THE UNIVERSITY OF CALGARY

AN ADAPTIVE SYSTEM FOR ACCURATE HANDWRITTEN NUMERAL RECOGNITION

by

Xiaohua Fan

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

CALGARY, ALBERTA

MAY, 1993

ISBN 0-315-88507-6

Name Xiaohua Fan

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

Electronics and Electrical — 0544 — U·M·I

SUBJECT TERM — SUBJECT CODE

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS
Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION
General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523
Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS
Language
- General 0679
- Ancient 0289
- Linguistics 0290
- Modern 0291

Literature
- General 0401
- Classical 0294
- Comparative 0295
- Medieval 0297
- Modern 0298
- African 0316
- American 0591
- Asian 0305
- Canadian (English) 0352
- Canadian (French) 0355
- English 0593
- Germanic 0311
- Latin American 0312
- Middle Eastern 0315
- Romance 0313
- Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY
Philosophy 0422
Religion
- General 0318
- Biblical Studies 0321
- Clergy 0319
- History of 0320
- Philosophy of 0322

Theology 0469

SOCIAL SCIENCES
American Studies 0323
Anthropology
- Archaeology 0324
- Cultural 0326
- Physical 0327

Business Administration
- General 0310
- Accounting 0272
- Banking 0770
- Management 0454
- Marketing 0338

Canadian Studies 0385
Economics
- General 0501
- Agricultural 0503
- Commerce-Business 0505
- Finance 0508
- History 0509
- Labor 0510
- Theory 0511

Folklore 0358
Geography 0366
Gerontology 0351
History
- General 0578
- Ancient 0579
- Medieval 0581
- Modern 0582
- Black 0328
- African 0331
- Asia, Australia and Oceania 0332
- Canadian 0334
- European 0335
- Latin American 0336
- Middle Eastern 0333
- United States 0337

History of Science 0585
Law 0398
Political Science
- General 0615
- International Law and Relations 0616
- Public Administration 0617

Recreation 0814
Social Work 0452
Sociology
- General 0626
- Criminology and Penology 0627
- Demography 0938
- Ethnic and Racial Studies 0631
- Individual and Family Studies 0628
- Industrial and Labor Relations 0629
- Public and Social Welfare 0630
- Social Structure and Development 0700
- Theory and Methods 0344

Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES
Agriculture
- General 0473
- Agronomy 0285
- Animal Culture and Nutrition 0475
- Animal Pathology 0476
- Food Science and Technology 0359
- Forestry and Wildlife 0478
- Plant Culture 0479
- Plant Pathology 0480
- Plant Physiology 0817
- Range Management 0777
- Wood Technology 0746

Biology
- General 0306
- Anatomy 0287
- Biostatistics 0308
- Botany 0309
- Cell 0379
- Ecology 0329
- Entomology 0353
- Genetics 0369
- Limnology 0793
- Microbiology 0410
- Molecular 0307
- Neuroscience 0317
- Oceanography 0416
- Physiology 0433
- Radiation 0821
- Veterinary Science 0778
- Zoology 0472

Biophysics
- General 0786
- Medical 0760

EARTH SCIENCES
Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES
Environmental Sciences 0768
Health Sciences
- General 0566
- Audiology 0300
- Chemotherapy 0992
- Dentistry 0567
- Education 0350
- Hospital Management 0769
- Human Development 0758
- Immunology 0982
- Medicine and Surgery 0564
- Mental Health 0347
- Nursing 0569
- Nutrition 0570
- Obstetrics and Gynecology 0380
- Occupational Health and Therapy 0354
- Ophthalmology 0381
- Pathology 0571
- Pharmacology 0419
- Pharmacy 0572
- Physical Therapy 0382
- Public Health 0573
- Radiology 0574
- Recreation 0575
- Speech Pathology 0460
- Toxicology 0383

Home Economics 0386

PHYSICAL SCIENCES
Pure Sciences

Chemistry
- General 0485
- Agricultural 0749
- Analytical 0486
- Biochemistry 0487
- Inorganic 0488
- Nuclear 0738
- Organic 0490
- Pharmaceutical 0491
- Physical 0494
- Polymer 0495
- Radiation 0754

Mathematics 0405
Physics
- General 0605
- Acoustics 0986
- Astronomy and Astrophysics 0606
- Atmospheric Science 0608
- Atomic 0748
- Electronics and Electricity 0607
- Elementary Particles and High Energy 0798
- Fluid and Plasma 0759
- Molecular 0609
- Nuclear 0610
- Optics 0752
- Radiation 0756
- Solid State 0611

Statistics 0463

Applied Sciences

Applied Mechanics 0346
Computer Science 0984

Engineering
- General 0537
- Aerospace 0538
- Agricultural 0539
- Automotive 0540
- Biomedical 0541
- Chemical 0542
- Civil 0543
- Electronics and Electrical 0544
- Heat and Thermodynamics 0348
- Hydraulic 0545
- Industrial 0546
- Marine 0547
- Materials Science 0794
- Mechanical 0548
- Metallurgy 0743
- Mining 0551
- Nuclear 0552
- Packaging 0549
- Petroleum 0765
- Sanitary and Municipal 0554
- System Science 0790

Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY
General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled, "An Adaptive System for Accurate Handwritten Numeral Recognition", submitted by Xiaohua Fan in partial fulfillment of the requirements for the degree of Master of Science.

Dr. J. Gu, Supervisor
Dept. of Electrical & Computer Engineering

Dr. J.W. Haslett
Dept. of Electrical & Computer Engineering

Dr. F.N. Trofimenkoff
Dept. of Electrical & Computer Engineering

Dr. X. Mao
Dept. of Mechanical Engineering

Date: Aug. 31, 1993

ABSTRACT

The use of computers for the recognition of handwritten characters is one of the main trends in office automation. The most troublesome problem is the great variation among characters. Most of the published structural character recognition algorithms depend on qualitative curve feature extraction and a predescribed model. Failures may occur when they are applied for the recognition of unconstrained handwritten characters. This thesis presents a new approach for handwritten numeral recognition. A both qualitative and quantitative feature extraction technique, an adaptive structural classification algorithm and an iterative spur removal strategy are proposed. A prototype recognition system has been implemented. For the isolated handwritten numerals in the Standard Handwritten Character Database, a recognition rate of 98% with 0% rejection rate has been achieved.

Acknowledgement

I would like to express my sincere appreciation to Dr. Jun Gu, my supervisor, for his kind assistance, guidance and encouragement during this entire work.

I thank Dr. X.F. Huang for his suggestions and discussion in areas related to neural networks, pattern recognition and image processing.

To

those who love me

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

Computer approaches for handwritten numeral recognition are widely used in office automation and many engineering applications. In post offices, tons of mail have to be classified by hand every day. A computer aided postal code recognition techniques can significantly reduce mail processing time. In tax offices, millions of tax forms are processed each year. The use of computer handwritten recognition could save the manpower currently required to process tax forms. Other applications can be found in banks, libraries, and even in helping visually impaired people.

Compared with machine printed character recognition, the problem of handwritten character recognition is more difficult because of the irregularity of handprinted characters. The differences in writing habits, tools and conditions result in a handwritten character with many variations in style, scale, thickness of stroke and noise on the image. A character may be written in many different ways. At the same time, different written characters may look so similar that people make mistakes in recognition.

The purpose of this thesis is to investigate the problem of automatic handwritten numeral recognition. A new high accuracy handwritten numerical recognition approach is proposed. The significant aspects of the study include:

(a) development of a new approach of accurate structural feature representation and extraction,

(b) proposal of an approach for pre-classification and the corresponding neural networks for the adaptive and non-parametric recognition of hand-written numerals, and

(c) presentation of a new technique of iterative spur removal to improve the recognition accuracy.

1.2 REVIEW

The problem of automatic character recognition has been extensively attempted for many years. Various approaches have been explored. An early attempt was implemented by Grimsdale *et al* in 1958 [34]. In their method, the input character patterns were described in terms of the length and slope of straight line segments and the length and curvature of curved segments. The patterns were then compared with those of the pre-stored prototypes to reach a proper decision about the identity of the input character.

Eden [21] proposed the *analysis-by-synthesis* method. He claimed that all Latin characters can be formed by 4 strokes, namely, hump, bar, hook and loop. Based on this idea, Cox *et al* [17] presented two main groups of grammar-like rules to deal with the variability in the type fonts. Yoshida and Eden [89] then proposed a generative process to extract a stroke sequence from the input pattern and used a look-up dictionary of strokes to effect the recognition.

A *split-and-merge* algorithm for the polygonal approximation of a character for numeral recognition was suggested by Pavlidis and Ali [66]. Feng and Pavlidis [26] utilized a feature generation technique for syntactic pattern recognition by approximating character boundaries with polygons and then decomposing the polygons based on the concavity.

Shridhar and Badrelin [75] presented a two-stage character recognition algorithm which used Fourier and topological descriptors in the recognition of numerals. Later on, they applied a new set of topological features derived from a global description of the character. The recognition system they developed includes a syntactic classifier to analyze the topological structure of the patterns.

A system for classification by relaxation matching of handwritten zip-code numbers was described by Lam and Suen [51]. The system comprised of a feature extractor, a structural classifier, and a relaxation classifier. The feature extractor decomposed the skeleton of the character into geometrical primitives. The structural classifier identified the majority of the samples. The relaxation classifier then classified the rest of the data. Baptist and Kulkarni [4] employed a multilevel approach to process the visual information and recognize handwritten characters.

A pattern description and generation method for structural characters was reported by Nagahashi and Nakatsuyama [62] in which a character was regarded as a composite pattern consisting of several simpler subpatterns and described in terms of the subpatterns using three kinds of positional relationships among them.

Although handwritten character recognition has been studied extensively, none of the algorithms mentioned above gives a description of character structure in detail. Also, a threshold based noise spur removal strategy has been widely used in those algorithms. This noise spur removal method removes strokes shorter than a threshold. Thus failure may occur when:

1. different characters have a similar structure,
2. noise spurs are longer than the threshold, and
3. meaningful strokes are shorter than the threshold.

This thesis reports a new quantitative structure description to represent the structure of characters so that minor differences between similar characters can be detected. The iterative spur removal strategy interacts between the feature extraction and recognition stages, just as a human being does. Spurs are removed not on the bases of any threshold, but on the current available knowledge of the character to be recognized. By doing so, most of the problems introduced by threshold based algorithms can thus be resolved.

1.3 THE ORGANIZATION OF THE THESIS

This thesis is organized as follows.

Chapter 2 gives a general review of handwritten recognition techniques, with emphasis on the structural methods. The basic principle of the structural methods, the representative algorithms, and a comparison of various handwritten recognition techniques are included.

Chapter 3 pays a special attention to the structural recognition methods. A survey of the currently used methods of smoothing, thinning, feature extraction, and matching is presented.

Chapter 4 presents the general description of a new high accuracy recognition algorithm. The quantitative feature description and extraction, adaptive structure classification and iterative spur removal are introduced separately.

Chapter 5 describes the implementation of the new handwritten character recognition system.

Chapter 6 contains a summary of this research.

CHAPTER 2

HANDWRITTEN CHARACTER RECOGNITION

There are many algorithms for handwritten character recognition. Generally, these algorithms can be classified into three categories: structural, statistical, and neural network based methods.

2.1 STRUCTURAL METHODS

A structural method, i.e. a syntactic, linguistic, grammatical method, is one of the major methods used for handwritten character recognition. A structural method defines a set of elementary forms of which a character is composed. A character is represented by the elementary forms and the way they are assembled.

The simplest structural descriptions of a pattern consist of ordered sequences of elementary components, an indication of the presence or absence of a component, and an indication of the relative positions of the components. A comparison of two such descriptions provides a measurement of the extent to which the corresponding patterns resemble each other and, consequently, is a method for recognition. Usually, a finite set of letters, or alphabet, is used to represent a finite set of elements. The relationship between elements is represented by simple juxtaposition or the concatenation of these letters, i.e., a string. For example, if we have an alphabet:

$$X = \{a, b, c\} \tag{2.1}$$

then a string on X:

$$x = \text{``}cabbc\text{''} \tag{2.2}$$

can be a structural representation of a certain pattern.

After the elements and their relationship are extracted from a character and are represented as a string, the similarities or the distances between the extracted string and those pre-classified and stored in a data-base are calculated. The category with which the extracted string is the most similar is considered to be the category for the string and therefore the result of the recognition.

Fig. 2.1 shows a simple example of the concept of the structural pattern recognition. Suppose we expect to characterize two families of shapes among a set of black and white images: equilateral triangles and squares, respectively. By tracing a contour, it can be seen that the triangle in Fig. 2.1 can be characterized as follows:

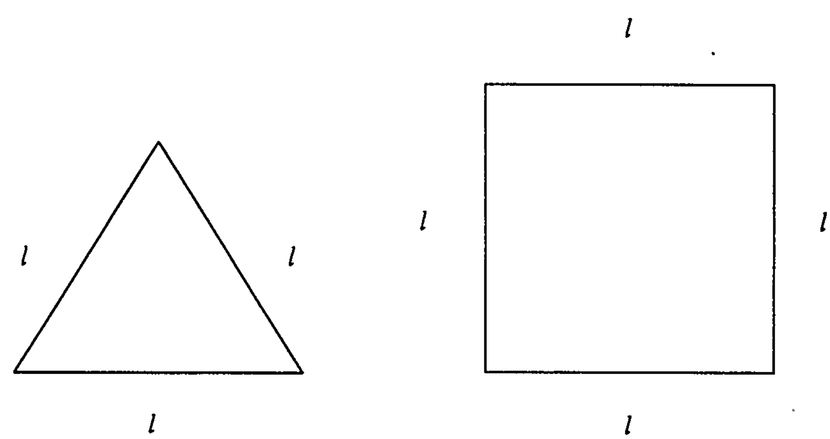


Figure 2.1. An example of the structural method.

Starting from the left bottom corner, the boundary consists of a horizontal segment of length l followed by an oblique segment of length equal to l followed by another oblique segment of length equal to l whose end-point is the starting point of the first segment. If the straight line segment with length l can be defined as letter "a," then the triangle can be expressed as string "aaa."

A similar description can also be given for the square shown in Fig. 2.1, and it is "aaaa."

These two strings can be stored in a data base. When an unknown pattern is encountered, its string expression is extracted and compared with those pre-stored in the data base. The result of the comparison tells what kind of shape the input pattern is. In practice, more complex string expressions and more sophisticated classification algorithms are employed.

2.2 STATISTICAL METHODS

Another classical method for pattern recognition is based on the statistical study of measurements made on the objects to be recognized. As a result of studying the distribution of these measurements in a metric space and the statistical properties of the classes, a decision on recognition can be taken. Such methods are usually based on hypotheses, explicitly stated or not, concerning the statistical description of families of related objects in the representation.

It is assumed in this approach that the measurements can be expressed as a vector X = $(x_1, \ldots, x_n)$ in the space R^n. Assuming a teaching set is available, i.e., a set of vectors for which the classes to which they belong are known, the problem can then be stated as follows: given an unknown vector obtained as a result of measurements on some pattern, find a class to which the pattern should be assigned. A sample problem is given in Fig. 2.2. The question is: does the point $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2\}$ belong to class 1, class 2, or class 3?

