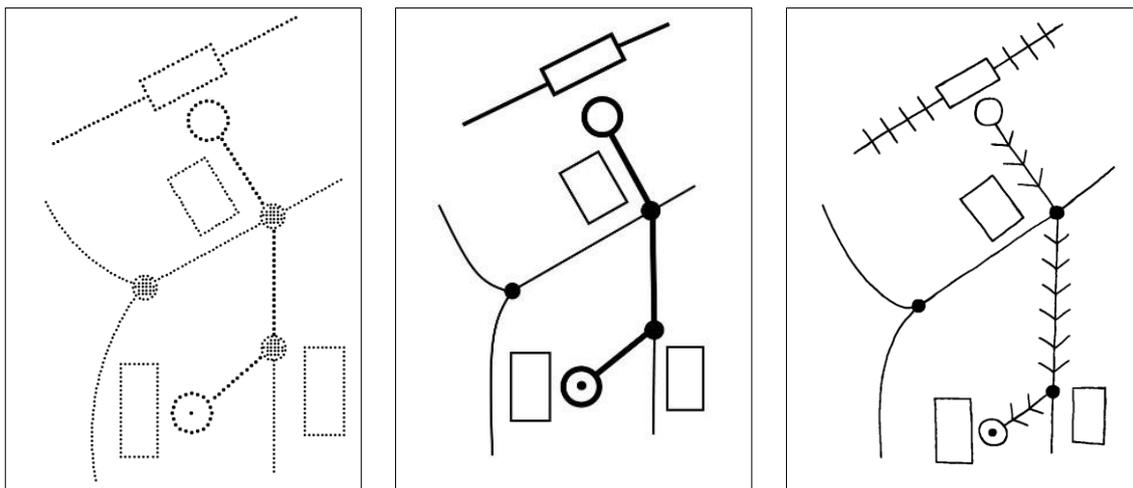
We applied two-way ANOVA to the answers for the 2 questions Q4 and Q5. The first factor, denoted by Factor 1, is the difference in systems, and the second factor, denoted by Factor 2, is the difference in learning. Factor 1 includes three levels, Edel, PPT, and our system, and Factor 2 includes two levels, before learning and after learning. The results from the two-way ANOVA for Q4 shows that there was no significant interactions between Factor 1 and Factor 2 ($F(2, 72) = 0.92, P = 0.40$). Furthermore, there was a significant difference for Factor 1 ($F(2, 72) = 19.26, P < 0.01$), while no significant difference existed for Factor 2 ($F(1, 72) = 1.58, P = 0.21$). The results for Q5 are similar to those for Q4. That is, there was no significant interaction between Factor 1 and Factor 2 ($F(2, 72) = 0.64, P = 0.53$), there was a significant difference for Factor 1



(a) Edel Map for Session 1

(b) PPT Map for Session 1

(c) Hand-drawn Map for Session 1



(d) Edel Map for Session 2

(e) PPT Map for Session 2

(f) Hand-drawn Map for Session 2

Figure 4.14: Maps for Session 1 and Session 2

($F(2, 72) = 14.00, P < 0.01$), and there was no significant difference for Factor 2 ($F(1, 72) = 1.32, P = 0.25$).

We then conducted a multiple comparison test for the answers of the questions Q4 and Q5; we selected Tukey's honestly significant difference (HSD) criterion as the multiple comparison test. For Q4, there was a significant difference between Edel and our system ($P < 0.01$) and there was also a significant difference between PPT and our system ($P <$

0.01); however, there was no significant difference between Edel and PPT (see Figure 4.15). Furthermore, for Q5, there was a significant difference between Edel and our system ($P < 0.01$), and there was also a significant difference between PPT and our system ($P < 0.05$); however, we observed a marginally significant difference between Edel and PPT ($P < 0.1$) (see Figure 4.16).

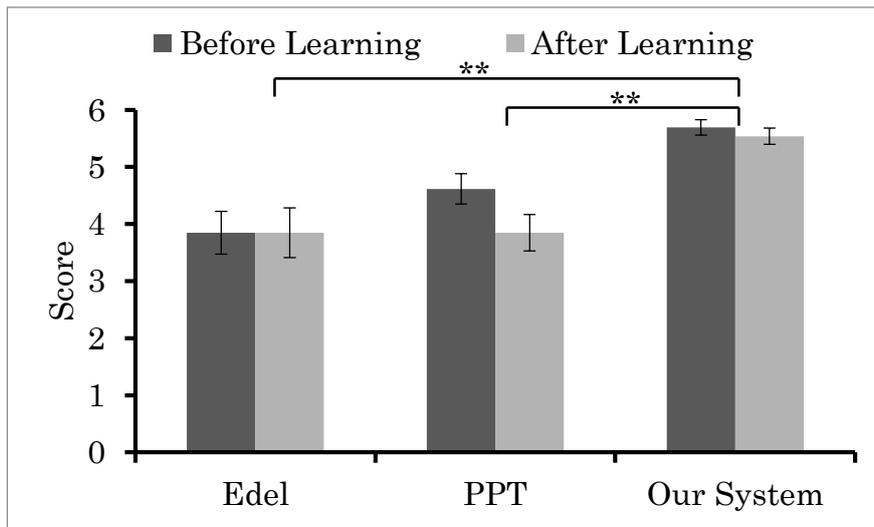


Figure 4.15: Results for Q4: The Average and Standard Error for Each System. (** $P < 0.01$)

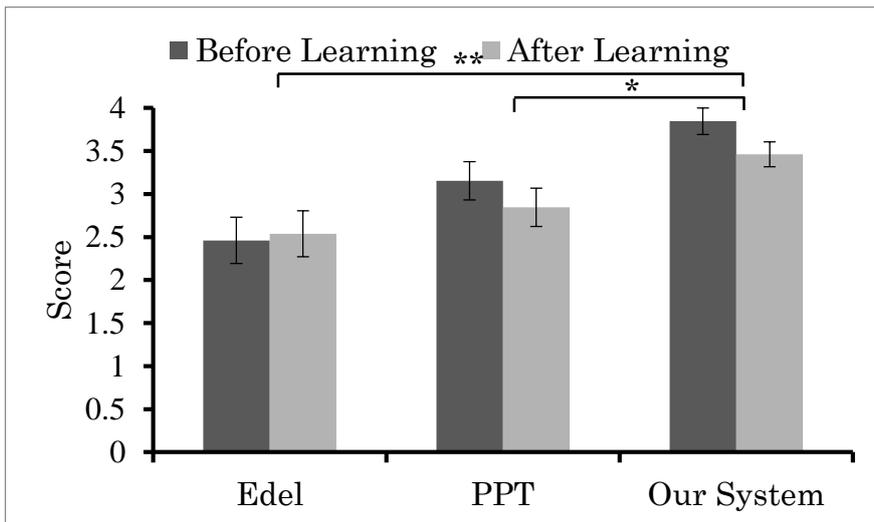


Figure 4.16: Results for Q5: The Average and Standard Error for Each System. (** $P < 0.01$, * $P < 0.05$)

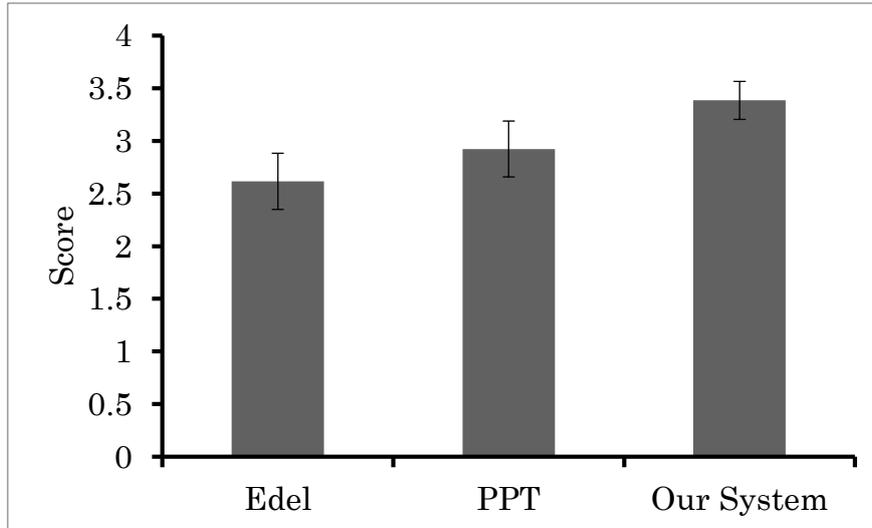


Figure 4.17: Results for Q6: The Average and Standard Error for Each System.