To solve the problem, first, consider the process of learning from a teaching set. The situation described here is an idealized one. It is assumed that we know in advance the number of classes and the decision that certain points – those of the teaching set – belong to certain classes. Practical problems can not always be expressed so neatly. The classes that might be thought, from observation, to be the most 'natural,' may

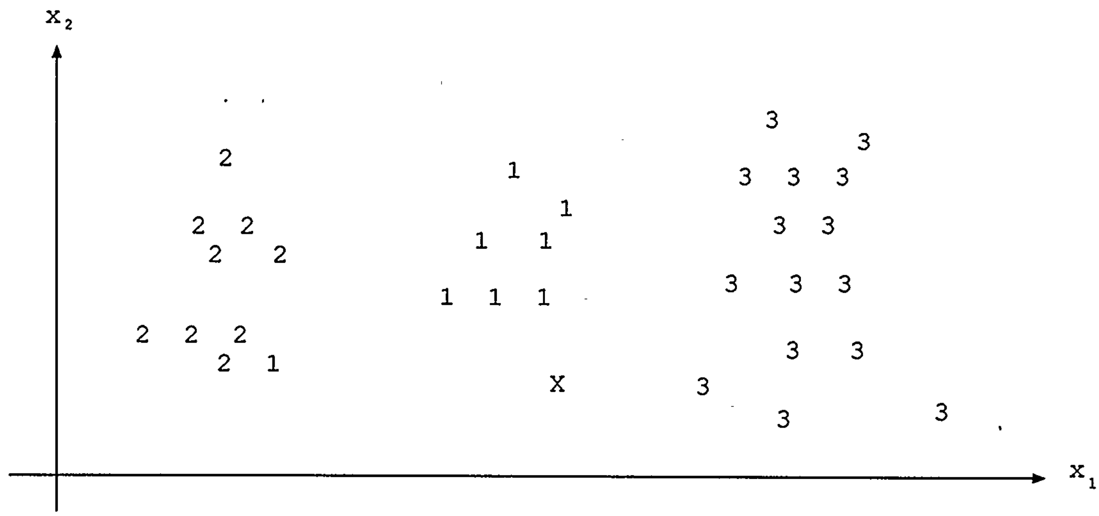


Figure 2.2. An example of the statistical method.

turn out to be ill suited to the chosen representation space. They may overlap or provide a poor description of the teaching set. Fully or partly automated learning methods may prove to be of help in redefining these classes so as to improve the situation, and here pattern recognition has benefited from research on data analysis.

After the question of learning has been settled, we return to the problem of arriving at a decision. There are two major types of methods, called *non-parametric* methods and *parametric* methods, respectively. A non-parametric method defines the boundaries of the different classes in the representation space so that a series of simple tests are sufficient to assign an unknown point to one of these classes. The parametric method constructs a model (e.g. Gaussian) for the distribution of the points of each class and on this basis decides to which class the unknown point has the greatest probability of belonging.

A simple non-parametric method involves finding the hyperplanes that give the 'best' separation of the classes in the teaching set. The decision process is then simplified to calculating a series of scalar products. This method is called *linear separation.* Much work has been done on determinating the equations of the hyperplanes and also on evaluating the theoretical probabilities of the errors resulting from the use of the method. Another non-parametric method is called *nearest neighbors* method, in which the unknown point is assigned to the class of its nearest neighbor in the teaching set. This method gives good results statistically but often requires long times for calculation. Various methods have been proposed for speeding up the process.

In a parametric method, assumptions concerning the statistical properties of the classes are made to minimize the mean errors of a classification. The decision can be optimal if these assumptions hold for the teaching set as well as for the representation space.

2.3 NEURAL NET CLASSIFIERS

Artificial neural net (ANN) models or simply "neural nets" go by many names such as connectionist models, parallel distributed processing models, and neuromorphic systems. Whatever the name, all these models attempt to achieve good performance via a dense interconnection of simple computational elements. In this respect, an artificial neural net structure is based on the present understanding of biological nervous systems. Neural net models have great potential in areas such as image recognition where many hypotheses are pursued in parallel, and high computation rates are required. Instead of performing a program of instructions sequentially as is done in a von Neumann computers, neural net models explore many competing hypotheses simultaneously using massively parallel nets composed of many computational elements connected by links with variable weights.

A neural network has several features:

Learning

ANNs can modify their behavior in response to their environment - learn from experience. It is here that the neural nets can perform functions beyond the capacity of rule based, conventional systems. In understanding handwriting, the problem with conventional approaches is either that the rules are difficult to find, or that the number of rules becomes large. This makes the learning ability the most attractive feature of ANNs.

Generalization

Once trained, an ANN's response can be insensitive to the minor variations in its input. This ability to see the pattern despite noise and distortion is vital to pattern recognition in a real-world environment. This inherent ability to generalize is due to the structure of ANNs rather than any program, hypothesis, or threshold.

Applicability

ANNs are computationally complete. That is, given an appropriate neural net and training, there is no computational task that cannot be performed by neural nets. Therefore, they become the preferred techniques for a large class of pattern recognition tasks that conventional computers do poorly.

Neural networks have been recently applied to the character recognition problem. Burr [12] used back-propagation networks to recognize the shadow code of hand printed digits. Cun et al [18] employed the same network to match input digits with 49 pre-stored templates. A goal seeking neurons were applied to classify vector patterns by Filho [27]. Guyon [36] used a multi-layer, feed-forward network for on-line character recognition.

2.4 DISCUSSION

The structural approach is concerned more with the intrinsic characteristics of a pattern rather than with its metric properties. This feature is significant in handwritten character recognition. Due to different writing styles, the images of a character could be variable. These variations cause difficulties with statistical methods based on metric features of characters. However, the basic structures of the elements and their interconnections in a character do not change much. It is the structure which represents the primary, intrinsic and natural information of a character.

The use of formal language-theoretic models to represent patterns is the main drawback of the structural approach [31]. Patterns are natural entities which cannot strictly obey the mathematical constraints set by formal language theory. Imposing a strict rule on the pattern structure is not particularly applicable to character recognition, where the interclass variations are infinite. Furthermore, structural approaches do not pay much attention to feature extraction but are sensitive to spur noises which introduce irregular variations into the structure.

The statistical approach has a solid and general mathematical foundation and has been successfully applied to machine-printed character recognition. However, the statistical approach has the disadvantage of ignoring the varied nature of the measurements made on the patterns and treating them all in an abstract manner. As a result, structural information about the inter-connections in complex handwritten characters cannot be handled efficiently by statistical pattern recognition techniques.

ANNs have an intrinsic ability to adapt to the various deformations in handwritten characters. ANNs store the learned knowledge distributively by weights instead of rules. No regulation base and matching algorithm is required. Since most ANNs are analog, they are more suitable to the continuous variation of handwritten character than the classical discrete string representation. However, most ANNs learn using images of characters and this implies that large networks are required [18]. Long training and recognition time and poor convergence are characteristics of large networks. Image based recognition also introduces limitations in recognizing characters with large variations and deformations.

2.5 SUMMARY

Compared to the statistical approach, the structural method concentrates more on the intrinsic and natural features of a character rather than operating on the metric features only. This property of the structural approach makes it more suitable than the statistical approach, especially for handwriting recognition. The drawbacks of the structural approach are its language model representation of the character structure, its feature description and extraction, and its sensitivity to spur noise. Neural networks are adaptive, distributive and analog. These features of neural networks provide compensation for the drawbacks of the structural approach.

The combination of structural recognition and neural network will take advantage of both methods. The structural information of handwritten numerals can be represented and classified. Complex rule building and matching are eliminated, and small neural networks can be expected. Adaptation and learning abilities are improved. The combination of the structural approach and the neural network is a promising direction for handwritten character recognition.

CHAPTER 3

STRUCTURAL HANDWRITTEN RECOGNITION

Structural handwriting recognition is usually divided into four steps: smoothing, thinning, feature extracting, and matching. The system structure is shown in Fig. 3.1.

3.1 SMOOTHING

Feature extraction or shape measurement in character recognition can be misleading if the images are not preprocessed. Noise-contaminated character images produce extra spurs, dots, and holes in their thinned images and may result in mislabeling by the recognition logic. Hence an image preprocessing or smoothing process is usually included in structural recognition systems. A noise-contaminated image and a thinned image are shown in Fig. 3.2 and Fig. 3.3.

3.1.1 ANALYSIS OF NOISE

Noises in the raster scanned images can be classified into the following six types (refer to Fig. 3.2):

Type 1: *Linear artifact.* This kind of noise introduces artifacts of various lengths in different directions, and will produce false branches during a thinning process.

Type 2: *Isolated dot.* This kind of noise usually arises from stains or dirt on the analog input or is caused by pen-up at the end of a stroke. Extra false strokes will be produced in the thinning process by such dots.

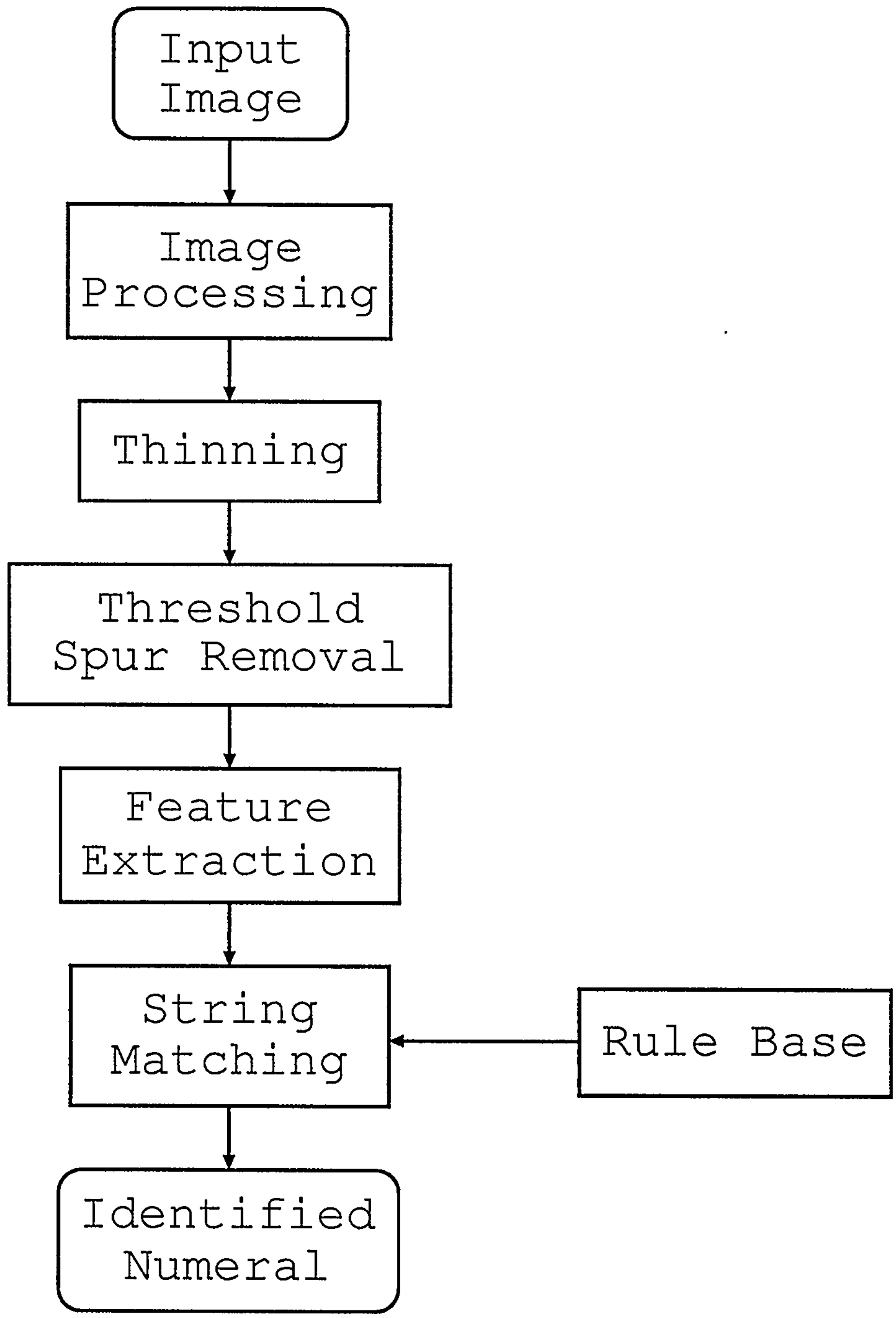


Figure 3.1. System structure for a structural character recognition system.

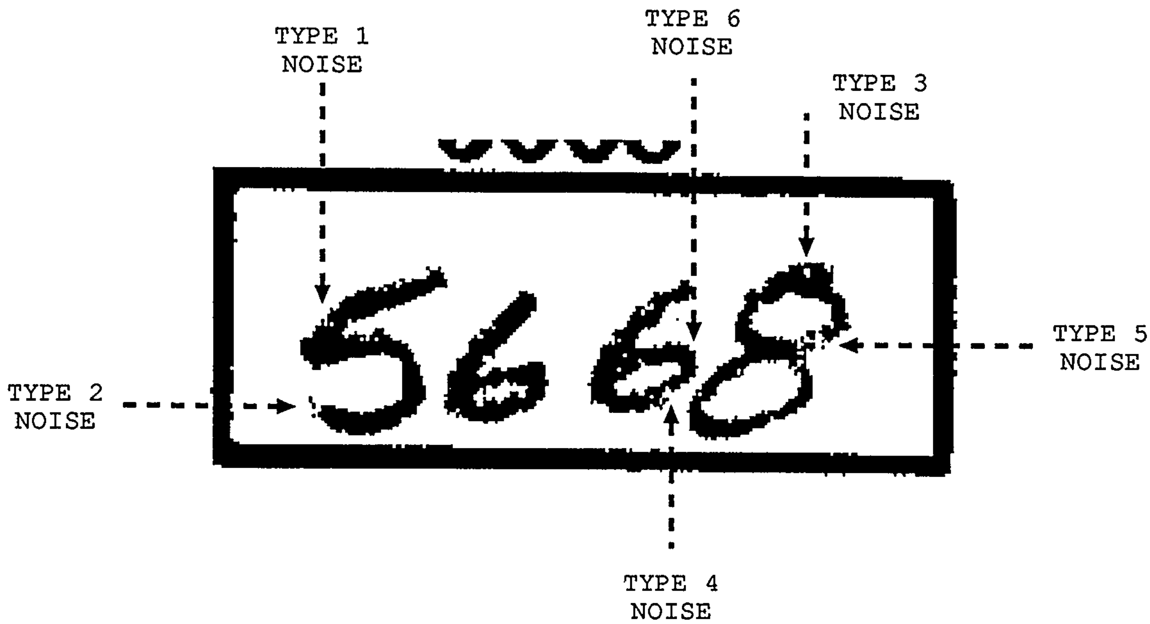


Figure 3.2. A noise-contaminated image.

Figure 3.3. A thinned image without smoothing.

Type 3: *Artificial hole.* The size of the hole can be single or multiple pixel dropouts. This kind of noise will mislead a thinning procedure.

Type 4: *Rough edge.* Such an edge results in irregularity in a character contour.

Type 5: *Broken stroke.* The gap between broken parts of a stroke can be of various sizes.

Type 6: *False corner.* This noise will cause false end points and branches in thinning.

The above noises can be further grouped into two basic classes: black noise and white noise. The black noise consist of the extra black dots or branches on the image. Black noise will bring in extra points, strokes and rough edges, as shown in Fig. 3.3. In the case of black noise, some extra black image points need to be erased. Type 1, type 2 and type 6 noises belong to the black noise. White noise is the extra white dots or missing black pixels in the image. White noise will introduce broken strokes, artificial holes and rough edges, as shown in Fig. 3.3. In the case of white noise, some blank spaces need to be filled in. Type 3 and type 5 belong to the white noise category. Type 4 noise can belong to both classes.

3.1.2 SMOOTHING ALGORITHM

The smoothing method used by Brown *et al* [11] is a single pass, moving average (low-pass filter). This filter is nonrecursive. If more than half of the pixels in a $(2k+1)$ by $(2k+1)$ window centered on the pixel in question are black, the centered pixel is marked black; otherwise, it is marked as a white point. The problem in using this smoothing method is essentially related to the determination of the proper window size. If the window is too small, there may be too much smoothing. The window size (the value of k) is usually determined by users.

3.2 THINNING

To extract the structural feature of a handwritten symbol, a thinning process is necessary to thin the numeral image toward a skeleton form. The thinning process removes points or layers of outline from a pattern until all the lines or curves are of unit width, i.e., a single pixel wide. The resulting set of lines and curves is called the *skeleton* of the object.

3.2.1 REQUIREMENTS FOR THINNING

The essential characteristics of a skeleton can be summarized as follows:

1. Connectivity should be preserved. If the object is connected, the resulting skeleton should also be connected. If the initial background is connected, the background resulting from thinning should also be connected. The points which can affect the connectivity of the pattern to be thinned are usually called *break points*.

2. Excessive erosion should be prevented. The *end points* of a skeleton should be detected so that the length of a line or curve that represents a true feature of the object is not shortened excessively.

3. The skeleton should be immune to small perturbations in the outline of an object. Noise, or small convexities, which do not belong to a skeleton, will very often result in a tail after thinning. The length of these tails should be minimized.

3.2.2 PARALLEL AND SEQUENTIAL THINNING ALGORITHMS

Most thinning algorithms are iterative. In each iteration (or pass), the edge points of a pattern to be thinned are examined against a set of criteria to decide whether the edge points should be removed or not. These algorithms were classified as *par-*

allel algorithms or *sequential algorithms* by Rosenfield et al [70] [79]. In a parallel algorithm, only the result obtained from the previous iteration affects the decision to remove a point in the current iteration, making it suitable for processing by parallel hardware such as array processors. A sequential algorithm uses the result obtained from the previous pass and the results obtained so far in the current pass to process the current pixel. Thus at any point in an iteration, a number of pixels has already been processed. These results can be used immediately to process the following pixels.

3.2.2.1 PARALLEL THINNING ALGORITHMS

Parallel thinning algorithms differ in the way they handle the break points and the end points. The break points and end points are called *safe points*. Some algorithms [61] test safe points by examining a set of windows for a given edge point situation, while others [90] test safe points by checking the number of white/black transitions when the eight neighbors are traversed and by counting the number of black neighbors.

In some parallel algorithms [53], a 2-pixel wide line will be completely removed, since at the beginning of the pass points on both sides of the line will not break the connectivity of the pattern if they are examined independently. If both sides are examined in parallel using the results from the previous pass, they will be removed simultaneously because the result of removing one side of a line is unknown to the other side during the same pass.

The time complexity of a parallel algorithm implemented on a sequential computer consists of three components:

1. In every pass and in every subiteration, each pixel in the bitmap has to be examined once to identify the dark pixels. The number of operations is proportional to the area of the bitmap.

2. Every black pixel has to be examined for edge points. The number of operations

is proportional to the area of the objects in every pass.

3. The number of passes is related to the "thickness" of the object.

3.2.2.2 SEQUENTIAL THINNING ALGORITHMS

The sequential technique is an alternative to parallel methods. Less memory is required in sequential algorithms. Besides, it is generally believed that a sequential algorithm is faster than a parallel algorithm implemented on a sequential computer [70]. The time complexity of sequential algorithms still depends on the size of the bitmap. However, a significant reduction in time complexity can be achieved by examining only those points that belong to the outline of an object. Xu and Wang [88] introduced the idea of contour generation, which was demonstrated to be superior to many thinning algorithms.

Among sequential algorithms, the contour tracing technique was introduced to deal with nearly thinned objects or thick objects. In this case the contour describing the edge of an object is traced in each iteration.

3.3 FEATURE EXTRACTION

Past research on a human's pattern recognition skill shows that human pattern recognition consists of two major procedures: perception and cognition. Both procedures play important roles in human image pattern recognition and directly affect the result of the recognition.

In the perception step, people extract features of the image they are looking at such as the special points, edges, length, direction, curvature of curved segments, etc. This procedure corresponds to the feature extraction procedure in a pattern recognition program. However, a human being has an ability far superor to that of a computer program in perception. A five-year-old child can recognize more things, in a more precise way, from a more complicated scene than any program, even if it does not know what the object it is looking at is.

In the second procedure, i.e., cognition, people use the information acquired in the first procedure to classify the object they are trying to recognize. Feedback is provided to the first procedure to ensure optimum classification.

Following the above principles, most of the current handprint optical character recognition (OCR) algorithms employ both perception and cognition procedures to recognize handwritten characters. Glenn Baptista and K.M. Kulkarni [4] proposed a method which extracted terminal points, intersection points, bend points (threshold based method) and line/curve features (mean square error threshold method) and then employed a structural (syntax) approach to recognize the handwritten numerals. R.M. Brown, T.H. Fay and C.L. Walker [11] used only closed/unclosed segments and a tree structure classifier using the above segment information and project sum information for characters with the same segment information [11], in their recognition method. The approach proposed by F.H. Cheng *et al* [15] used the middle point of a stroke to define its location, and the orientation of a stroke to define its shape, and then used fuzzy logic to calculate the similarity of strokes.

As one can see from the above discussion, most algorithms pay much attention to the cognition procedure, i.e., the classification algorithms, but the perception procedures are relatively less sophisticated. Only a simple description of segment features such as closed/unclosed segments [11], and line/curved segment [4] are given. The definitions of curves are mostly based on pre-set thresholds on which strong assumptions of the probability distribution of curves are made.

For machine printed or formally handprinted characters, the above descriptions and assumptions work well and a high recognition rate can be expected, since writing styles are relatively similar and predictable. However, for handprinted characters with a large variety of writing styles, no strokes are identical even if they are written by the same hand. No definite assumptions can be made for handwritten numerals.

Simple descriptions of strokes do not work as well as they do for machine printed characters. The recognition rates are not as high as claimed, and recognition failures occur for characters with similar shape or stroke, as shown in Fig. 3.4.

1 7 2 3 4 9 0 6

Figure 3.4. Examples of similar handprints.

A human being has a far greater ability for recognition than computer programs. Studying and imitating how human beings recognize is an important way to improve the performance of computer programs. When correctly extracted features are given, as in the cases of printed character recognition, a computer program can work almost as well as any human being does. However, in the cases of real handwriting recognition, the rate of computer recognition drops rapidly while the rate of a human being remains high. It shows that a human being has much greater ability in perception than a computer program. A human being can still extract features correctly while a computer fails, but the ability of cognition are almost the same. One can easily notice that a human being makes a more precise description on strokes than most algorithms currently do and a human being also has interaction between perception and cognition to give the possibility of adjusting and correcting, while most algorithms do not.

The following chapters of this thesis propose a new approach for recognition which gives a more precise stroke description, interacts between perception and cognition, and recognizes through learning and memorizing, like a human being does. This approach provides adaptivity over a large range of writing styles.

3.4 MATCHING

Matching is the final step in structural recognition. The extracted features of the unknown character are compared with the features of pre-stored standard characters. The standard character which has the closest match to the input character is considered as the result of the recognition.

String matching techniques are employed by most researchers. Baptista [4] developed a table of 114 strings for 10 numerals. The feature string of an input character is compared with each of the strings in the table to get the closest matching.

Lam and Suen[51] used a relaxation matching techniques for string matching. In their method, the feature of each segment or substring of a template was matched against the features of each segment of an input pattern. The likelihood of the match was based on the proximity of the features, and the initial probability of this match was defined by this local information. The initial probability was then revised according to the similarity of the context in which each segment appears. Contextual information was used to determine the closeness of the match.

A decision tree is also employed to implement the recognition process. Brown, Fay and Walker [11] developed a binary tree, namely, the Pattern Analysis Language tree, to analyze the structure of a character. The numbers of points and segment features were used to build up the tree.

CHAPTER 4

A NEW HIGH ACCURACY RECOGNITION ALGORITHM

4.1 INTRODUCTION

In this research, an accurate and adaptive structural recognition method is developed. It uses structural information to represent the primary, intrinsic and natural properties of a pattern. This algorithm has the following features:

1. **Accurate Feature Extraction**

   In many cases of handwritten character recognition, the differences between features or characters are tiny but critical. Qualitative feature description and extraction, like polygon approximation, is usually not detailed enough to detect such differences, and results in an increase in the rejection or failure rate of the recognition. To overcome this drawback of the structural recognition method, a new approach for describing and extracting features qualitatively as well as quantitatively is developed in this study.

2. **Neural Network Classifier**

   In this study, neural network classifiers are employed to replace the classical language theory classifier used in structural pattern recognition. Compared with the language theory classifier, a neural net has the following properties:

   - Quantitative.

     A back-propagation neural network is a determinate classifier in an N dimensional continuous space. This space can be divided into an arbitrary number of sub-spaces with any size and shape. Therefore, an N dimensional structure vector of real numbers can be precisely clustered into the sub-space to what it belongs. The language theory classifier deals with a qualitative discrete string representation which is not precise enough for the continuous variations in handwritten characters.

- Forthright.

  the neural network approach is a non-algorithm method. Neither a reasoning, matching, and searching programs nor any kind of data-base requires construction. After proper training, the results of recognition can be obtained from the output nodes when a structure vector is entered into the input nodes.

- Adaptive.

  An neural networks are adaptive to different styles and distortions of characters with training. No modification of the data-base or algorithm is required.

3. **Iterative Spur Removal**

Spur noises have been a persistent problem for those structural character recognition methods which depend on a thin-line representation generated from a raster image [11]. Spur noises bring in extra branches, dots, and curves into the structure of a character. They may appear to be meaningful segments.

One approach for spur removal assumes that all spurs are shorter than a certain threshold. Another approach [11] assumes that spurs occur only in certain areas. Therefore the methods remove all segments shorter than a threshold level and/or those which are located in a certain area before recognition.

However, these assumptions have some drawbacks. In handwritten character recognition, the spurs may be longer than the pre-set thresholds, the meaningful segments may be shorter than the threshold, and spurs could appear anywhere outside the assumed area. In such cases, either spurs remain untouched or some meaningful segments are removed.

The spur removal approach developed in this work is an iterative approach. It is an imitation of the human recognition process, with interaction between the recognition and the feature extraction. Firstly, the extracted feature of a character is sent for recognition. When the result of the recognition is not satisfied, which means spurs exist, the most likely spur is removed by a spur removal process. Then the character is sent back for re-extraction and re-recognition. The process ends when a satisfactory recognition is obtained.

By this method, no assumption about threshold and/or the area of the spur is required and problems due to such assumptions can be avoided.

4.2 ACCURATE FEATURE EXTRACTION

After the image preprocessing process, the thin-line or skeleton figure representation of the original raster image of the input character is obtained. This thin-line representation is in chain-code format. Then, this image is organized to give it the necessary "stroke structure" for further processing.

The subsequent processing performed on this thin-line image increases its information density content by organizing the image points into a feature list. On the basis of these generated list structures, a complete "stroke structure" of a character can be constructed.

Two kinds of features, namely, point features and segment features are employed in this approach.

4.2.1 POINT FEATURE

Feature points are divided into three categories:

TYPE 1: *Terminal Point.* A pixel on a segment which is adjacent to only one other pixel is called a terminal point (Point 1 and point 4 in Fig. 4.1).

TYPE 2: *Circle Point.* A pixel with three distinct neighbors and which is passed twice by the outer contour is called a circle point (Point 3 in Fig. 4.1).

TYPE 3: *Fork Point.* A pixel which is the neighbor of three other pixels and is passed three times by the outer contour is called a fork point (Point 2 in Fig.4.1).

A feature point together with its position in a character is called *point feature.*

Corner is a rather vague concept in character recognition because it depends on the size and style of each character. Thus, no clear definition has been given [46]. A corner is often detected by a pre-set threshold, which is avoided in this approach. The corner point used in [4] is not defined as a feature point here. Thus, the massive computation cost and the mistakes introduced by a corner detection process can be avoided. Instead, the property of a corner is well represented by the segment features as discussed in the next section.

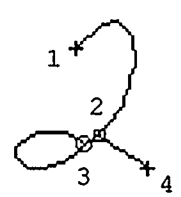


Figure 4.1. An illustration of different feature points.

The accurate structural feature description proposed above is unique and continuous to the description of continuous variations in handprinted characters through both qualitative and quantitative analyses. Combined with a back-propagation neural network for feature extraction, the whole feature extraction process can describe and classify the feature of a character precisely. By means of this description, features which can not be detected or classified by other methods, such as polygon approximation, and corner detection, can be defined and classified. Furthermore, no threshold for the classification is required.

4.2.2 SEGMENT FEATURE

A set of pixels bounded at both ends by a terminal point, a fork point or a circle point is called a *segment* or a *stroke*. The shape of a stroke between two feature points is called a "*segment feature*". A segment feature is a very important feature in character recognition. Generally speaking, the rate of handprint recognition depends on the rate of stroke recognition because a stroke may have hundreds of variations. On the other hand, the difference in shape between different strokes may be trivial but vital, as shown in Fig. 3.4 [18]. In such a situation, obviously any assumption about a particular parametrical probability density function is highly dubious. It is almost impossible to find the actual distribution of any measurement without guessing something. Based on the above observations, we have the following points:

- The representation of a segment feature should be not only qualitative but also quantitative[15], i.e., the description of a stroke has to be sufficiently precise to differentiate the tiny differences in shape from other strokes.
- The stroke classifier should be robust to tolerate noise and variation of writing styles and sensitive to differentiate different strokes with similar shapes.
- A nonparametric method, assuming no particular function, should be used.

In this research, an approach to precisely describe a segment has been developed. A classification network has been built and trained to achieve the above goals.

4.2.2.1 CHAIN-CODE REPRESENTATION

To process a curve segment, one needs to represent the image of the curve with a data structure which can be easily stored and accessed. Chain code representation can be used to represent a curve.

A chain code representation of a curve segment is a sequence of directions in which the curve is traversed pixel by pixel. A chain code representation contains complete shape and scale information of a curve. Thus the curve can be fully recovered from its chain code representation. This chain code is represented by a linear data structure and can be easily stored and accessed.

The direction to the west is defined as "0," the direction to the southwest as "1," the direction to the south as "2," and so on, as shown in Fig. 4.2.

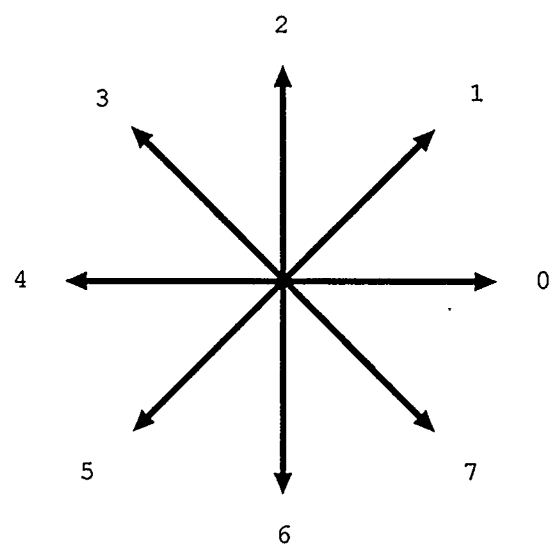


Figure 4.2. An illustration of chain code.

To generate a chain code representation of a curve, the pixel on one end of the curve is taken as a *start point.* From the start point to the next point on the curve, a direction can be obtained. This direction is recorded as the first element of the chain code representation. From the next point to the following point on the contour, another direction can be obtained. This direction is recorded as the second element of the chain code representation. This process is continued until the whole curve has been traversed.

An example of generating the chain code representation of a curve "2" is shown in Fig. 4.3.

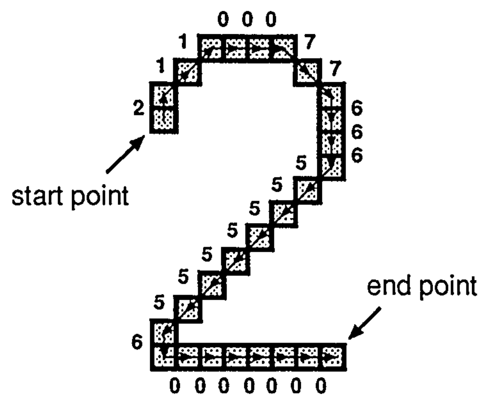


Figure 4.3. An illustration of chain code generation.

The generated chain code representation for the above numeral "2" is:

2 1 1 0 0 0 7 7 6 6 6 5 5 5 5 5 5 5 6 0 0 0 0 0 0 0.

4.2.2.2 CURVATURE SEQUENCE REPRESENTATION

The chain code representation of a curve segment is rotation dependent. If the curve tilts a bit, its chain codes change completely. This characteristics of the chain code is not ideal for the representation of shape information of a curve. To eliminate the rotation dependency, the curvature sequence representation is often used to describe the shape of a stroke.