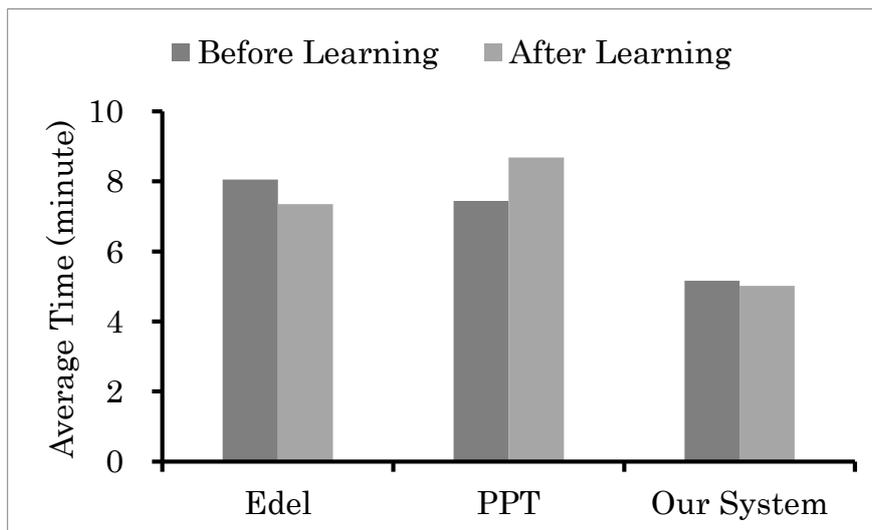


Figure 4.18: Average Time for Creating Digital Files

Lastly, we applied a one-way ANOVA to the answers for the question Q6; the factor of this one-way ANOVA is the difference between systems. The results are shown in Figure 4.17, and the results for the one-way ANOVA show that there was a marginally significant difference ($F(2, 36) = 2.59, P = 0.089$). We then applied Tukey's HSD, and we observed a marginally significant difference between Edel and our system.

For each method, the average time for creating digital files is calculated, as shown in

Figure 4.18, users can use our method to create a digital file in short time, but they need more time while using the other two methods. After the experiment, the participants are asked to give some comments about the three methods for creating tactile graphics. Most participants thought that it is time consuming for creating tactile original file by using both Edel and PPT. It is difficult to rotate, move and remove an object by using Edel. It is a tedious work to define the width of line by using PPT, but it is easy to remove object. It is not easy to draw a curve by using Edel and PPT. Almost all of participants thought that our system is easier for creating a tactile original file.

As a result of the discussion above, we can conclude that hand-drawn method is easier for users to produce tactile graphics than the conventional two methods. Therefore, it is proved that hand-drawn method is a necessary way for producing tactile graphics rather than other computer based operation system. Our system can be used by users even if they do not have much experience of computer operation. Furthermore, map images produced by our system are visually as good as map images produced by Edel and PPT.

4.6 Summary

In this chapter, a computer-aided system was developed for automating translation of hand-drawn figures into tactile graphics. The system includes the following four main procedures: (1) preprocessing, (2) segmentation, (3) pattern recognition based on fuzzy inference, and (4) SVG and Edel documents creation. The preprocessing is applied to an initial hand-drawn map in order to remove noise, and then a binary image is obtained. In the segmentation procedure, the binary image is first skeletonized, and then the skeleton is segmented into elements by eliminating intersections and corner points. Then, a pattern recognition method is applied to recognize symbols. Lastly, SVG and Edel documents are created to save the recognized results. The usability for our system was evaluated by comparing our method with the two conventional systems for creating tactile maps. The results showed that our system is an effective way for creating tactile graphics.

In our fuzzy inference systems, triangular and trapezoidal membership functions are applied to define the fuzzy sets due to their simplicity. The shapes of membership

functions influence the accuracy of recognition results. How to choose an optimal membership function is our future task. In this issue, we do not assume a map includes character strings. However, adding character strings to a map is very important because it can increase the comprehension for the map. In the present work, the types of symbols used in hand-drawn figures are restricted. Developing a system for automating translation of hand-drawn figures with character strings and various symbols is one of our future works.

References

- [1] J. A. Miele, S. Landau, and D. Gilden, "Talking TMAP: Automated Generation of Audio-tactile Maps Using Smith-Kettlewell's TMAP Software", *The British Journal of Visual Impairment*, Vol.24, No.2, pp.93-100, 2006.
- [2] T. Watanabe, T. Yamaguchi, K. Watanabe, J. Akiyama, K. Minatani and M. Miyagi, "Development and Evaluation of a Tactile Map Automated Creation System Accessible to Blind Persons", *IEICE Transactions on Information and Systems*, Vol.J94-D, No.10, pp.1652-1663, 2011.
- [3] TMAPS, <http://tmacs.eng.niigata-u.ac.jp/tmacs-dev>.
- [4] <http://tenpuchizu.gsi.go.jp/shokuchizu/>.
- [5] S. E. Krufka and K. E. Barner, "Automatic Production of Tactile Graphics from Scalable Vector Graphics", in *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility*, pp.166-172, 2005.
- [6] R. E. Ladner, M. Y. Ivory, R. Rao, S. Burgstahler, D. Comden, S. Hahn, M. Renzelmann, S. Krisnandi, M. Ramasamy, B. Slabosky, A. Martin, A. Licenski, S. Olsen and D. Groce, "Automating Tactile Graphics Translation", in *Proceedings of the 7th International ACM SIGACCESS Conference on Computer and Accessibility*, pp.150-157, 2005.
- [7] Edel, <http://www7a.biglobe.ne.jp/EDEL-plus>.
- [8] J. David Eisenberg, *SVG Essentials*, O'Reilly, 2002.
- [9] K. Broelemann, X. Jiang and A. Schwering, "Automatic Street Graph Construction in Sketch Maps", in *Graph-Based Representations in Pattern Recognition*, X. Jiang, M. Ferrer, and A. Torsello (Eds.), Vol.6658 of Lecture Notes in Computer Science, Springer, Berlin, Germany, pp.275-284, 2011.
- [10] L. Burak Kara and T. F. Stahovich, "An Image-based, Trainable Symbol Recognizer for Hand-drawn Sketches", *Computers & Graphics*, Vol.29, No.4, pp.501-517, 2005.
- [11] B. Edwards and V. Chandran, "Machine Recognition of Hand-drawn Circuit Diagrams", in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol.6, pp.3618-3621, 2000.
- [12] P. Sala, "A Recognition System for Symbols of Electronic Components in Hand-written Circuit Diagrams", *CSC 2515-Machine Learning Project Report*, 2004.

- [13] Y. Qiao and M. Yasuhara, "Recovering Dynamic Information from Static Handwritten Images", in *Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, pp.118-123, 2004.
- [14] M. Notowidigdo and R. C. Miller, "Off-line Sketch Interpretation", in *Proceedings of the AAAI Fall Symposium on Making Pen-Based Interaction Intelligent and Natural*, pp.120-126, 2004.
- [15] R. J. Gonzalez and R. E. Woods, *Digital Image Processing*, Pearson Education, Upper Saddle River, NJ, USA, 3rd edition, 2007.
- [16] N. Otsu, "A Threshold Selection Method from Gray-level Histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.9, No.1, pp.62-66, 1979.
- [17] C. Judith Hilditch, "Linear Skeletons from Square Cupboards", in *Machine Intelligence*, Vol.IV, B. Meltzer and D. Michie (Eds.), Elsevier, New York, NY, USA, pp.403-420, 1969.
- [18] S. Hermann and R. Klette, "Global Curvature Estimation for Corner Detection", *Communication and Information Technology Research Technical Report 171*, 2005.
- [19] N. Nain, V. Laxmi, B. Bhadviya and A. Gopal, "Corner Detection Using Difference Chain Code as Curvature", in *Proceedings of the 3rd IEEE International Conference on Signal-Image Technologies and Internet-Based Systems*, pp.821-825, 2007.
- [20] B. Kerautret, J. Lachaud and B. Naegel, "Curvature Based Corner Detector for Discrete, Noisy and Multi-scale Contours", *International Journal of Shape Modeling*, Vol.14, No.2, pp.127-145, 2008.
- [21] K. Wall and P. Danielsson, "A Fast Sequential Method for Polygonal Approximation of Digitized Curves", *Computer Vision, Graphics, & Image Processing*, Vol.28, No.2, pp.220-227, 1984.
- [22] D. Chetverikov, "A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves", in *Computer Analysis of Images and Patterns*, Vol.2756 of Lecture Notes in Computer Science, Springer, Berlin, Germany, pp.746-753, 2003.
- [23] T. Terano, K. Asai and M. Sugeno, *Fuzzy Systems Theory and Its Applications*, Academic Press, Boston, Mass, USA, 2nd edition, 1992.
- [24] N. Ono and R. Takiyama, "On Calculations of Curvature of Sampled Curves", *Technical Report of IEICE*, Vol.IE93-74, pp.7-14, 1993.