A curvature is the difference in directions of two adjacent chain segments. The curvature sequence is obtained as follows: starting from the first chain segment s_1 at one end of the stroke, if the direction of the next segment turns left k $\times$ 45° from the direction of the previous segment, the curvature at the turn point is k; if it turns right k$\times$ 45°, the curvature is -k, that is:

$$c_i = s_{i+1} - s_i;$$
$$\text{if } (c_i < -3)\ c_i = c_i + 8;$$
$$\text{else if } (c_i > 4)\ c_i = c_i - 8;$$

where,

c_i is the ith element of the curvature sequence,

s_i is the ith element of the chain code sequence (i = 1,2,3,..., n-1), and

n is the number of chain segments on the stroke.

An example of generating an element of a curvature sequence is shown in Fig. 4.4.

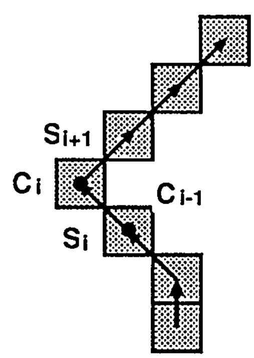
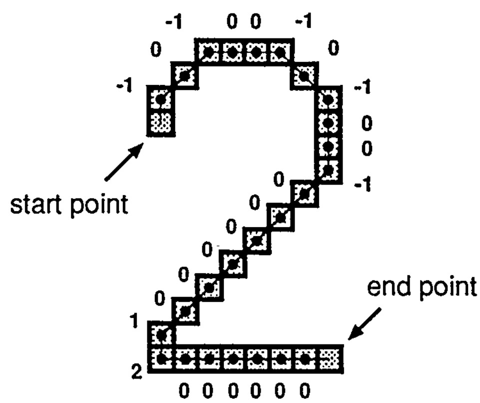


Figure 4.4. An illustration of generating a curvature sequence representation.

An example of a curvature sequence is shown in Fig. 4.5.

4.2.2.3 COMPACT CURVATURE SEQUENCE REPRESENTATION

A curvature sequence can describe the shape of a curve precisely, independent of rotation and translation of the curve. However, a curvature sequence is scale



(a) Curvature sequence generation for a numeral "2"

-1 0 -1 0 0 -1 0 -1 0 0 -1 0 0 0 0 0 0 1 2 0 0 0 0 0 0

(b) Curvature sequence representation of the numeral "2" in (a)

Figure 4.5. An illustration of a curvature sequence representation.

dependent. When the scale of a curve changes, its curvature sequence representation changes correspondingly. Besides, the curvature sequence of a curve is usually very long. Its length equals the length of the curve minus 2, ranging from 1 to above 100.

A new representation of segment feature — *compact curvature sequence* is proposed in this research. The new representation is based on the following observations:

1. only the turning points on a curve carry the basic shape information of the curve. If the position of turning points and the extent of the turning are known, a curve can be precisely described.

2. the non-turning points provide the position information of the turning points.

To generate a compact curvature sequence, the following operations are put on the curvature sequence of a curve:

- All elements with absolute value |k| >1 are extended to k elements with absolute value equal to 1 for normalization. After this operation, the curvature sequence in Fig. 4.5 will change from

  -1 0 -1 0 0 -1 0 -1 0 0 -1 0 0 0 0 0 0 1 2 0 0 0 0 0 0

  to

  -1 0 -1 0 0 -1 0 -1 0 0 -1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0.

- The curvature sequence is compacted by removing all elements whose value is zero. All non-zero elements are moved forward to fill the position left by zero elements. The above sequence will change to

  -1 -1 -1 -1 -1 1 1 1.

- Every element with a non-zero value is multiplied by a weight equal to the distance from the element to the nearest end of the stroke divided by the length of the stroke. Since the length of the curve "2" in Fig. 4.5 is 27, the above curvature sequence will change to

$$\frac{-1}{27}\ \frac{-3}{27}\ \frac{-6}{27}\ \frac{-8}{27}\ \frac{-11}{27}\ \frac{8}{27}\ \frac{7}{27}\ \frac{7}{27}$$

- Shift up the values of all elements by +0.5. The range of the values of all elements is changed to [0.0, 1.0] to meet the requirement of the classification network.

The resultant curvature sequence representation of the shape of a stroke is then ready for classification and it has the following features:

- *precise:* This representation is not only qualitative but also quantitative. Every stroke has its unique and precise representation.

- *complete:* This representation contains the complete shape information of a stroke. The shape of stroke can be completely recovered through this representation.

- *compact:* Since only turning points on a curve are recorded in this sequence, the sequence is relatively short. No matter how long it is, the most complex stroke takes only 10 digits for its representation which is suitable for stroke representation and classification.

- *unified:* length of the stroke. All the values of the representation are between 0.0 and 1.0, which is suitable for processing in the classification stage.

- *independent:* Only shape information is kept in this sequence. The representation is independent of translation, rotation and scale of a curve segment.

4.2.3 DATA STRUCTURE FOR FEATURES

All the features extracted so far are stored in vectors or matrices for easy access to the following classification process. The feature points are numbered according to their order of appearance in the chain of a thinned character. Their coordinates are stored in two vectors, POINT_X and POINT_Y. Then these numbers are stored in a vector POINT_CHAIN according to their order in the chain:

$$\mathbf{POINT\_CHAIN} = (feature\_point, feature\_point, feature\_point, \cdots) \tag{4.1}$$

The type of each feature point in vector POINT_CHAIN is also stored into a vector called POINT_TYPE:

$$\mathbf{POINT\_TYPE} = (point\_type, point\_type, point\_type, \cdots). \tag{4.2}$$

The curve features are stored in matrices. The length of each curve segment is in a matrix called CURVE_LENGTH. The row number i and column number j of the matrix are the numbers of feature points, and the content of the matrix is the length of the curve segment from feature point i to feature point j.

The CURVE_LENGTH matrix of the character in Fig 4.1 is

$$\mathbf{CURVE\_LENGTH} = \begin{bmatrix} L_{00} & L_{01} & \ldots & L_{0n} \\ L_{10} & L_{11} & \ldots & L_{1n} \\ \ldots & \ldots & \ldots & \ldots \\ L_{n0} & L_{n1} & \ldots & L_{nn} \end{bmatrix}. \tag{4.3}$$

And the corresponding CURVE_TYPE matrix is

$$\mathbf{CURVE\_TYPE} = \begin{bmatrix} T_{00} & T_{01} & \ldots & T_{0n} \\ T_{10} & T_{11} & \ldots & T_{1n} \\ \ldots & \ldots & \ldots & \ldots \\ T_{n0} & T_{n1} & \ldots & T_{nn} \end{bmatrix}. \tag{4.4}$$

where, L_{ij} is the length of curve from feature point i to feature point j, T_{ij} is the type of curve from feature point i to feature point j ($i, j = 1, 2, 3, \ldots, n$), and n is the number of feature points.

All the elements of the curve chain are replaced by the corresponding elements of the curvature sequence. The first element of the curvature sequence is the direction of the curve segment.

4.2.4 STROKE CLASSIFICATION

Based on the analysis of handprinted numerals, it can be seen that all strokes can be clustered into 7 basic classes as shown in Fig. 4.6.

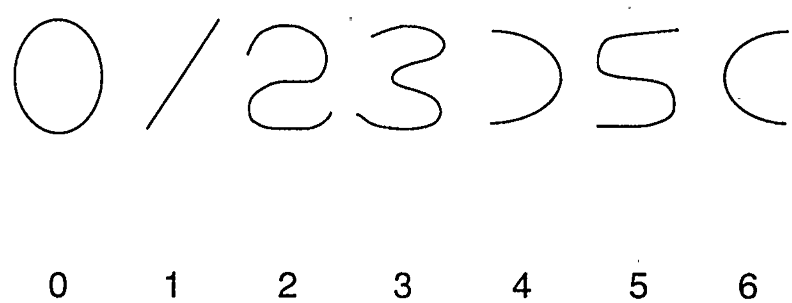


Figure 4.6. Basic strokes of handprinted numerals and the corresponding codes.

Each numeral is composed of the basic strokes. All strokes of numerals are classified through a multi-layer perceptron network trained with the *error back-propagation* algorithm.

4.3 A NEURAL NETWORK CLASSIFIER

The task of a classifier is to classify the structural information extracted above into a proper category. The structural information of a character is firstly represented by a structure vector. Then the structure vector is classified by a set of neural networks.

4.3.1 STRUCTURE VECTOR

Qualitative structural features of a character can be represented by the type of feature points, the type of strokes between feature points, and the relationship among them.

Quantitative structural information is included in scale, stroke length, relative position of feature points, etc.

The segment features and point features of a character are combined into a single structure vector as follows:

$$\begin{aligned}\textbf{structure vector} &= (x\_scale, y\_scale, \\ &\quad x_1,\ y_1,\ stroke\_type_1,\ stroke\_length_1, \\ &\quad x_2,\ y_2,\ stroke\_type_2,\ stroke\_length_2,\ \cdots) \end{aligned} \tag{4.5}$$

where:

x_scale, y_scale — the normalized x and y scales of the numeral by dividing the x and y sizes of the numeral by preset maximum values

x_i, y_i — the relative x and y coordinates of feature point i divided by its x and y sizes respectively for normalization

stroke_type — the normalized stroke type code obtained through the stroke classification network

stroke_length — the normalized length of the stroke by the size of the numeral.

This structure vector contains both qualitative and quantitative structural information. It has the following properties:

- it provides a precise description of a character, and

- it contains complete information of a character so that the character can be fully recovered from its structure vector.

Due to the above advantages of the new structure vector, better recognition results can be achieved.

4.3.2 CLASSIFICATION

For the classification of structure vectors, multilayer feed forward neural networks are applied.

The structure vectors are fed directly to the neural network. Since a structure vector is relatively small compared to its image matrix, the network required for structure vector classification is much smaller than that for image matrix classification. The time of training and classification can be reduced significantly.

To further reduce the time of training and classification, to improve the convergence of the network, and to obtain a higher recognition rate, the structure vectors are divided into several groups according to their primary structure information, i.e., *type code.*

Type code is a three digital decimal number. It contains the basic information of a character such as the number of terminal points, fork points, and circle points. This information can be used to preclassify the structure vector of the character.

The type code is defined as follows:

$$\begin{aligned} type\ code \ &= \ number\ of\ cross\_points * 100 + \\ & \quad\ number\ of\ fork\ points * 10 + \\ & \quad\ number\ of\ terminal\ points. \end{aligned} \tag{4.6}$$

From the type code of a character, the basic structural information can be obtained. For instance, the type code of the numeral "2" in Fig. 4.7 is "112". From the type

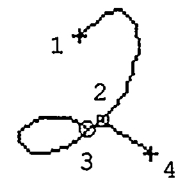


Figure 4.7. The type code of a character.

code "112," it can be deduced that there are 1 fork point, 1 circle point, and 2 terminal points in this character.

For handwritten numerals, the number of their type code is limited. Only 12 type codes are reasonable. These possible type codes are 000, 002, 011, 020, 022, 031, 033, 103, 105, 112, 123, 204. Any other type codes than those above would be considered as noisy characters.

After being preclassified by its type code, the structure vector of a handwritten numeral is input into a multilayer feed forward neural network for classification. The corresponding output of the network indicates the recognition result of the numeral.

Compared with classical language model classification methods, the neural networks operate by means of distributed processing rather than reasoning, and no rules and regulations are required for classification and learning. Therefore, a neural network can be easily trained to adapt to various characters written in different styles and conditions.

A multilayer perceptron is a kind of continuous neural network which can operate on continuous variables in real number space. This continuity makes a multilayer perceptron superior to the classical method in solving the continuous variation problem of handwritten recognition. In general, any difference between different characters can be detected and described by this neural network.

Compared with neural networks applied in matrix processing methods, the neural networks presented in this study have a much smaller size. Since only the structure vector rather than the whole image matrix needs to be processed, the number of neurons ranges from 50 to 70 for structure vector classification as compared to 300 to 1000 for image classification. Hence, training and classification time can be reduced and better convergence and higher recognition rates can be obtained.

Because normalization is performed only on the structure vector instead of the whole image, the time consuming alignment, rotation and translation operations on the image are no longer required. However, better adaptation to the variations of characters is provided by the structure vector classification.

4.4 ITERATIVE SPUR REMOVAL

When the input numeral does not fit any one of the 12 classes, and it is not a single meaningless stroke, it is regarded as a character with noise. The numeral with noise is returned to the feature extraction step to remove noise and to re-extract its features.

Handwritten characters may have noise on them. A thinning process can also produce extra strokes which make the structure of a character different from the one it is supposed to be. Those extra strokes which make the structure of a character abnormal are regarded as spurs and spurs produce difficulties in recognition if they are not dealt with properly.

Few rules on writing styles can be applied because of individual differences. It is almost impossible to set up a threshold to remove noise since the important stroke in one numeral is sometimes much smaller than the noise in other numerals.

Furthermore, human beings do not remove noise in visual patterns by threshold logic. Instead, there is an interaction between perception and cognition. In our approach, we built up an interactive channel between the stroke classification network and the recognition networks through iterative filtering and refinement. The numeral with noise is sent back to the feature extraction step. The shortest stroke in this character is considered likely to be noise. The shortest stroke and its associated feature points are removed, the related strokes are re-extracted, and the feature storages are modified. After this the numeral is sent back to the recognition step once more. The process ends when all noise is removed.

Most of the published algorithms [4] [15] [42] [46] do not consider the effect of spurs. The input characters are assumed ideal, i.e., the structures are invariable, except with changes in shape, scale, slope, etc. No spurs caused by thinning and image noise are included. Loisia Lam's [51] method simply removes all short segments, i.e., all segments whose lengths are under a certain threshold are assumed to be spurs and are removed. R.M. Brown [11] reported that all spurs occur only in a certain area of the character called the "feasible region" which is the combination of the lower 25%, the upper 70% of the height, and the left 30% of the width of a character. The exceptions for this rule are the middle segment of the numeral "3" and the right segment of the numeral "4." The other rule for spur definition is that a spur must be the shortest one of the three branches of a "Y" intersection, as shown in Fig. 4.8.

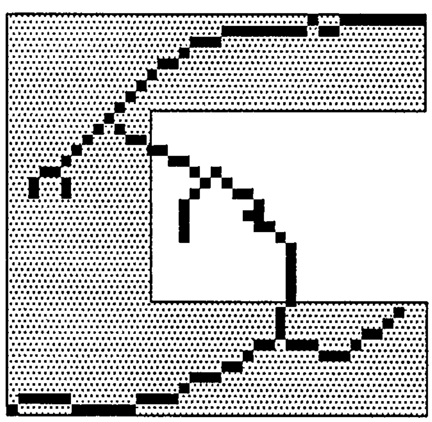
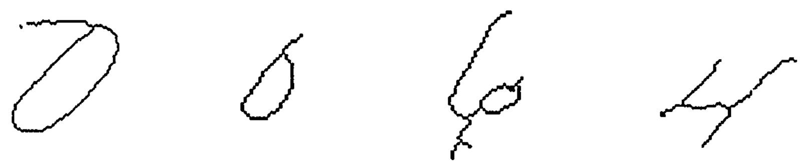


Figure 4.8. Brown's feasible region for spurs (shaded area).

The above assumptions about spurs are not applicable in the following frequently occurring situations in handwritten numerals:

- Spurs may not be short (see Fig. 4.9).



0 0 6 4

Figure 4.9. An illustration of long spurs.

- Strokes may be short(see Fig. 4.10).

2 2 2 4 2 9

Figure 4.10. An illustration of short strokes.

- Spurs could be in any area, not necessarily in only certain areas (see Fig. 4.11).
- Spurs could be on any parts of a character, not necessarily in a "Y" intersection.

The method of spur removal described in the following section is called *most probable spur method.* By this method, the shortest branch of a character with an abnormal structure is erased. The procedure is iterated until all the spurs are removed. This method is based on the assumption that the spur is always the shortest branch of a

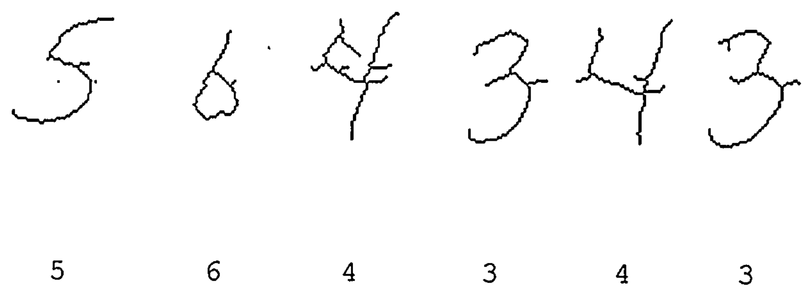


Figure 4.11. Spurs in random positions.

character with abnormal structure. Because spurs may be produced by a thinning process or a noisy image, they might not be shorter than a threshold, but they are relatively shorter than other meaningful strokes. If there are some strokes which are shorter than spurs on the same character, the strokes can be usually ignored without much effect on recognition.

The input character is first sent to the recognition process. If the result shows that the structure of this character is abnormal, the shortest branch is detected and removed by the spur removal process. The character is then sent back to the recognition process, and so on, until the character is recognized.

The major differences between this method and other spur removal methods mentioned above are:

- The concept of this method is that the spurs are shorter than strokes.
- No threshold is used for spur removal. This solves the conflict between short strokes.

4.5 THE SYSTEM STRUCTURE

The system structure is shown in Fig. 4.12. It consists of the following five components, namely, smoothing, thinning, feature extraction, classification and spur removal. After the raster image of a character is entered, smoothing is performed to remove black and white noise and to smooth the rough edges of the image. In the thinning process, the raster image of the character is shrunk to a skeleton, which represents the structure of the character. The skeleton of the character is analyzed in the feature extraction process. A feature vector is formed in this stage as a value representation of the structure. The structure vector is then classified by several neural networks according to its structure code in the recognition process. If no satisfactory result is obtained, the spur removal process is applied to remove the most likely spur. The character is fed back to the feature extraction process for re-extraction and re-recognization, until a satisfactory result is obtained.

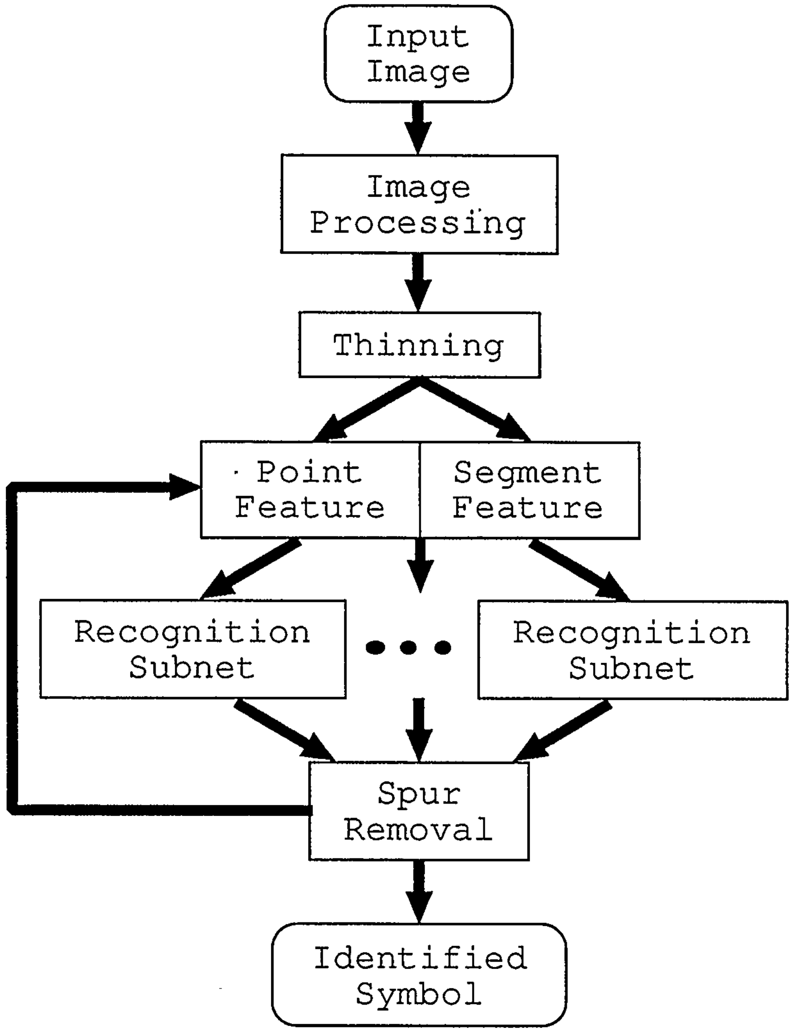


Figure 4.12. The system structure of an accurate handwritten numeral recognition.

CHAPTER 5

IMPLEMENTATION

In this chapter, experimental data for testing and training are introduced and the approaches proposed in Chapter 4 are implemented.

5.1 INPUT DATA

The experimental data used in this research for training and testing is from the U.S. NIST (National Institute of Standards and Technology) handprinted character database. The database consists of 2100 pages of bilevel (i.e. black and white) image data of hand printed numerals and text with over 1,000,000 characters. The data was collected from 2100 individuals distributed across the United States with a sampling roughly proportional to population density [86].

5.1.1 SAMPLE CHARACTERS

Each of the 2100 samples consists of an image of a handprinting sample form. The data collected on the form is located in 34 boxes. In addition to the primary image, the database contains isolated images of 33 of these boxes. A sample form is shown in Fig. 5.1. The forms have been scanned at a resolution of 300 pixels/inch. The total image database consists of 3 Gigabytes of image data with 273,000 numerals.

The numerical samples can be divided into two types of samples: three boxes of the ten individual digits and twenty-five boxes of numbers, five boxes each of two, three,

HANDWRITING SAMPLE FORM

NAME	DATE	CITY	STATE	ZIP
[redacted]	08/02/89	FLINT	Mi	48504

This sample of handwriting is being collected for use in testing computer recognition of hand printed numbers and letters. Please print the following characters in the boxes that appear below.

0123456789	0123456789	0123456789
0123456789	0123456789	0123456789

86	506	8941	95304	891405
86	506	8941	95304	891405

521	5407	60170	689547	98
521	5407	60170	689547	98

6081	77132	314200	78	464
6081	77132	314200	78	464

93847	256369	63	224	6902
93847	256369	63	224	6902

551339	78	722	5798	21313
551339	78	722	5798	21313

b g v x u j d y o h s m t f c w q i a k r e z p l n

bgvxujdyohsmtfcwqiakrezpln

F S H K D X T E Z R Q M L A B G V I Y P U C O J W N

FSHKDXTEZRQMLABGVIYPUCOJWN

Please print the following text in the box below:
We, the People of the United States, in order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common Defense, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our posterity, do ordain and establish this CONSTITUTION for the United States of America.

We, the People of the United States, in order to form a more perfect Union, establish Justice, insure domestic tranquility, provide for the common Defense, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our posterity, do ordain and establish this CONSTITUTION for the United States of America.

Figure 5.1. An illustration of a sample form.

four, five and six digit numbers. The numerical samples are uniformly distributed with 26,000 samples for each digit.

5.1.2 DATA COLLECTION

The original form was generated on a laser printer with fifty variations. Each variant has different number and alphabet sequences selected at random. The laser printed forms were reproduced by photocopying for mail distribution to the writers.

The handprinting sample was obtained from a selection of field data collection staff of the Bureau of the Census with a geographic sampling corresponding to the population density of the United States. No effort was made to sample based on education, or occupation, etc.

5.1.3 CHARACTERISTICS OF THE DATA

No restriction on the writing implements was used in the sample. The range of implements used ranged from wide, felt-tipped pens to hard, sharp-pointed pencils. This results in images with a wide variety of line types.

The size of characters varies from 3mm high to 7mm high, with average size of 4.5mm high. The size of large characters is constrained by the box height of 7.5mm.

The range of characters and spatial placement of those characters is broad enough to present a very difficult challenge to image recognition systems [86].

5.2 IMAGE PREPROCESSING

The image preprocessing consists of two major processes: smoothing and thinning.

5.2.1 SMOOTHING

The smoothing method used in this study is a morphographical low pass filter. This low-pass filter is implemented by the expansion and erosion functions in morphography. The expansion and erosion functions are referred to as filling and erasing operations in this work for clarity. The filling operation fills or removes the white noise, while the erasing operation removes the black noise.

The filling operation in one direction is implemented by moving the whole image by one pixel in the demanded direction and executing an "OR" function between the original image and the moved image; then moving the resultant image back by one pixel and executing an "AND" function between the resultant image and the moved result image, i.e.,

$$A = A + \mathbf{MOVE}(A) \tag{5.1}$$

$$A = A \times \mathbf{MOVE}^{-1}(A) \tag{5.2}$$

where,

A — Image being processed.

MOVE() — Move image in one given direction by one pixel.

$\mathbf{MOVE}^{-1}$() — Move image in the opposite direction of **MOVE** by one pixel.

$+$ — Pixel by pixel "OR."

$\times$ — Pixel by pixel "AND."

To remove noise in all directions, the movement should be in all directions. The actual filling algorithm takes the following forms:

1. Removing white noise in the horizontal direction:

$$A = A + \mathbf{LEFT}(A), \tag{5.3}$$

$$A = A \times \mathbf{RIGHT}(A). \tag{5.4}$$

2. Removing white noise in the vertical direction:

$$A = A + \mathbf{UP}(A), \tag{5.5}$$

$$A = A \times \mathbf{DOWN}(A). \tag{5.6}$$

3. Removing white noise in the 45^o direction:

$$A = A + \mathbf{UPRIGHT}(A), \tag{5.7}$$

$$A = A \times \mathbf{DOWNLEFT}(A). \tag{5.8}$$

4. Removing white noise in the 135^o direction:

$$A = A + \mathbf{UPLEFT}(A), \tag{5.9}$$

$$A = A \times \mathbf{DOWNRIGHT}(A). \tag{5.10}$$

The filling algorithm expands the image and then shrinks it in all directions. After this operation, white noise with unit width can be removed.

To remove black noise, an erasing algorithm is employed. Erasing is the opposite operation of filling. It shrinks the image first and then expands it in all directions. It can remove all black noise with sizes similar to that above.

1. Removing black noise in the horizontal direction:

$$A = A \times \mathbf{LEFT}(A), \tag{5.11}$$

$$A = A + \mathbf{RIGHT}(A). \tag{5.12}$$

2. Removing black noise in the vertical direction:

$$A = A \times \mathbf{UP}(A), \tag{5.13}$$

$$A = A + \mathbf{DOWN}(A). \tag{5.14}$$

3. Removing black noise in the 45° direction:

$$A = A \times \mathbf{UPRIGHT}(A), \tag{5.15}$$

$$A = A + \mathbf{DOWNLEFT}(A). \tag{5.16}$$

4. Removing black noise in the 135° direction:

$$A = A \times \mathbf{UPLEFT}(A), \tag{5.17}$$

$$A = A + \mathbf{DOWNRIGHT}(A). \tag{5.18}$$

The filling and erasing operations are VLSI oriented algorithms. They can be implemented by current VLSI technology in a single chip. The other feature of these algorithms is that they both operate in a fully parallel style so that a high computing speed can be expected.

After smoothing, the box frame around the figures is detected by row/column histogram analysis and then removed. The original image and the smoothed image are shown in Fig. 5.2. and Fig. 5.3.

5.2.2 THINNING BY CONTOUR GENERATION

In both parallel and contour tracing techniques, pixels are removed from the contours without knowing what is going to remain in the object. The result is that either all the pixels will have been removed or, to prevent this from happening, a non-unit-width skeleton will remain after final iteration.

The thinning algorithm employed in this research, *thinning by contour generation*, was developed by Kwok [50]. The feature of this algorithm is that it considers the previous results obtained for processing the current pixel. If a pixel is to be removed, the new contour, which will be exposed to the background can be computed. Thus, when the current contour is traversed, a section of the new contour is generated for

Figure 5.2. An illustration of an original image.

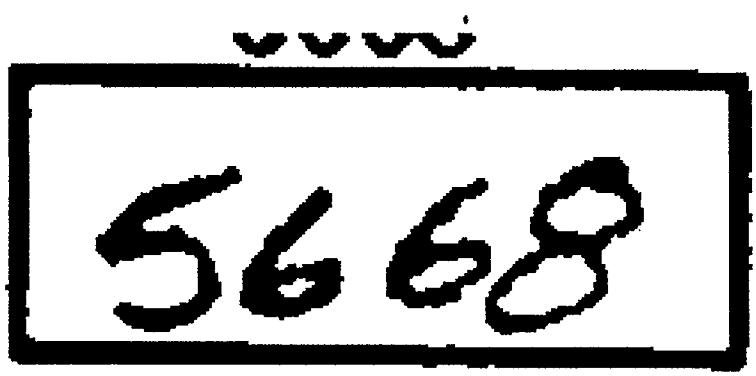


Figure 5.3. An illustration of a smoothed image.

every pixel in the current contour visited. The section is checked for break points and this information is available when subsequences are visited. At the end of the iteration, a new contour will be available for the next iteration without having to remove the old one.

At any time, the algorithm will have complete knowledge of what remains of the object when the current contour is removed. Thinning is completed when there are no nonsafe points in any of the new contours.

5.2.2.1 CONTOUR GENERATION

The contour generation is implemented by shrinking the object in all four directions by one pixel. The completed object is one pixel smaller than the original object. By "ANDing" the original image and the negative of the shrunken image, we can have a one pixel wide contour generated. In the following discussion, we have:

A — an image being processed.

RIGHT() — move the image to the right by one pixel.

LEFT() — move the image to the left by one pixel.

$+$ — pixel by pixel "OR" operation.

$\times$ — pixel by pixel "AND" operation.

$\bar{A}$ — a negative image of image A.

B — am image buffer.

The detailed operations are as follows:

$$B = A, \tag{5.19}$$

$$A = A \times \mathbf{RIGHT}(A), \tag{5.20}$$

$$A = A \times \mathbf{LEFT}(A), \tag{5.21}$$

$$A = A \times \mathbf{UP}(A), \tag{5.22}$$

$$A = A \times \mathbf{DOWN}(A), \tag{5.23}$$

$$A = B \times \bar{A}. \tag{5.24}$$

Following Fig. 5.3, the result of contour generation is shown in Fig. 5.4.

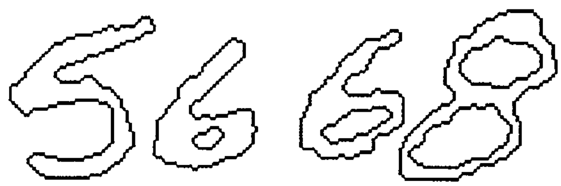


Figure 5.4. An illustration of contour generation.

5.2.2.2 CHAIN-CODE GENERATION

To generate a chain code representation of a contour, the pixel on the top and left corner of the contour is taken as a *start point.* The last point on the contour traversed is called a *end point.* The end point is adjacent to the start point. The direction on the end point to the start point is the last direction element of the chain code representation.

The direction of the traversing is decided by the Left Hand Rule, i.e., when traversing through the contour, the object is always on the left hand of the traverse, as shown in Fig. 5.5.

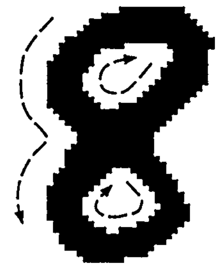


Figure 5.5. An illustration of directions of contour traversing.

5.2.2.3 THINNING BY CONTOUR GENERATION

Using the chain-codes, the outline is plotted on a bitmap **B** with all the pixels having a value of zero. Every pixel visited will have its value increased by one. A pixel visited more than once will have a value greater than one and is therefore a break point.

After plotting the first contour on **B**, the algorithm goes though a number of iterations. The iteration terminates for a particular contour when there are no more nonsafe points in that contour. When the operation is completed, the skeleton is formed in **B** and a chain-code describing the skeleton is also available.