Chapter 5

Text Extraction from Natural Scene Images

This chapter proposes a new method for extracting text in natural scene images. First, research motivation is introduced. Then, related research for text extraction from scene images is reviewed. Third, we propose a new algorithm for segmenting an image into homogeneous regions. Fourth, Canny edge detector and SVM classifier are used to extract characters from homogeneous regions. Lastly, our proposed method is performed to extract characters from a set of natural scene images. Accuracy, precision and recall rates of our proposed method is calculated. Experimental results is discussed to conclude this chapter.

5.1 Introduction

According to the statistics in 2010 [1], there are approximately 285 million people suffer visual impairments in the world. Visually impaired people are generally divided into two categories: people with low vision and the completely blind. The former have reduced vision, even when using the best possible corrective lenses; the latter completely lack a perception of form and visual light. The completely blind require some help from, for example, guide dogs or caregivers when walking outdoors; on the other hand, people with low vision tend to walk outdoors alone, and thus they often lose their ways when first visiting a new place. Therefore, developing a walking guide device plays a key role in assisting people with visual impairments, especially people with low vision.

Braille for the feet and traffic lights with acoustic devices are common methods for assisting the visually impaired when walking outdoors. Furthermore, a number of walking

guide devices have been developed to assist the visually impaired in location and navigation. Most of these assistive devices are developed based on application of the following techniques: Global Positioning System (GPS) [2, 3], Radio Frequency IDentification (RFID) [4], and Wireless Fidelity (Wi-Fi) [5]. However, each of these devices has significant limitations. Braille for the feet and traffic lights with acoustic devices require widely laying and installation; GPS has poor resolution in urban settings and is unavailable indoors; RFIDs can only be read at close range, and therefore they are difficult to locate by blind travelers; and Wi-Fi localization requires extensive deployment to ensure complete coverage.

It is important to obtain text information present in natural scenery while a person is walking around. However, such text information is inaccessible to many visually impaired people unless it is represented non-visually, such as with Braille or speech. Therefore, reading text automatically from natural scene images is an important application for assisting visually impaired people. The characteristics of walking guide devices that use image processing are as follows: because such devices are based on software techniques that extract text information from scene images, installing infrastructures might not be required; we can provide them at a low price; and people with visual impairments can carry them easily when walking outdoors. Based on this idea, several aid systems have been developed based on image processing to help visually impaired people read text from sceneries and product labels [6 - 12]. All these systems have limitations for text region extraction because natural scene images usually consist of text and complex backgrounds, such as buildings, trees, window frames and other.

We are now committed to the development of a portable camera-based assistive system to aid visually impaired people read text from natural scenery. To develop such a system, the following techniques are required: wearable camera for capturing images; text region extraction software; off-the-shelf Optical Character Recognition (OCR) software for recognizing and converting printed text into machine encoded text; and Microsoft Speech Software Development Kit (SDK) for transforming the text code into speech for the visually impaired. Figure 5.1 depicts the flowchart of such a system. The aim of this chapter is to develop a methodology for extracting characters from natural scene images.

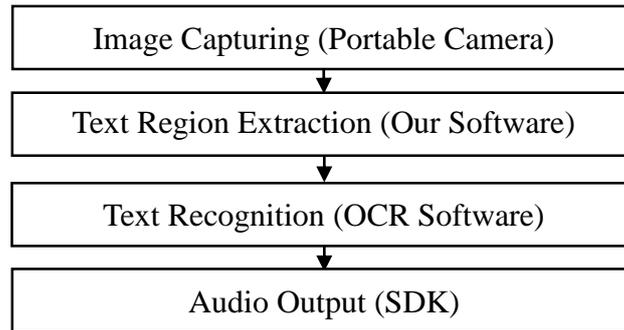


Figure 5.1: Flowchart of an Assistive System

5.2 Related Studies in Text Extraction

There are several studies related to character extraction from scene images. Most research use one of the following three techniques: (1) edge detection algorithms, (2) clustering algorithms for colored pixels, and (3) image transformations such as Fourier transform and Discrete Cosine Transform (DCT).

For methods based on edge detection, the edge detection algorithms extract not only character edges but also edges from non-character objects, such as roadside trees and other. Therefore, removing non-character edges is important for improving the accuracy of the character extraction methods. Methods based on color clustering [13 - 15] often apply the conventional clustering algorithms to those detected regions where character pixels are located. However, extracting character regions correctly is difficult because, for example, there can be many different colors in a scene image. Character extraction methods that use Fourier transform and DCT have been investigated [16 - 18]. In [17] and [18], the authors discussed methods that use DCT because more cue features can be observed in the DCT coefficients of the character regions than non-character regions. However, such features have not been observed clearly; that is, for a scene image with complex backgrounds, it is not always true that the same features in the DCT coefficients for character regions can be observed [19].

Most methods used in past research first detect text regions in an image, and then segment the image into text and background. Their performance relies on the text

detection algorithm and image complexity. Scene text is usually presented on signboards, and the signboard surface has uniform color. The ideal way for extracting characters from signboard regions is if a given captured image can be segmented into homogeneous regions, and then extract characters from such regions. The proposed method is described and discussed in the following sections.

5.3 Preprocessing

A natural scene image, I_{rgb} , is supposed to be a bitmap image based on the RGB color model. A smoothing process, Weighted Median Filter-based Anisotropic Diffusion (WMFAD) [20], is first applied to the original image. Let the R value for a pixel, (x, y) , be $I_r(x, y)$; and WMFAD transforms $I_r(x, y)$ through the following equations:

$$I_r'(x, y) = I_r(x, y) + \lambda \sum_d C_d \cdot \nabla_d I_r(x, y) \quad (5.1)$$

where $I_r'(x, y)$ is the transformed R value; d represents one of the four directions (up, down, left, and right); $\nabla_d I_r(x, y)$ is the difference between $I_r(x, y)$ and the R value for the adjacent pixel in direction d ; λ is a constant; and C_d , a weight of Eq. (5.1), is calculated by the following equation:

$$C_d = \exp(-(\nabla_d \text{median}(Y(x, y)) / K)^2) \quad (5.2)$$

$$Y(x, y) = 0.299 \cdot I_r(x, y) + 0.587 \cdot I_g(x, y) + 0.144 \cdot I_b(x, y) \quad (5.3)$$

In the equations above, $Y(x, y)$ is the brightness for pixel (x, y) , $\text{median}(Y(x, y))$ represents the intensity at pixel (x, y) of the image obtained by performing a median filter to image Y , $\nabla_d \text{median}(Y(x, y))$ is the difference between $\text{median}(Y(x, y))$ and the intensity of the adjacent pixel of $\text{median}(Y(x, y))$ in direction d , and K is a constant.

The G and B values for pixel (x, y) are also transformed by the equations similar to Eq. (5.1). Finally, a smoothed image, I_{WMFAD} , is obtained. I_{WMFAD} is then converted

to a grayscale image I_{gray} through the following equation:

$$I_{gray} = 0.2989 \cdot R + 0.5866 \cdot G + 0.1145 \cdot B \quad (5.4)$$

where R , G , and B are the intensities for red, green, and blue, respectively, of the smoothed image I_{WMFAD} .

5.4 Homogeneous Region Segmentation

Segmentation is the process of partitioning a digital image into multiple pixel regions. The pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic. In this section, a mathematical morphological operator, Toggle Mapping (TM) [21], is introduced to segment a grayscale image into homogenous regions according to the pixel intensity. The procedure is described in the following subsections.

5.4.1 Contrast Enhancement Using Toggle Mapping

TM is a mathematical morphological technique to enhance contrast in grayscale images. The toggle map T is defined by the following equations:

$$T(x, y) = \begin{cases} E(x, y) & \text{if } I_{gray}(x, y) - E(x, y) < D(x, y) - I_{gray}(x, y) \\ I_{gray}(x, y) & \text{if } I_{gray}(x, y) - E(x, y) = D(x, y) - I_{gray}(x, y) \\ D(x, y) & \text{otherwise} \end{cases} \quad (5.5)$$

where E is an eroded image obtained by performing an anti-extensive transformation for morphological erosion of I_{gray} , and D is a dilated image obtained by performing an extensive transformation for morphological dilation of I_{gray} . The structuring element for both erosion and dilation operators is a 3×3 square. (x, y) is the coordinate of the corresponding pixel.

5.4.2 Grayscale Image Smoothing

TM has been used for contrast enhancement, but also for salt and pepper noise

enhancement. Therefore, an average filter is performed on the toggle map T in order to reduce salt and pepper noise. While performing the average filter on T , each output pixel contains the average value in the 3-by-3 neighborhood around the corresponding pixel in image T . A smoothed image, T_{smooth} , is then obtained.

5.4.3 Homogeneous Region Segmentation

This is a simple way to segment a grayscale image into homogeneous regions based on a toggle operator. Such method is defined as follows:

$$H(x, y) = \begin{cases} 1 & \text{if } D(x, y) - E(x, y) \leq th \\ 0 & \text{otherwise} \end{cases} \quad (5.6)$$

where, H takes two values to express a binary image, th is a threshold value, D and E are the dilation and erosion images of input image, respectively. Although, the method of segmentation has been improved by Dorini [22] and Fabrizio [23], in their algorithm the toggle operator is used once to segment an image, and thus the value of thresholds and coefficients is fixed in their toggle operator. However, it is difficult to automatically find an optimal value for their parameters corresponding to a given image, i.e., this is often a serious cause of over segmentation or insufficient segmentation. In order to overcome this limitation, we propose a new algorithm for segmenting a grayscale image into homogeneous regions by applying the toggle operators iteratively. The threshold value is changed in each iteration. The procedure of our algorithm is described in Algorithm 5.1. The notations used in Algorithm 5.1 are presented in Table 5.1.

Figure 5.2 gives an example of result for image segmentation. It shows that the proposed approach can segment homogeneous regions effectively, and with high accuracy.

5.4.4 Candidate Signboard Region Detection

After obtaining the homogeneous regions image, U_{region} , by performing the segmentation procedure, we separate the homogeneous regions into two parts through a simple criterion. One part includes regions whose areas are small, denoted by I_{small} . The other part includes regions whose areas are large, denoted by I_{large} . Signboard areas are

usually large regions. Figure 5.3 (a) and (b) show the separated result of Figure 5.2 (c). In the following procedure, we focus only on the large regions processing.

For the large region image, I_{large} , non-signboard regions are omitted first. Therefore, for a given large region, if it does not satisfy the following condition, it is removed:

$$Th1 < \frac{S_{hole}}{S_{region} + S_{hole}} < Th2 \quad (5.7)$$

where S_{region} is the area of the large region (i.e., the number of pixels for the connected component), and S_{hole} is the total area of holes located inside the large region. $Th1$ and $Th2$ are thresholds. After removing the non-signboard regions, the remaining regions are called candidate signboard regions. Figure 5.3 (c) shows the results after removing the non-signboard regions from Figure 5.3 (b).

Table 5.1: Notations Used in Algorithm 5.1

Parameter	Meaning
$H^{(i)}$	The i -th binary image obtained from equation (5.6).
$CC^{(i)}$	A set of connected components of $H^{(i)}$, $CC^{(i)} = \{C_1^{(i)}, C_2^{(i)}, \dots, C_{n(i)}^{(i)}\}$. Where $n(i)$ is the number of connected components.
$C_j^{(i)}$	The j -th connected component of $CC^{(i)}$.
$NP_{CC}^{(i)}$	The total number of pixels of $CC^{(i)}$.
$MA_{CC}^{(i)}$	The maximum area of connected component in $CC^{(i)}$.
$SD_{C_j}^{(i)}$	The standard deviation of intensity for the j -th connected component in $CC^{(i)}$.
U	A set used for saving homogeneous regions.
$ThSD$	A threshold used for classifying homogeneous regions while calculating standard deviation for connected component.
$ThArea$	A threshold used for judging the total area for all connected components.
$ThRatio$	A threshold used for the ratio between two areas.
$NumStep$	The number of iterations used for segmentation.

Algorithm 5.1: Homogeneous Region Segmentation

1. Equation (5.6) is applied to the smoothed grayscale image, T_{smooth} , here, $th = 1$. A binary image $H^{(1)}$ is then obtained. A labeling processing operator is performed to $H^{(1)}$ in order to obtain a set of connected components, $CC^{(1)} = \{C_1^{(1)}, C_2^{(1)}, \dots, C_{n(1)}^{(1)}\}$. $NP_CC^{(1)}$ and $MA_CC^{(1)}$ are calculated. Let $U \leftarrow \phi$ and $NumStep = 255$.

2. **FOR** $i = 2 : NumStep$ **DO**

Performing equation (5.6) to T_{smooth} , here, $th = i$, then a binary image $H^{(i)}$ is obtained. A set, $CC^{(i)} = \{C_1^{(i)}, C_2^{(i)}, \dots, C_{n(i)}^{(i)}\}$, is obtained by performing a labeling operator to $H^{(i)}$.

FOR $j = 1 : n(i)$ **DO**

IF $SD_C_j^{(i)} > ThSD$, **THEN**

Let $\{C_{j(1)}^{(i-1)}, C_{j(2)}^{(i-1)}, \dots, C_{j(m)}^{(i-1)}\}$ be a subset of $CC^{(i-1)}$, which are covered by $C_j^{(i)}$.

$U \leftarrow U \cup \{C_{j(1)}^{(i-1)}, C_{j(2)}^{(i-1)}, \dots, C_{j(m)}^{(i-1)}\}$.

$CC^{(i)} \leftarrow CC^{(i)} - \{C_j^{(i)}\}$.

END IF

END FOR

$NP_CC^{(i)}$ and $MA_CC^{(i)}$ are calculated, respectively.

IF $\frac{\min(NP_CC^{(i-1)}, NP_CC^{(i)})}{\max(NP_CC^{(i-1)}, NP_CC^{(i)})} > ThRatio$ and $MA_CC^{(i)} < ThArea$, **THEN**

BREAK.

END IF

END FOR

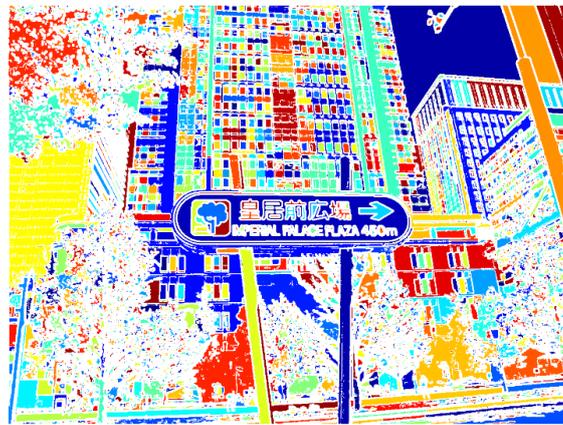
3. Update U by performing the union operator of set theory, and output the image, U_{region} , which includes the connected components of U .



(a) Original Natural Scene Image



(b) Grayscale Image



(c) Labeling Image for Homogeneous Region

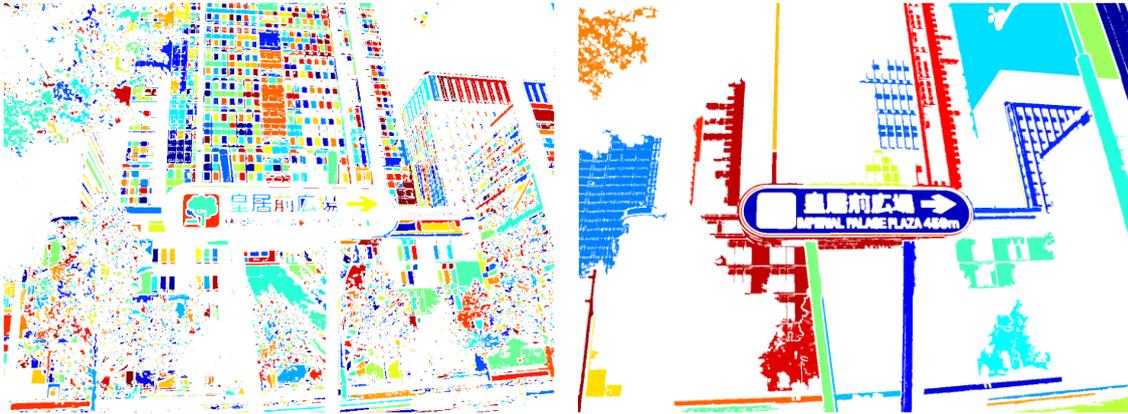
Figure 5.2: An Example for Natural Scene Image Segmentation

5.5 Character Detection from Candidate Signboard Regions

A character in text is usually constructed with uniform color that contrasts significantly with the background. Therefore, in this section, the edges of character components are first detected from candidate signboard regions. Candidate single characters are then detected according to the layout and relationships between the edge components. Finally, a Support Vector Machine (SVM) classifier is applied to classify characters and non-characters.

5.5.1 Edge Detection

Canny edge detector [24] is used for finding the edges of objects from grayscale image.



(a) Labeling Image for Small Regions

(b) Labeling Image for Large Regions



(c) Labeling Image for Candidate Signboard Regions

Figure 5.3: Homogeneous Regions Classification

The Canny edge detector produces a thinning edge image.