At any point in an iteration, a section of the new contour is generated to correspond to the pixel p_i under consideration. Two direction vectors, dir_{i-1} and dir_i are maintained. The xy coordinates of p_i are updated from the xy coordinates of p_{i-1} using dir_{i-1}. A look up table (Table 5.1) is used that gives the offsets needed for each of the 8 directions.

A flag p_{i-1}_safe is kept to show whether the previous pixel p_i is a safe point or not.

The current pixel p_i is checked for safe point by examining its value in the bitmap **B**. With this information, a unique section of the new contour can be generated. The new contour includes all the safe points uncovered so far as well as the dark points which are neighbors of the current outline. After the final iteration, all of the points on the new contour are safe points and the skeleton is thus obtained. The result of the thinned image is illustrated in Fig. 5.6.

5.3 AN ACCURATE FEATURE EXTRACTION

5.3.1 STROKE CLASSIFICATION

Seven basic strokes are extracted from a training set. These strokes are transformed from their chain code expressions to compacted curvature sequence representation. The basic strokes and their compacted curvature sequence representations are illustrated in Fig. 5.7 $\sim$ Fig. 5.13.

From Fig. 5.7 to Fig. 5.13, it can be seen that the compacted curvature sequences reflect the complete structural information of the strokes. The shape of a stroke is known from the distribution of its compacted curvature sequence. The direction of the bending is reflected by values below or above 0.5. Other detailed information is reflected by the absolute values of the sequence.

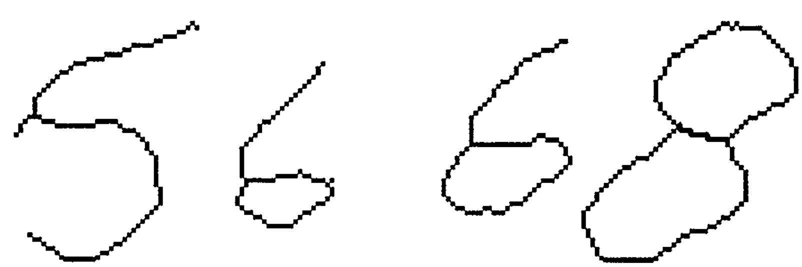
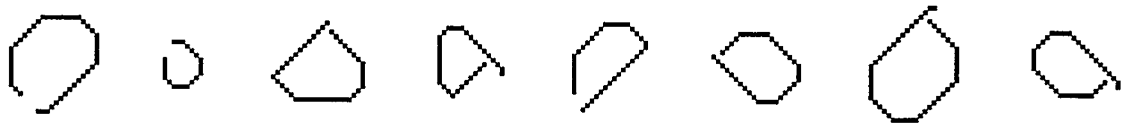


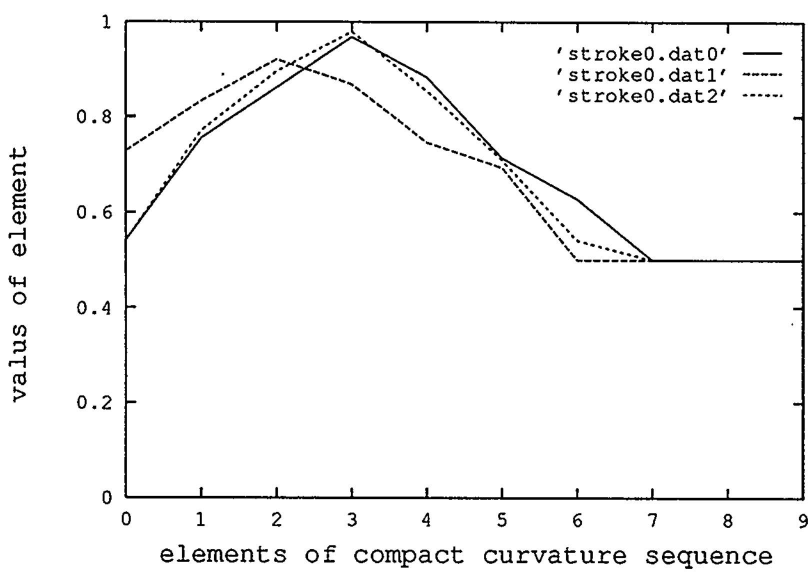
Figure 5.6. A thinned image after smoothing.

	p_{i-1} is a safe point.				p_{i-1} is not a safe point.			
dir_i - dir_{i-1} &7	dir_i &1 ==0	dir_i &1 != 0	dir_i &1 ==0	dir_i &1 != 0	dir_i &1 ==0	dir_i &1 != 0	dir_i &1 ==0	dir_i &1 != 0
0	dir_i +1	dir_i +1 dir_i	dir_i +2	dir_i +3 dir_i +1	dir_i	dir_i	dir_i +2	dir_i +1
1	dir_i	dir_i	dir_i +2	dir_i +1	none	dir_{i-1}		dir_i +1
2	none	dir_i +7		dir_i +1	none	none	delete p_i	
3, 4	dir_{i-1}		p_i itself		dir_i +4		p_i itself	
6	dir_i +3 dir_{i-1} dir_i		dir_i +5 dir_i +3 dir_i +1		dir_{i-1} dir_i		dir_i +3 dir_i +1	
7	dir_i +2 dir_{i-1}	dir_i +2 dir_i	dir_i +4 dir_i +2	dir_i +3 dir_i +1	dir_{i-1}	dir_{i-1} dir_i	dir_i +2	dir_i +3 dir_i +1
	chain code elements		contour point direction w.r.t. p_i		chain code elements		contour point direction w.r.t. p_i	

Table 5.1. A summary of chain code elements and the generated contour points.

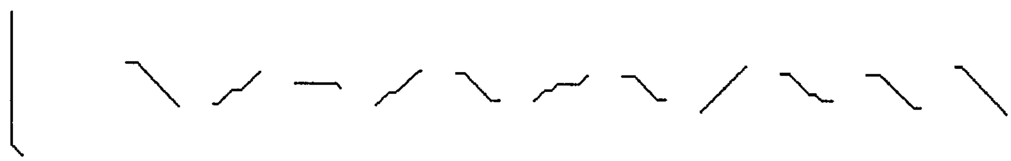


(a) Samples of stroke 0

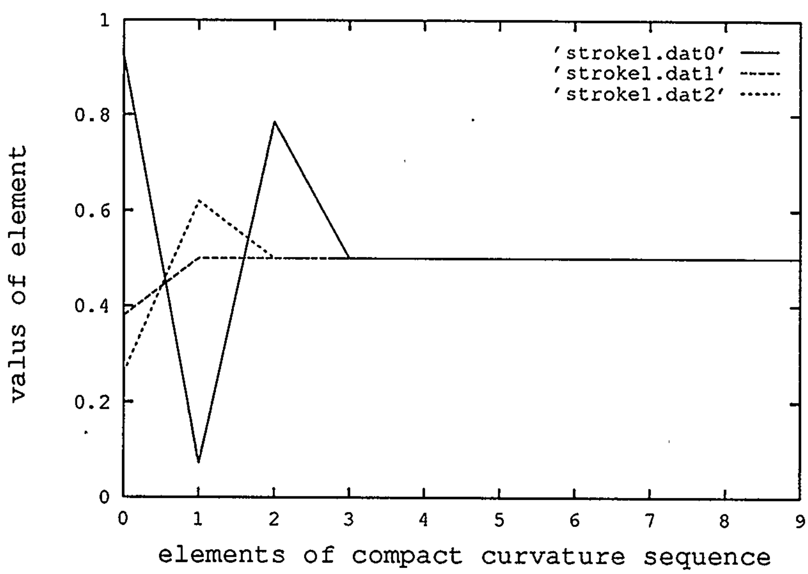


(b) Graphical illustration of compact curvature sequences of stroke 0 in (a).

Figure 5.7. The basic stroke 0 and its compacted curvature sequence representation.

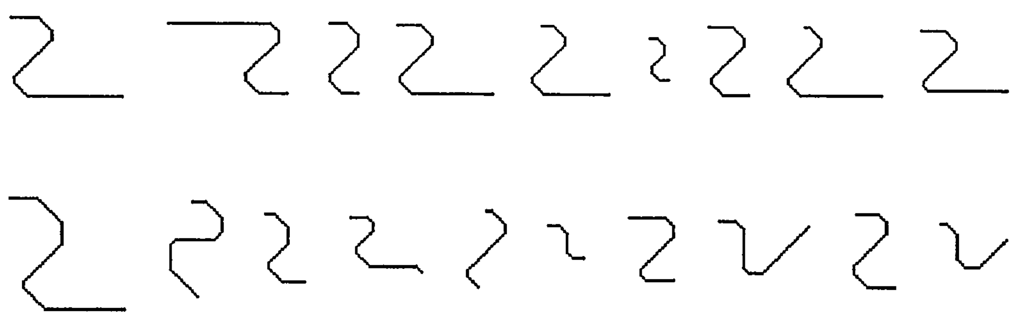


(a) Samples of stroke 1

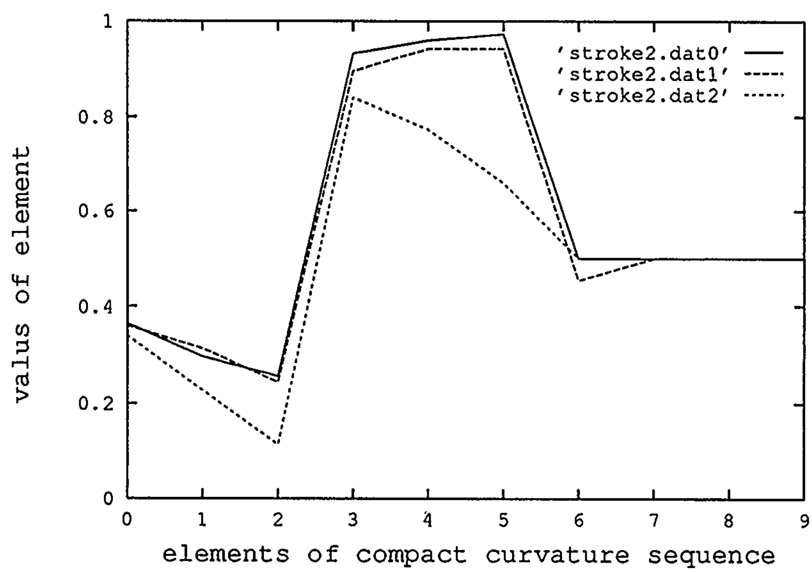


(b) Graphical illustration of compact curvature sequences of stroke 1 in (a).

Figure 5.8. The basic stroke 1 and its compacted curvature sequence representation.

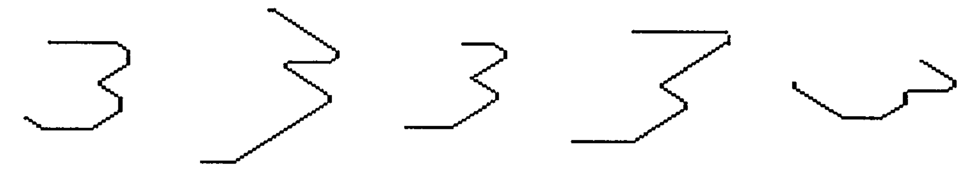


(a) Samples of stroke 2

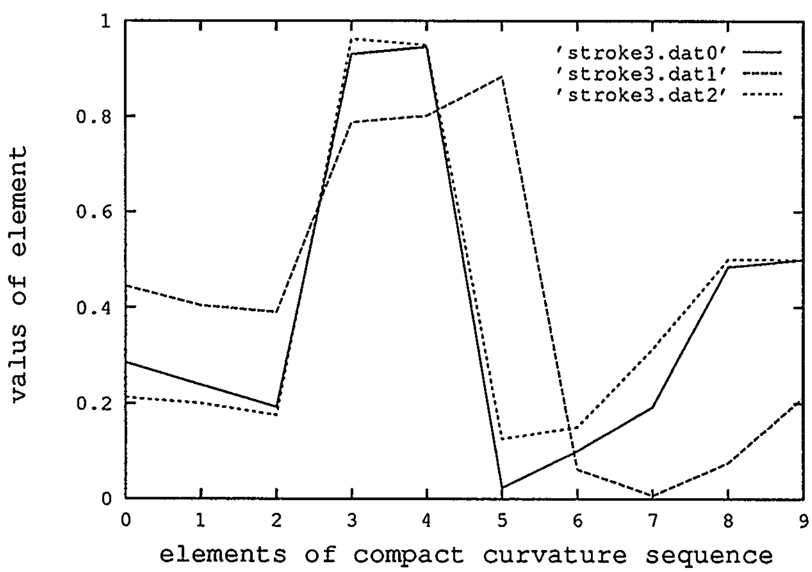


(b) Graphical illustration of compact curvature sequences of stroke 2 in (a).

Figure 5.9. The basic stroke 2 and its compacted curvature sequence representation.

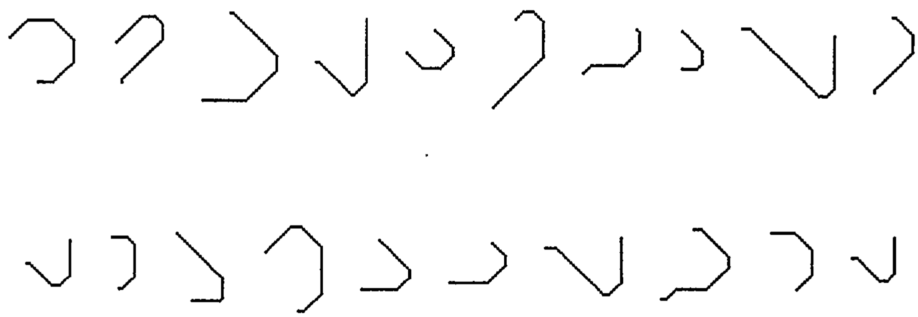


(a) Samples of stroke 3

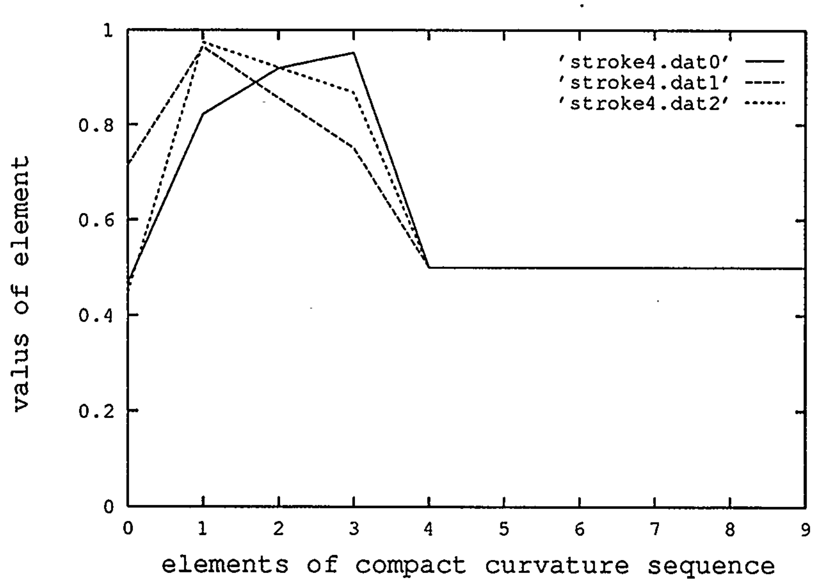


(b) Graphical illustration of compact curvature sequences of stroke 3 in (a).

Figure 5.10. The basic stroke 3 and its compacted curvature sequence representation.