For a given candidate signboard region, the following procedure is performed in order to find the character edges:

- (1) Finding the bounding box (i.e. circumscribed rectangle) and convex hull region of the candidate signboard region.
- (2) Segmenting the region from grayscale image corresponding to the convex hull region.
- (3) Applying Canny edge detector to find the edge components inside the segmented gray region.

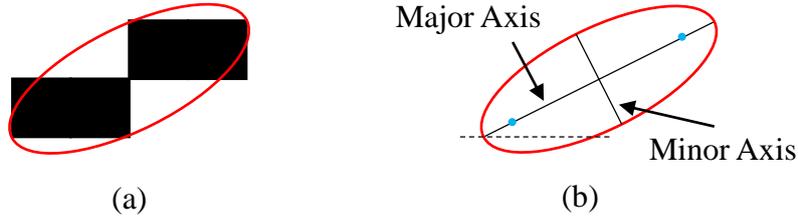


Figure 5.4: Major Axis and Minor Axis of Connected Component

- (4) Performing labeling process on the Canny edge image to find the connected components. The bounding box of each connected edge component is detected.
- (5) Eliminating the connected components that are too small, too large, and too long by calculating the following properties:

$$V_1 = Height_{com} \cdot Width_{com} \quad (5.8)$$

$$V_2 = \frac{\max(Height_{com}, Width_{com})}{\max(Height_{sign}, Width_{sign})} \quad (5.9)$$

$$V_3 = \frac{MajorAxisLength_{com}}{MinorAxisLength_{com}} \quad (5.10)$$

where $Height_{com}$ and $Width_{com}$ are the height and width of the bounding box of the connected component, respectively. $Height_{sign}$ and $Width_{sign}$ are the height and width of the bounding box of the candidate signboard region, respectively. $MajorAxisLength_{com}$ and $MinorAxisLength_{com}$ are the lengths (in pixels) of the major axis and minor axis of the ellipse that has the same second-moments as the connected component region, respectively. Figure 5.4 shows an example of the major axis and minor axis of connected component. A connected component is small if $V_1 < T_1$ holds; it is large if $V_2 > T_2$; and it is long if $V_3 > T_3$. T_i ($i = 1, 2, 3$) are thresholds. Figure 5.5 shows an example of the character edge element detection.

5.5.2 Single Character Detection



(a) Candidate Signboard Region



(b) Grayscale Region Covered by Convex Hull of Signboard Region



(c) Canny Edges inside Convex Hull

Figure 5.5: Character Edge Detection

A character usually consists of two or more strokes, for example Chinese characters and Japanese characters. Therefore, a single character has a plural number of edges/boundaries detected by Canny edge detector, as shown in Figure 5.6. In this subsection, connected edge components are grouped in order to form a single character.

Let e_1, e_2, \dots, e_p be the connected components that are obtained after performing the procedure described in the previous subsection. The following merging processes are then applied to merge the components, e_1, e_2, \dots, e_p , based on their circumscribed rectangles (i.e., bounding boxes). Therefore, let R_1, R_2, \dots, R_p be the rectangles that circumscribe e_1, e_2, \dots, e_p , respectively. Then, two rectangles, R_i and R_j , are merged into a single rectangle if one of the following three conditions holds:

- (1) **Completely Overlapped Rectangles Merging:** If one of (R_i, R_j) is included in the other one of (R_i, R_j) , R_i and R_j are merged into the single rectangle that circumscribes R_i and R_j .
- (2) **Partially Overlapped Rectangles Merging:** In the case where R_i and R_j partially



(a) Labeling Image of Canny Edge Components



(b) Labeling Image for Candidate Single Characters



(c) Labeling Image after Removing Small Characters

Figure 5.7: Single Character Merging

merging process, each rectangle circumscribes a plural number of edge components that form a single character. We remove such characters if the size of their circumscribed rectangles is less than 40×40 pixels. Figure 5.7 shows an example of single character detection. In the following discussion, R'_1, R'_2, \dots, R'_q denote the circumscribed rectangles of the detected single characters.

5.5.3 Character Classification

In the previous subsection, the merging results includes not only single characters but also non- characters. Therefore, it is necessary to remove the non-characters. Now we focus on the regions $\{R^1, R^2, \dots, R^q\}$ of I_{gray} corresponding to R'_1, R'_2, \dots, R'_q . We found that, there are usually apparent difference in luminance between the characters and their background. This implies that the shape of the intensity histogram of the character area is bimodal. On the other hand the shape of the non-character areas is often unimodal or multimodal, as shown in Figure 5.8. Therefore, a SVM classifier will be introduced to classify the characters and non-characters based on these features.

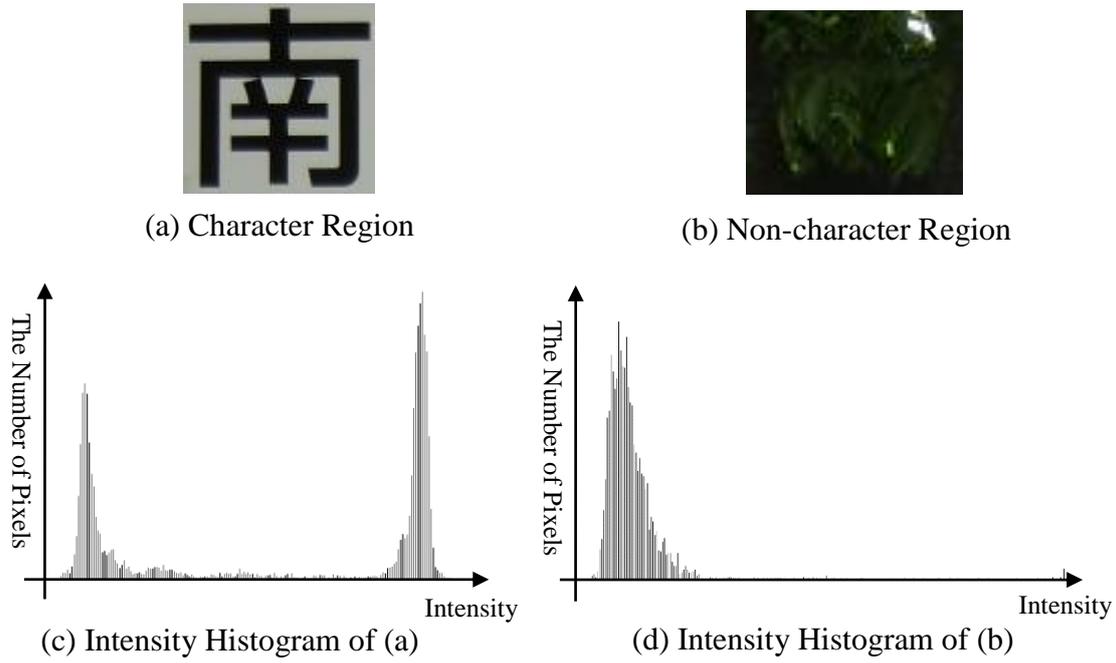


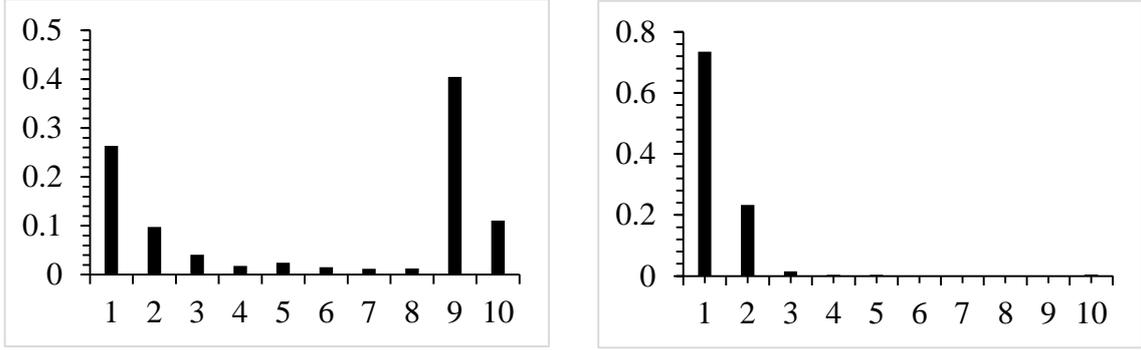
Figure 5.8: Intensity Histograms for Character and Non-character

SVM is a learning method based on margin maximization principle. SVM performs binary classification by finding optimal separating hyperplane in the feature space. Suppose that a set of training examples $\{(x_i, y_i)\}_{i=1}^N$ are given, the SVM classify the input x based on the following function:

$$f(x) = \sum_{i=1}^N \alpha_i K(x, x_i) - b \quad (5.14)$$

where $K(x, y)$ is a kernel function which defines the inner product in the feature space. Coefficients α_i s are non-zero only for the subset of the input data called support vectors. The performance of SVM depends on the kernel. In this paper, we use LIBSVM [25] with Radius Basis Function (RBF) kernel to classify the features. We build a training database containing 4745 samples of characters and 2938 negative samples. The testing database is composed of 1582 samples of characters and 1469 samples of other patterns. All trainings were performed by the tools provided by [26] and [27]. Finally, the parameter $cost$ and γ were decided as 890 and 2, respectively.

Before performing the classification, the intensity histogram should be normalized. For



(a) Normalized Histogram for Figure 5.8 (c) (b) Normalized Histogram for Figure 5.8 (d)

Figure 5.9: Histogram Normalization

a given region R^i , we use $h(x)$ to denote the number of pixels for intensity value x , where $x \in \{0, 1, 2, \dots, 255\}$. $S_{max} = \max\{x | 0 < h(x)\}$ and $S_{min} = \min\{x | 0 < h(x)\}$ denote the maximum and minimum intensities of $\{x | 0 < h(x)\}$. We first divide the interval $[S_{min}, S_{max}]$ into 10 subintervals and the subinterval length denoted by w .

$$w = \left\lfloor \frac{S_{max} - S_{min} + 1}{10} \right\rfloor \quad (5.15)$$

Note that $\lfloor x \rfloor$ is the maximal integer but not larger than real x . In the following, we normalize the histogram of intensity.

(1) If $(S_{max} - S_{min} + 1)/10$ is an integer,

$$h^*(i) = \frac{1}{N} \sum_{j=0}^{w-1} h(S_{min} + w \cdot (i-1) + j) \quad (\text{for } i = 1, 2, \dots, 10) \quad (5.16)$$

(2) If $(S_{max} - S_{min} + 1)/10$ is not an integer,

$$k = (S_{max} - S_{min} + 1) \% 10$$

$$h^*(i) = \frac{1}{N} \sum_{j=0}^w h(S_{min} + (w+1) \cdot (i-1) + j) \quad (\text{for } i = 1, 2, \dots, k) \quad (5.17)$$

$$h^*(i) = \frac{1}{N} \sum_{j=0}^{w-1} h(S_{min} + (w+1) \cdot k + w \cdot (i-k-1) + j) \quad (\text{for } i = k+1, \dots, 10)$$

where, $N = \sum_{x=0}^{255} h(x)$, $a \% b$ means a mod b .

Figure 5.9 shows the normalized histograms corresponding to Figure 5.8.

5.6 Experiment and Results

5.6.1 Experimental Images

150 images of various text scenes such as signboards, traffic signs, shop names and more, are used in our experiment. All the original images captured with a size of about 1000×1500 pixels, their format are RGB24 bitmap format. In order to provide a wide range of real life scenarios, images are captured with different compact digital cameras at different angles, positions and under variable lighting and weather conditions. Figure 5.10 shows some examples used in our experiment. Table 5.2 shows the experimental environment.

Table 5.2: Experimental Environment

OS	Microsoft Windows7 Enterprise
CPU	Intel Core i7 2.93GHz
Memory	8GB
Programming Language	MATLAB

5.6.2 Evaluation Results of Single Character Detection

In our experiment, the size of detected single characters is larger than 40×40 pixels, and this value is determined by a character recognition experimental result. A commercial OCR software based on C language library is applied to recognize the characters detected from signboards in scene images; the results showed that almost all the characters are recognized correctly if their size is larger than 40×40 pixels, but many incorrect recognitions if the character size is less than 40×40 pixels. The optimal value of the thresholds used in our method is determined by training some sample images that are

different from the experimental images. In the following, precision, recall and f -measure are calculated to compare the accuracy of the two methods: our proposed method and the method proposed by Matsuda et al. [13].

Precision and recall can be calculated by the following equations:

$$precision = \frac{R}{N} \quad (5.18)$$

$$recall = \frac{R}{C} \quad (5.19)$$

where R is the number of single characters that have been correctly extracted; N is the number of extracted results (i.e., the results may consist of not only characters, but also incorrect results); and C is the number of all characters we want to find. F-measure is then the weighted harmonic mean of precision and recall, and is calculated by the following equation:

$$F = \frac{1}{\frac{\alpha}{precision} + \frac{1-\alpha}{recall}} \quad (5.20)$$

the parameter α is set to 0.5 in order to give equal weights to the precision and recall.

Because of the evaluated images used in Matsuda's method are different from our experimental images, our image data has been divided into three groups to do the comparison. Group 1 (G1) includes 100 good images, that is, the signboards and characters are clear. Group 2 (G2) includes 50 bad images, that is, the signboards or characters are distortion or not clear. Group 3 (G3) includes the total 150 images. We use these three groups of images to evaluate the accuracy of character extraction. Table 5.3 shows the results for single character detection by our methods compared to Matsuda's result.

It is clear that our proposed method gained a much higher accuracy than Matsuda's method if the experimental images are good, although a low accuracy for bad images. From the experimental results, our method may has a higher average accuracy than Matsuda's method through performing a random image.

Table 5.3: Accuracy of Character Extraction

	C	N	R	Recall (%)	Precision (%)	F-measure (%)
Matsuda Method	--	--	--	78.8	82.5	80.6
Our Method (G1)	1470	1426	1344	91.43	94.25	92.8
Our Method (G2)	753	640	493	65.5	77.0	70.8
Our Method (G3)	2223	2066	1837	82.6	88.9	85.7

5.6.3 Discussion

In our character extraction method, we first perform the image segmentation in order to detect signboard regions. At this stage, most noise, such as trees and other non-homogeneous areas, are removed. After that, characters are extracted from homogeneous regions. Therefore, we can obtain a high accuracy for character extraction if the scene images are good. Matsuda's method extract character strings by processing the entire image based on the following features: Noise Attribute Thresholding (NAT), color information, and edge distribution. For a scene image with complex background, Matsuda's method detected not only the character edges but also non-character edges, and the edges of characters often connect to the edges of non-characters. Furthermore, a scene image usually includes so many colors, even the same character is presented in different colors.

In our method, the effectiveness of signboard region detection directly affects the accuracy of characters extraction. The signboard regions cannot be segmented completely due to the following reasons: 1) the surface of signboard is corroded, e.g., Figure 5.10 (g); 2) shadow exists on the signboard, e.g., Figure 5.10 (h); 3) reflective effect, e.g., Figure 5.10 (i). Figure 5.11 shows the segmented homogeneous regions that correspond to Figure 5.10.

5.7 Summary

This chapter proposed a new method for character extraction from natural scene



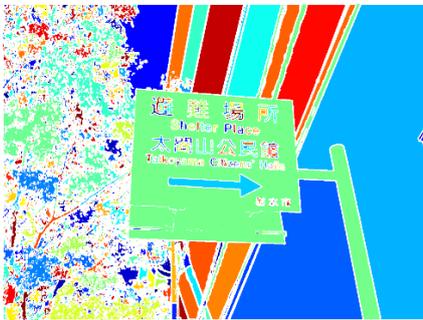
Figure 5.10: Examples of Scene Images

images based on the methodology of image segmentation and SVM. The proposed method was tested on different images, and the results showed that our method can be an effective way for improving the accuracy of character detection. However, the results indicated that the characters were extracted with low accuracy due to the presence of shadows or even corroded signboards.

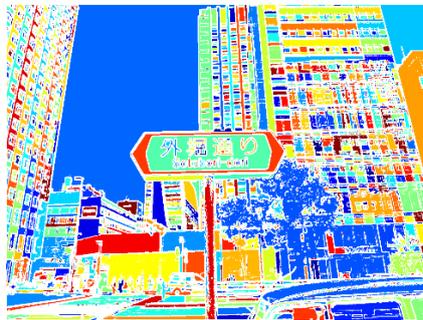
In this chapter, a new algorithm is proposed to segment homogeneous regions from scene images. In order to improve the accuracy of character extraction results, in the near

future, we will introduce another SVM classifier to our method in order to classify signboard regions and non-signboard regions. Fuzzy inference also will be applied to extract character strings.

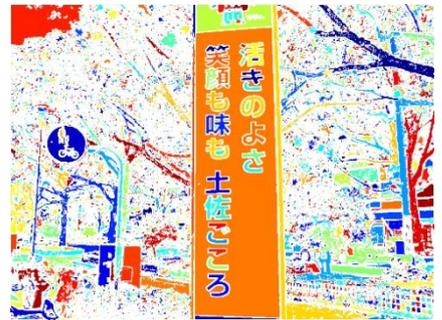
Because of limited time, this chapter did not evaluate the processing time of our method. Therefore, in the near future, our image database will first be expanded with thousands of new and already captured images; second, we will compare our proposed algorithm with other similar methods and analyze the performance in terms of computing time, execution complexity and accuracy.