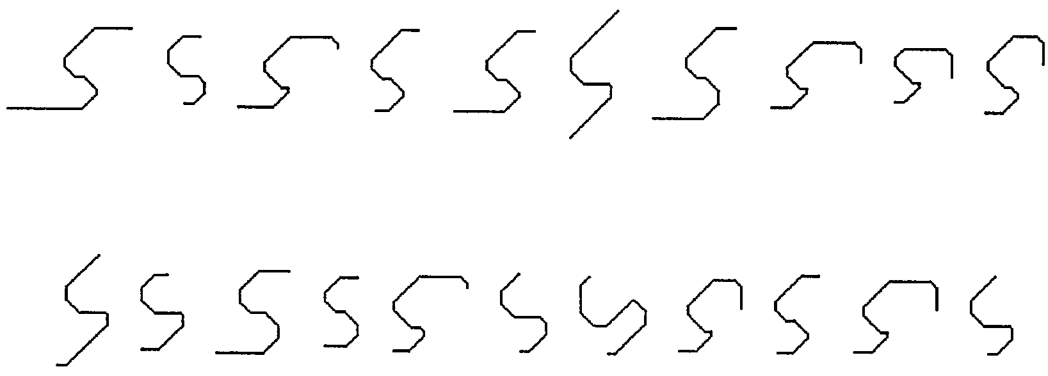


(a) Samples of stroke 4

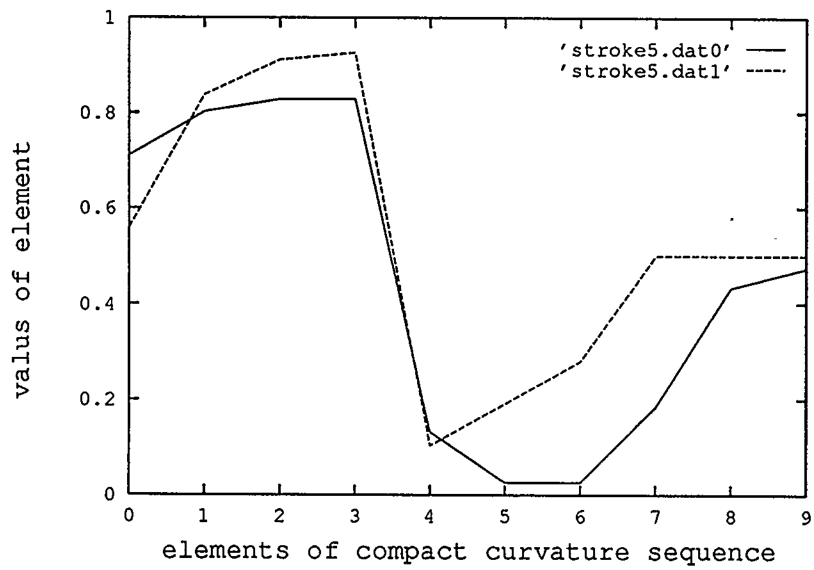


(b) Graphical illustration of compact curvature sequences of stroke 4 in (a).

Figure 5.11. The basic stroke 4 and its compacted curvature sequence representation.

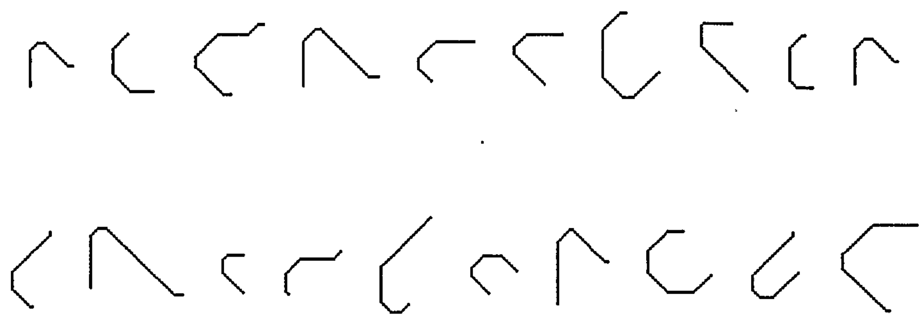


(a) Samples of stroke 5

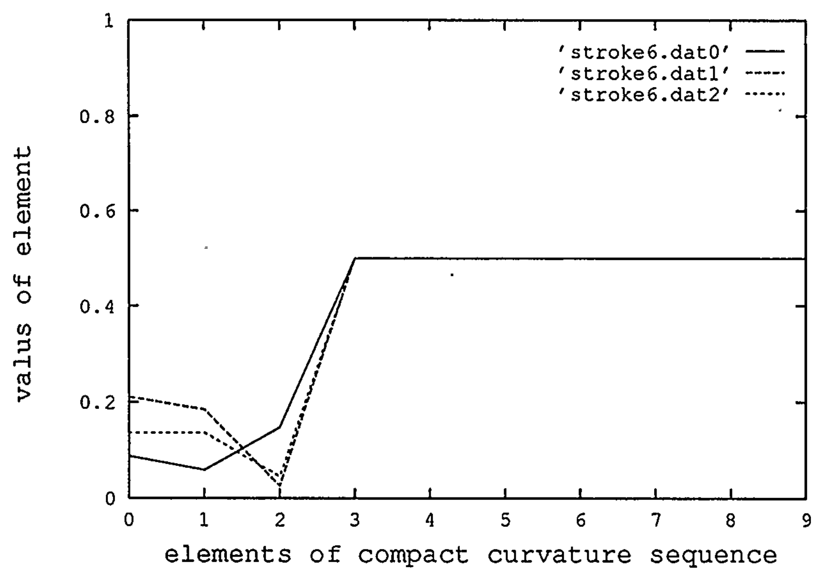


(b) Graphical illustration of compact curvature sequences of stroke 5 in (a).

Figure 5.12. The basic stroke 5 and its compacted curvature sequence representation.



(a) Samples of stroke 6



(b) Graphical illustration of compact curvature sequences of stroke 6 in (a).

Figure 5.13. The basic stroke 6 and its compacted curvature sequence representation.

These compacted curvature sequences are input into a stroke classification neural network for classification. The type of a stroke is represented by the number of the highest output plus the value of that highest output. For instance, if a compacted curvature sequence is put into the stroke classification network and the output node 4 has the highest value of 0.876, the type of the stroke to be classified is represented by 4.876.

5.3.2 CONFIGURATION OF THE STROKE CLASSIFICATION NETWORK

The neural network for stroke classification has 10 input nodes, 10 hidden nodes, and 7 output nodes, as shown in Fig. 5.14. The weights of the net are initialized with random values between -0.5 and +0.5 using a uniform distribution before training.

There are 10×10 free weights from the input layer to the hidden layer plus 10 thresholds, and 10×7 weights from the hidden layer to the output layer plus 7 thresholds. Thus, the total number of free weights to be trained is 187. The number of hidden units is obtained by testing.

5.3.3 TRAINING OF THE STROKE CLASSIFICATION NETWORK

The training set contains 200 strokes and it takes 210 iterations to train the neural network. The maximum system error is 0.005 while the rate of recognition on the training set is 100%. On a test set of another 100 strokes, the recognition rate is also 100%.

The training process and system error of the stroke classification neural network are shown in Table 5.2 and Fig. 5.15.

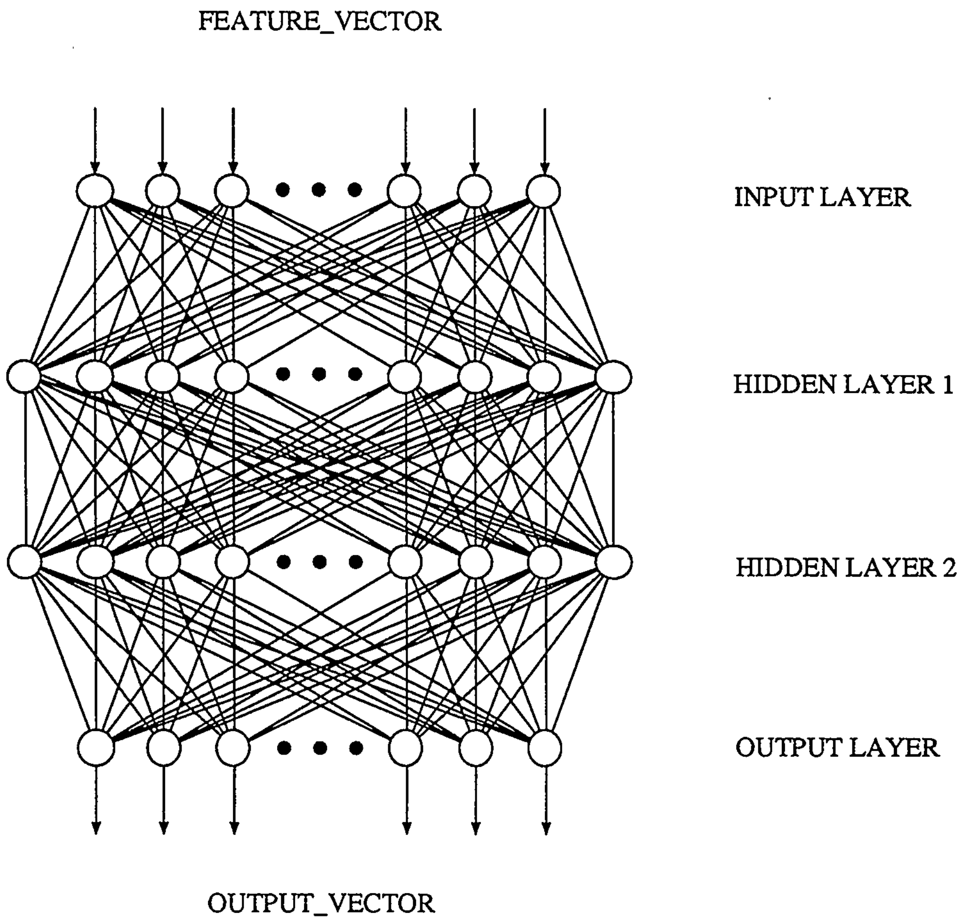


Figure 5.14. The structure of a recognition network.

Training Iterations	System Errors
20	0.244273
40	0.153109
60	0.100899
80	0.089203
100	0.067819
120	0.075133
140	0.035044
160	0.040885
180	0.044194
200	0.007916
220	0.002133
240	0.001485
260	0.001179
280	0.000994

Table 5.2. Training - system error of stroke classification.

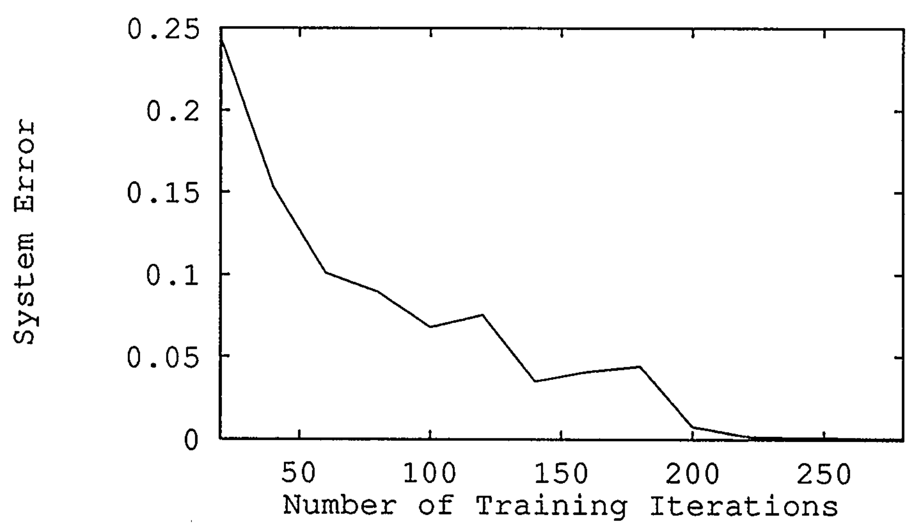


Figure 5.15. Training errors of stroke classification neural network.

5.4 AN ADAPTIVE STRUCTURE CLASSIFICATION

The whole recognition net is split up into 12 subnets, one for each type code (see section 4.3.2). Each subnet has 27 input nodes, 15 first hidden layer nodes, 15 second hidden layer nodes, and 15 output nodes. The training set for the subnets is about 1,000 handwritten numerals. The average training process for those subnets takes a few minutes. The average recognition process takes 855 floating point multiplications.

The structure of each subnet is shown in Fig. 5.14. The number of input nodes of each subnet is equal to the length of the structure vector of that subnet. The number of both hidden layers is adjustable according to the distribution of the structure vectors and the divisions required in the feature space. This number is finally decided through experiment.

The output layer contain 13 nodes. Nodes 0 $\sim$ 9 are for numeral "0" $\sim$ "9." Nodes 11 and 12 are for special characters "-" and "/," respectively. Node 10 is for noise.

Due to different writing conditions and styles, some handwritten characters may have abnormal structures as shown in Fig. 5.16. Even a thinning process produces some extra strokes which make the structure of the character much different from the one it supposed to represent (see Fig. 5.16). Those extra strokes which make the structure of a character abnormal are considered to be noise. The noise produces misleading results in the recognition process if it is not handled properly.

Since the training set and testing set are large, the training and recognition processes of the recognition network take hours and the convergence rates are poor. It is efficient to split the whole recognition network into a number of smaller subnetworks. Numerals are pre-classified by their type codes and are sent to these subnetworks. Since the sizes of subnetworks can be much smaller than the whole network, the subnetworks not only speed up the training and recognition processes significantly, but also converge in really all cases.

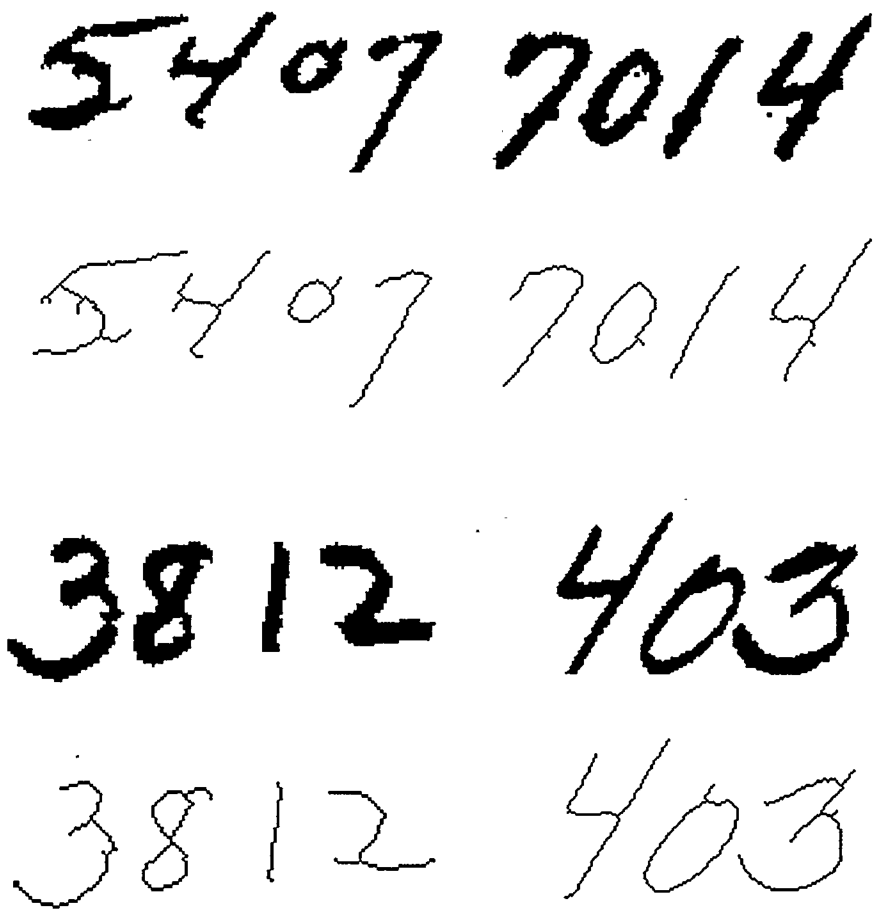


Figure 5.16. Samples of spurs.

5.4.1 DISCUSSION

The structure of a handwritten numeral is represented by a real number structure vector. According to the type code, the structure vector is classified by one of several 3 layer back-propagation neural networks. Less training time and better convergence are achieved because of the small size of each neural network.

By using the structure vector and classification neural networks, a recognition rate of 98% is obtained.

5.5 AN EXAMPLE OF RECOGNITION

An example is given in this section to illustrate the recognition process.

Original image:

The original input image is shown in Fig. 5.17.

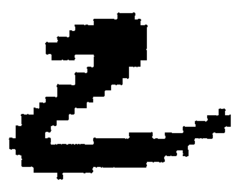


Figure 5.17. An original input image.