(a) Homogeneous Regions of Fig.5.10 (a)



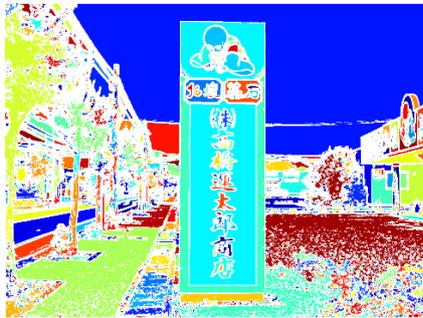
(b) Homogeneous Regions of Fig.5.10 (b)



(c) Homogeneous Regions of Fig.5.10 (c)



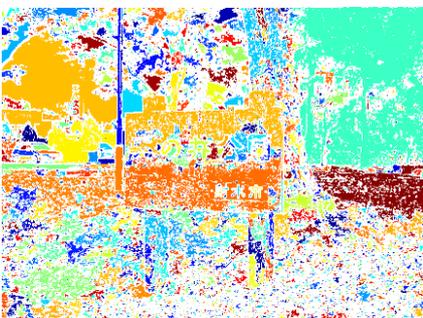
(d) Homogeneous Regions of Fig.5.10 (d)



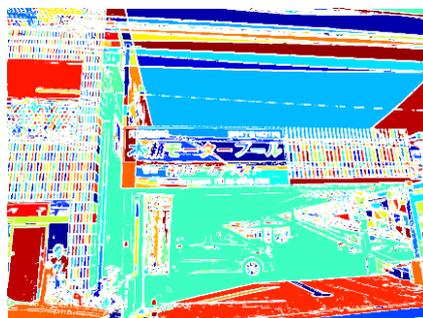
(e) Homogeneous Regions of Fig.5.10 (e)



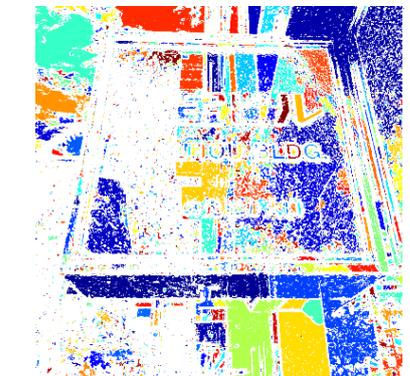
(f) Homogeneous Regions of Fig.5.10 (f)



(g) Homogeneous Regions of Fig.5.10 (g)



(h) Homogeneous Regions of Fig. 5.10 (h)



(i) Homogeneous Regions of Fig.5.10 (i)

Figure 5.11. Labeling Image for Homogeneous Regions

References

- [1] D. Pascolini and S. P. Mariotti, "Global Estimates of Visual Impairment: 2010", *British Journal Ophthalmology*, Vol.96, No.5, pp.614-618, 2012.
- [2] <http://www.senderogroup.com>
- [3] J. Ishikawa and Y. Hyodo, "Developing an Accessible GPS System for the Blind", *Technical Report of IEICE*, Vol.104, No.554, pp.51-56, 2005.
- [4] <http://www.mlit.go.jp/seisakutokatsu/jiritsu/>
- [5] A. M. Ladd, K. E. Bekris, A. P. Rudys, D. S. Wallach and L. E. Kavraki, "On the Feasibility of Using Wireless Ethernet for Indoor Localization", *IEEE Transaction On Robotics and Automation*, Vol.20, No.3, pp.555-559, 2004.
- [6] N. Tanaka and M. Okudaira, "A Study on Image Processing for Action Assistance of Visually Impaired Humans Using a Camera Equipped Cellular Phone", *Technical Report of The Institute of Image Information and Television Engineers*, Vol.33, Issue 6, pp.173-176, 2009.
- [7] N. Ezaki, K. Kiyota, B. T. Minh, M. Bulacu and L. Schomaker, "Improved Text-Detection Methods for a Camera-based Text Reading System for Blind Persons", in *Proceedings of the 8th International Conference on Document Analysis and Recognition*, pp.257-261, 2005.
- [8] S. M. Hanif and L. Prevost, "Texture Based Text Detection in Natural Scene Images: A Help to Blind and Visually Impaired Persons", in *Proceedings of Conference & Workshop on Assistive Technologies for People with Vision & Hearing Impairments Assistive Technology for All Ages*, M.A. Hersh (Ed.), 2007.
- [9] M. Pazio, M. Niedzwiecki, R. Kowalik and J. Lebiedz, "Text Detection System for The Blind", in *Proceedings of the 15th European Signal Processing Conference*, pp.272-276, 2007.
- [10] U. Kalai selvi and J. Anish Kumar, "Camera based Assistive Text Reading System using Gradient and Stroke Orientation for Blind Person", *International Journal of Latest Trends in Engineering and Technology*, Vol.4, Issue 1, pp.325-330, 2014.
- [11] O. Foong and N. Razali, "Signage Recognition Framework for Visually Impaired People", in *Proceedings of International Conference on Computer Communication*

and Management, pp.488-492, 2011.

- [12]C. Yi, Y. Tian and A. Arditi, “Portable Camera-based Assistive Text and Product Label Reading from Hand-held Objects for Blind Persons”, *IEEE/ASME Transactions on Mechatronics*, 2014.
- [13]Y. Matsuda, S. Omachi and H. Aso, “String Detection from Scene Images by Binarization and Edge Detection”, *IEICE*, D, Vol.J93-D, No.3, pp.336-344, 2010.
- [14]H. Hase, M. Yoneda, M. Sakai and H. Maruyama, “Consideration of Color Segmentation to Extract Character Areas from Color Document Images”, *IEICE*, D-II, Vol.J83-D-II, No.5, pp.1294-1304, 2000.
- [15]K. Ashida, H. Nagai, M. Okamoto, H. Miyao and H. Yamamoto, “Extraction of Characters from Scene Images”, *IEICE*, D-II, Vol.J88-D-II, No.9, pp.1817-1824, 2005.
- [16]Y. Liu, T. Yamamura, N. Ohnishi and N. Sugie, “Extraction of Character String Regions from a Scene Image”, *IEICE*, D-II, Vol.J81-D-II, No.4, pp.641-650, 1998.
- [17]S. Saitoh, H. Goto and H. Kobayashi, “Analysis and Comparison of Frequency Features for Scene Text Detection”, *Technical Report of IEICE*, PRMU2004-128, pp.31-36, 2004.
- [18]D. Crandall, S. Antani and R. Kasturi, “Extraction of Special Effects Caption Text Events from Digital Video”, *International Journal on Document Analysis and Recognition*, Springer-Verlag, Vol.5, pp.138-157, 2003.
- [19]M. Inami, *A Character String Extraction Method from Scene Images Using Edge Detection and Color Clustering*, Master Thesis of Intelligent Systems Design Engineering, Toyama Prefectural University, Japan, 2012.
- [20]M. Gabbouj, P. Haavisto and Y. Neuvo, “Recent Advances in Median Filtering”, *Communication, Control and Signal Processing*, Vol.II, E. Arikan (Ed.), Elsevier Science Publishers B, Ankara, Turkey, pp.1080-1094, 1990.
- [21]J. Serra, “Toggle mappings”, *From Pixels to Features*, J.C. Simon (Ed.), North-Holland, Elsevier, pp. 61-72, 1989.
- [22]L. B. Dorini and N. J. Leite, “A Scale-space Toggle Operator for Morphological Segmentation”, in *Proceedings of the 8th International Symposium on Mathematical Morphology*, Vol.1, pp.101-112, 2007.

- [23] J. Fabrizio, B. Marcotegui and M. Cord, “Text Segmentation in Natural Scenes Using Toggle-Mapping”, in *Proceedings of the 16th IEEE International Conference on Image Processing*, pp.2349-2352, 2009.
- [24] J. Canny, “A Computational Approach to Edge Detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.PAMI-8, No.6, pp.679-698, 1986.
- [25] CC. Chang and CJ. Lin, LIBSVM: A Library for Support Vector Machines, 2014. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. detailed documentation (algorithms, formulae, ...) can be found in <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.ps.gz>
- [26] CW. Hsu, CC. Chang and CJ. Lin, “A Practical Guide to Support Vector Classification”, 2010, <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>
- [27] CC. Chang and CJ. Lin, LIBSVM Tools, <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools>

Chapter 6

Conclusions

This chapter summarizes the contributions of this dissertation for assisting visually impaired people to access visual information. We also present the main conclusions of the research done in tactile graphics production and scene text detection. Finally, we conclude this dissertation by pointing out some possible future works.

6.1 Summary

This dissertation focuses on the development of technologies for aiding visually impaired people to access visual information.

In chapter 1, assistive technologies for supporting visually impaired people to perform daily living tasks were first reviewed. Research motivation of this dissertation was then discussed, finally our objectives were proposed.

The definition and necessity of tactile graphics were introduced in chapter 2. In this chapter, guidelines for designing tactile graphics also be summarized.

In chapter 3, a method of graph recognition was proposed in order to assist the translation of mathematical graphs into tactile graphics. This chapter mainly discussed a method for extracting and classifying graph elements. More than 30 mathematical graphs were applied to examine the accuracy of our method, and the result showed that our method can work well for many examples.

Chapter 4 discussed the development of a computer-aided system for automating

production of tactile graphics from hand-drawn maps. The techniques for hand-drawn recognition was first discussed. 15 samples of hand-drawn figures were then applied to evaluate the accuracy of our method, and the result gained a high accuracy. Lastly, the most important usability issue was evaluated to show that hand-drawn method is an effective way for producing tactile graphics. Although all of participants were familiar with computer operation, most of them though that hand-drawn method is useful and necessary for people to produce tactile graphics.

In chapter 5, a method for extracting characters from scene images was proposed. In this method, a new algorithm was first designed to segment an image into homogeneous regions. In the new algorithm, toggle mapping operator was applied iteratively to detect homogeneous regions. From the observation of segmentation results, our proposed algorithm can worked effectively and with high accuracy. Characters were then detected from such homogeneous regions. In this stage, Canny edge detector, merging rules, and SVM were applied to extract single characters. Finally, our proposed method was tested on different images, and the result showed that the method can improve the accuracy for character extraction. Results, however, indicated that the character extraction with low accuracy due to the presence of shadows or even corroded signboards.

6.2 Future Works

This dissertation has contributed many works in tactile graphic production and scene text extraction, but there are also many tasks need to be done as follows.

In this dissertation, we did not evaluate the readability of tactile graphics produced by our methods, testing the readability is one of our future endeavors.

In Chapter 3, the accuracy for extracting graph elements, especially for the broken line elements and solid line elements, were not so high. Therefore, in the near future, we aim to overcome the drawbacks for unsuccessful recognized graphs.

In Chapter 4, we assumed a map do not include characters and character strings. However, adding character strings to a map is very important because it can increase the

comprehension for the figure. In the present work, the types of symbols used in hand-drawn maps are restricted. Developing a system for automating translation of hand-drawn figures with character strings and various symbols is one of our future challenges.

In order to improve the accuracy for character extraction in Chapter 5, in the near future, Support Vector Machine (SVM) will be introduced to our method in order to classify signboard regions and non-signboard regions. A fuzzy inference also will be applied to extract character strings.

Because of time limitation, Chapter 5 did not evaluate the processing time of our method. Therefore in the near future, our image database will be first expanded with thousands of new and already captured images. The same image data then will be used to do the experiment in order to compare our method with other similar methods in terms of computing time and accuracy.

In the future, we will improve the methodologies for hand-drawn figure recognition in order to aid the production of tactile graphics corresponding to the figures in physics, chemistry, and biology textbooks.

Acknowledgements

I would like to show appreciation for my advisor, Professor Noboru Takagi, for his guidance and patience throughout the work that led to this thesis. I would also like to thank my friends and the members of the Takagi Lab for inspiration and feedback.

I would like to express my sincere gratitude to the advisor of Dalian Jiaotong University, Professor Weixing Bi who gave me the opportunity to study abroad.

I am grateful for the love and continuous support from my parents and elder sister.

List of Publications

This dissertation has led to the following communications:

Publications in Peer-reviewed Journals

1. Jianjun Chen and Noboru Takagi, “Development of a Method for Extracting and Recognizing Graph Elements in Mathematical Graphs for Automating Translation of Tactile Graphics”, *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, Vol.26, No.2, pp.593-605, 2014.
2. Jianjun Chen and Noboru Takagi, “Development of a System to Assist Automatic Translation of Hand-drawn Maps into Tactile Graphics and Its Usability Evaluation”, *Advances in Fuzzy Systems, Hindawi*. Vol.2014, Article ID 357380, 11 pages, 2014.
3. Jianjun Chen and Noboru Takagi, “Mathematical Morphology Based Image Segmentation and Character String Extraction Using Fuzzy Inference”, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol.19, No.4, pp.544-554, 2015.

Contributions to Peer-reviewed International Conferences (Primary Author)

1. Jianjun Chen, Ryoji Nagaya and Noboru Takagi, “Development of a Method for Extracting Graph Elements from Mathematical Graphs”, in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp.2921-2926, 2012.
2. Jianjun Chen, Ryoji Nagaya and Noboru Takagi, “An Extraction Method of Solid Line Graph Elements in Mathematical Graphs for Automating Translation of Tactile Graphics”, in *Proceedings of The 6th International Conference on Soft*

Computing and Intelligent Systems and The 13th International Symposium on Advanced Intelligent Systems, pp.422-427, 2012.

3. Jianjun Chen and Noboru Takagi, “A Pattern Recognition Method for Automating Tactile Graphics Translation from Hand-drawn Maps”, in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp.4173-4178, 2013.
4. Jianjun Chen and Noboru Takagi, “A Method to Extract Character Strings from Scene Images by Eliminating Non-Character Strings”, in *Proceedings of The 9th International Conference on Signal-Image Technology & Internet-Based Systems*, pp.848-853, 2013.
5. Jianjun Chen and Noboru Takagi, “A Homogeneous Region Based Methodology for Text Extraction from Natural Scene Images”, in *Proceedings of The 4th International Conference on Informatics, Electronics & Vision*, 2015.

Contributions to Peer-reviewed International Conferences (Coauthor)

1. Ryoji Nagaya, Jianjun Chen and Noboru Takagi, “An Extraction and Classification Method of Broken Line Elements in Mathematical Graphs for Automating Translation of Tactile Graphics”, in *Proceedings of The 6th International Conference on Soft Computing and Intelligent Systems and 13th International Symposium on Advanced Intelligent Systems*, pp.216-221, 2012.
2. Noboru Takagi and Jianjun Chen, “A Broken Line Classification Method of Mathematical Graphs for Automating Translation into Scalable Vector Graphic”, in *Proceedings of The 43rd International Symposium on Multiple-Valued Logic*, pp.71-76, 2013.
3. Noboru Takagi and Jianjun Chen, “Development of a Computer-aided System for Automating Production of Tactile Maps and Its Usability Evaluation”, *World Automation Congress*, pp.213-218, 2014.
4. Noboru Takagi and Jianjun Chen, “Character String Extraction from Scene Images by Eliminating Non-character Elements”, in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp.3685-3690, 2014.

Contributions to National Workshops (Primary Author)

1. 陳建軍, 長屋竜治, 高木昇, “数学グラフからの成分抽出方法について”, 第28回ファジィシステムシンポジウム講演論文集, pp.85-90, 2012.
2. 陳建軍, 高木昇, “手書き地図の触図翻訳システム開発”, 第29回ファジィシステムシンポジウム講演論文集, pp.248-253, 2013.
3. 陳建軍, 高木昇, “視覚障害者のための触地図作成支援システムの開発”, 第30回ファジィシステムシンポジウム予稿集, pp.502-507, 2014.
4. 陳建軍, 高木昇, “数学的モルフォロジーとファジィ推論を用いた情景画像からの文字列抽出方法”, 第31回ファジィシステムシンポジウム講演論文集, pp.521-527, 2015.
5. 陳建軍, 高木昇, “数学グラフの触図翻訳システム開発について”, 電子情報通信学会技術研究報告, Vol.112, No.475, WIT2012-99, pp.305-310, 2013.
6. 陳建軍, 出町駿明, 高木昇, “触地図作成支援のための手書き地図の自動翻訳手法について”, 電子情報通信学会技術研究報告, Vol.113, No.77, WIT2013-5, pp.25-30, 2013.

Contributions to National Workshops (Coauthor)

1. 長屋竜治, 陳建軍, 高木昇, “数学グラフからの破線グラフ成分の抽出方法について”, 第28回ファジィシステムシンポジウム講演論文集, pp.91-96, 2012.
2. 佐々木隆行, 陳建軍, 高木昇, “非文字成分除去に着目した情景画像からの文字列検出法”, 第29回ファジィシステムシンポジウム講演論文集, pp.264-269, 2013.
3. 佐々木隆行, 陳建軍, 高木昇, “Toggle Mapping を用いた情景画像からの看板領域抽出”, 第30回ファジィシステムシンポジウム予稿集, pp.476-479, 2014.
4. 長屋竜治, 陳建軍, 高木昇, “触図作成支援システムのための数学グラフ認

識手法について”, 多値論理研究ノート, 第 35 卷, 第 18 号, pp.18-1~18-6, 2012.

5. 佐々木隆行, 陳建軍, 高木昇, “数学的モルフォロジーとエッジ抽出を用いた情景画像からの看板文字抽出”, 多値論理研究ノート, 第 37 卷, 第 6 号, pp.6-1~6-6, 2014.