Step 1: Image processing.

Random noise on the original image is removed by filling and erasing operations. The resultant image is shown in Fig. 5.18.

Step 2: Thinning.

The processed image is shrinked to a skeleton using the Thinning by Contour Generation thinning algorithm. The resultant image is shown in Fig. 5.19.

Step 3: Feature extraction.

Segments between feature points are represented by compact curvature sequences

Figure 5.18. The image after processing.

Figure 5.19. The thinned image.

and classified by a neural network. Point features are also extracted as shown in Fig. 5.20. Since there is 1 fork point, 0 circle point and 3 terminal points, the type code of this character is "103". Feature points 1, 3, 4 are terminal points, marked by crosses. Point 2 is a fork point, marked by a square.

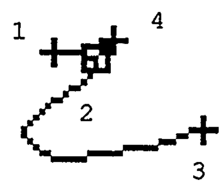


Figure 5.20. The illustration of feature extraction.

The segment and point features are stored in point-type vector, curve-type and curve-length arrays.

$$\textbf{POINT\_TYPE} = (1,\ 3,\ 1,\ 1,). \tag{5.25}$$

$$\textbf{CURVE\_TYPE} = \begin{bmatrix} 0.0 & 1.8 & 0.0 & 0.0 \\ 1.8 & 0.0 & 6.7 & 1.9 \\ 0.0 & 4.7 & 0.0 & 0.0 \\ 0.0 & 1.9 & 0.0 & 0.0 \end{bmatrix}. \tag{5.26}$$

$$\textbf{CURVE\_LENGTH} = \begin{bmatrix} 0 & 8 & 0 & 0 \\ 8 & 0 & 47 & 4 \\ 0 & 47 & 0 & 0 \\ 0 & 4 & 0 & 0 \end{bmatrix}. \tag{5.27}$$

Step 4: Recognition.

At this stage, a structure vector of the numeral to be recognized is formed and classified by the neural network corresponding to type code "103".

The output of the recognition neural network is:

$$\textbf{output} = (0.29, 0.30, 0.20, 0.11, 0.25, 0.17, 0.13, 0.23, 0.30, 0.23, 0.92, 0.13, 0.23) \tag{5.28}$$

Step 5: Iterative spur removal.

The 10th output node of the recognition neural network has the highest output value 0.92, which means there exists a spur on the numeral. The shortest branch and the relevant fork point are removed from the numeral. The numeral is then sent back to the feature extraction stage. Fig. 5.21 shows the result of spur removal.

Figure 5.21. The numeral after spur removal.

Step 6: Feature extraction.

Segment and point features are extracted again from the spur removed numeral. Fig. 5.22 shows the re-extracted feature points.

Figure 5.22. The feature extraction for the spur removed image.

The new type code of the numeral is "002". The segment and point features are stored in point-type vector, curve-type and curve-length arrays.

$$\textbf{POINT\_TYPE} = (1,\ 1). \tag{5.29}$$

$$\text{CURVE\_TYPE} = \begin{bmatrix} 0.0 & 2.9 \\ 2.8 & 0.0 \end{bmatrix}. \tag{5.30}$$

$$\text{CURVE\_LENGTH} = \begin{bmatrix} 0 & 55 \\ 55 & 0 \end{bmatrix}. \tag{5.31}$$

Step 7: Recognition.

A new structure vector of the spur removed numeral is formed and classified by the recognition network for type code "002".

The output of the recognition neural network is:

$$\textbf{output} = (0.29, 0.31, 0.93, 0.15, 0.25, 0.17, 0.13, 0.23, 0.33, 0.27, 0.11, 0.13, 0.23) \tag{5.32}$$

The 2nd output of the recognition network has the highest value 0.93, which indicates that the digit "2" is the most likely result of the recognition.

5.6 SUMMARY

In this chapter, an accurate feature extraction method, an adaptive structure classification method, and an iterative spur removal method are implemented. The structure of a handwritten numeral is precisely described by the structure vector. The structure vector is then classified by one of several back-propagation neural networks. The continuous variations on strokes and structure of a handwritten character can be well described and detected. Training time and system convergence are improved by type code pre-classification. Spurs are removed by interaction between recognition and feature extraction.

The overall performance of 98% recognition rate and 0% rejection rate are achieved for this handwritten numeral recognition system. The results of the recognition process are given in Fig. 5.23.

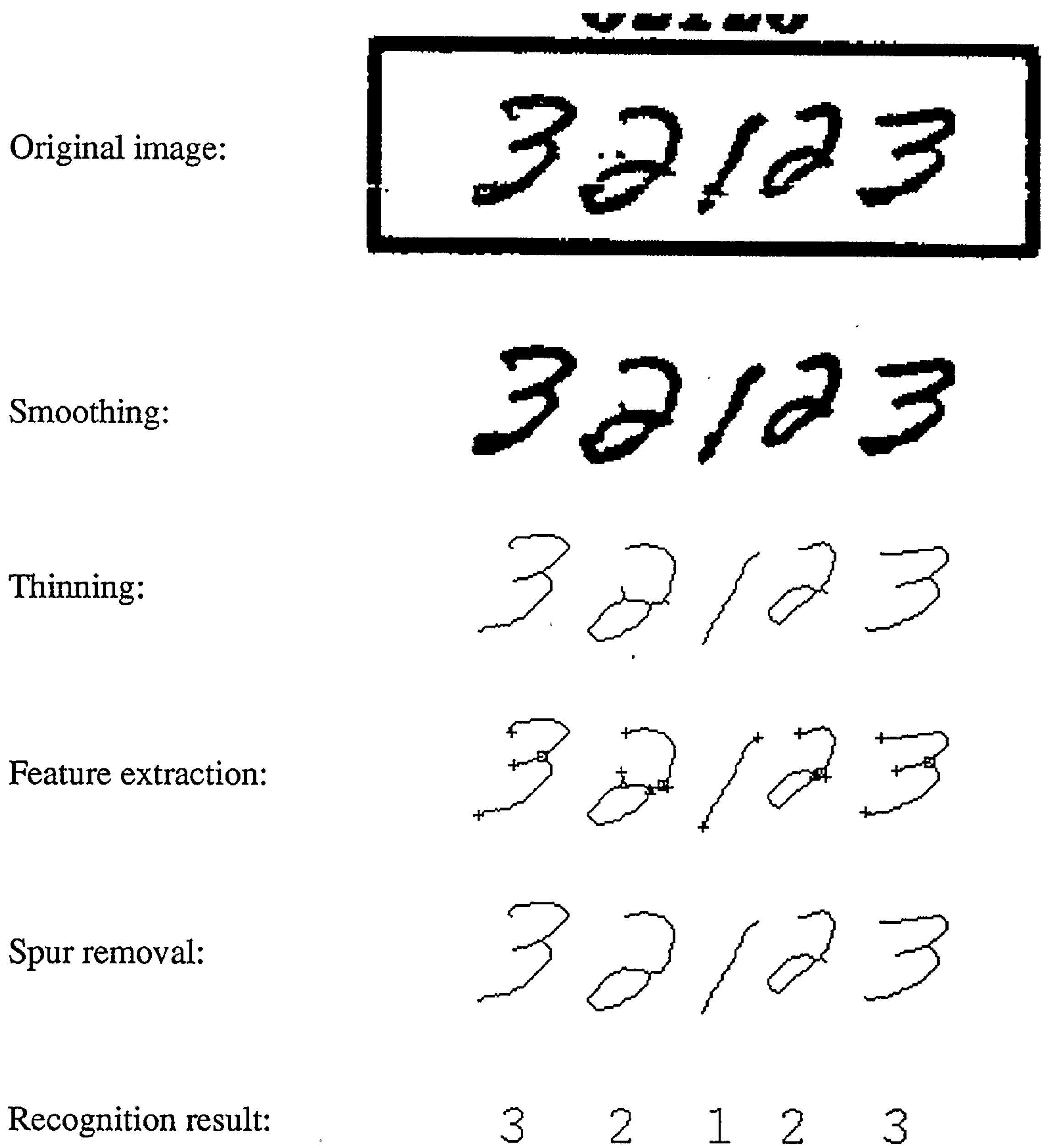


Figure 5.23. An illustration of recognition results.

CHAPTER 6

CONCLUSIONS

OCR is an old but relevant topic. Many algorithms have been developed. However, not much attention has been devoted to feature extraction, an important aspect of the human visual process. Features are typically not quantitative but qualitative, which is the reason for failures [42].

In our approach to handwritten numeral recognition, features in the handwritten numerals are described not only qualitatively but also quantitatively. The feature description can detect tiny differences between similar but different characters. When combined with a multilayer perceptron neural network, errors in feature extraction can be eliminated and a high success rate of final recognition can be achieved.

In addition, a neural classifier with interactive noise-removal is presented. It simulates the human visual recognition process and works effectively. The interactive noise-removing technique results in a zero rejection rate as compared to about 5 to 10% in the other systems.

REFERENCES

[1] Y.S. Abu-Mostafa and D. Pslatis. Optical neural computers. *Scientific American*, 256:88–95, 1987.

[2] R. Anderson. The complexity of parallel algorithms. Master's thesis, Stanford University, Stanford, CA, 1985.

[3] N. Ansari and E.J. Delp. Partial shape recognition: a landmark-based approach. *IEEE Transactions on PAMI*, 12:470–483, 1990.

[4] G. Baptista and K.M. Kulkarn. High accuracy algorithm for recognition of handwritten numerals. *Pattern Recognition*, 21:287–291, 1988.

[5] H.B. Barlow, R. Narasimhan, and A. Rosenfeld. Visual pattern analysis in machines and animals. In *Seminar on Visual Mechanisms and Form Perception held at*, T.I.F.R., Coloba, Bombay, 1971.

[6] S.T. Barnard. Choosing a basis for perceptual space. Technical Report 315, SRI International, 1984.

[7] H.G. Barrow and R.J. Poppelstone. *Relational Descriptions in Picture Processing*. American Elsevier Publishing Inc., Vol. 6:377-396. University of Edinburgh, 1971.

[8] J. Beck, B. Hope, and A. Rosenfeld. *Human and Machine Vision*. Academic Press, Boston, 1983.

[9] B. Bell and L.F. Pau. Contour tracking and corner detection in a logic programming environment. *IEEE Transactions on PAMI*, 12:913–917, 1990.

[10] B. Bhanu. Evaluation of automatic target recognition algorithms. In A. Oosterlinck and P.E. Danielsson, editors, *Architecture and Algorithms for Digital Image Processing*, Proceedings of SPIE 435:18-27, 1983.

[11] R.M. Brown, T.H. Fay, and C.L. Walker. Handprinted symbol recognition system. *Pattern Recognition*, 21:91–118, 1988.

[12] D.J. Burr. Experiments on neural net recognition of spoken and written text. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36:1162–1168, 1988.

[13] C.A. Cabrelli. Automatic representation of binary images. *IEEE Transactions on PAMI*, 12:1190–1196, 1990.

[14] G.A. Carpenter and S. Grossberg. Neural dynamics of category learning and recognition: attention, memory consolidation, and amnesia. In *Brain Structure, Learning, and Memory*. AAAS Symposium Series, 1986.

[15] F.H. Cheng, W.H. Hsu, and C.A. Chen. Fuzzy approach to solve the recognition problem of handwritten chinese characters. *Pattern Recognition*, 22:133–141, 1989.

[16] M.A. Cohen and S. Grossberg. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on System, Man and Cybernetics*, 13:815–826, 1983.

[17] C. Cox, B. Blesser, and M. Eden. The application of type font analysis to automatic character recognition. *Proceedings of 2nd International Conference on Pattern Recognition*, 226-232, Coprnhagen, 1974.

[18] Y.L. Cun, L.D. Jackel, and B. Boser. Handwritten digit recognition: applications of neural network chips and automatic learning. *IEEE Communications Magzine*, 41-46, 1989.

[19] B. Delgutte. Speech coding in the auditory nerve: processing schemes for vowel-like sounds. *Acoustical Society of America*, 75:879–886, 1984.

[20] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, 1973.

[21] M. Eden. On the formalization of handwriting. *Structure of Language and its Mathematical Aspects*, 83-88, 1961.

[22] M.C. Fairhurst and H.M.S.A. Wahab. An interactive two-level architecture for a memory network classifier. *Pattern Recognition Letters*, 11:537–540, 1990.

[23] X. Fan and J. Gu. An adaptive system for accurate handwritten numeral recognition. prepared for publication, 1993.

[24] X. Fan and J. Gu. An adaptive system for accurate handwritten numeral recognition. *Technical report*, University of Calgary, 1992.

[25] J.A. Feldman and D.H. Ballard. Connectionist models and their properties. *Cognitive Science*, 6:205–254, 1982.

[26] H.F. Feng and T. Pavlidis. Decomposition of polygons into simpler components: feature generation for syntactic pattern recognition. *IEEE Transactions on Computer*, 24:633–650, 1975.

[27] E.C.D.B.C. Filho, M.C. Fairhurst, and D.L. Bisset. Adaptive pattern recognition using goal seeking neurons. *Pattern Recognition Letters*, 12:131–138, 1991.

[28] K.S. Fu, editor. *VLSI for Pattern Recognition and Image Processing. Vol. 13: Information Sciences.* Springer-Verlag, Berlin, 1984.

[29] J. Gashnig. *Performance Measurements and Analysis of Certain Search Algorithms.* PhD thesis, Carnegie-Mellon University, 1979.

[30] O. Ghitza. Robustness against noise: the role of timing-synchrony measurement. In *Proceedings International Conference on Acoustics Speech and Signal Processing.* Dallas, Texas, 1987.

[31] V.K. Govindan. Character recognition — a review. *Pattern Recognition*, 23:671–683, 1990.

[32] H.P. Graf and L.D. Jackel. *VLSI* implementation of a neural network memory with several hundreds of neurons. In *AIP Conference Proceedings 151, Neural Networks for Computing*, Snowbird Utah, 1986.

[33] P.M. Grant and J.P. Sage. A comparison of neural network and matched filter processing for detecting lines in images. In *AIP Conference Proceedings 151, Neural Networks for Computing*, Snowbird Utah, 1986.

[34] R.L. Grimsdale, F.H. Sumner, C.J. Tunis, and T. Kilburn. A system for the automatic recognition of patterns. *Proceedings of IEE*, 106:210–221, 1959.

[35] S. Grossberg. *The Adaptive Brain I: Cognition, Learning, Reinforcement, and Rhythm, and The Adaptive Brain II: Vision, Speech, Language, and Motor Control,*. Elsevier/North-Holland, Amsterdam, 1986.

[36] I. Guyon, P. Albrecht, Y.L. Cun, J. Denker, and W. Hubbard. Design of neutral network character recognizer for a touch terminal. *Pattern Recognition*, 24:105–119, 1991.

[37] L.D. Harmon. Automatic recognition of print and script. *Proceedings of the IEEE*, 60:1165–1176, 1972.

[38] J.A. Hartigan. *Clustering Algorithms.* John Wiley & Sons, New York, 1975.

[39] J.J. Hopfield. Neural network and physical systems with emergent collective computational abilities. *National Academy of Sciences*, 79:2554–2558, 1982.

[40] J.J. Hopfield. Neurons with graded responsehave colective computational properties like those of two-state neurons. *National Academy of Sciences*, 81:3088–3092, 1984.

[41] J.J. Hopfield and D.W. Tank. Computing with neural circuits: A model. *Science*, 233:625–633, 1986.

[42] J.S. Huang and K. Chuang. Heuristic approach to handwritten numeral recognition. *Pattern Recognition*, 19:15–19, 1986.

[43] D.H. Hubel and T.N. Wiesel. Functional architecture of macaque monkey visual cortex. *Royal Society of London*, 198B:1–57, 1977.

[44] D.H. Hubel and T.N. Wiesel. Brain mechanisms of vision. *Scientific American*, 240:130–144, 1979.

[45] E.R. Kandel and J.H. Schwartz. *Principles of Neural Science.* Elsevier, New York, 1985.

[46] D.D. Kerrick and A.C. Bovik. Microcomputer-based recognition of handprinted characters from a tablet input. *Pattern Recognition*, 21:552–537, 5 1988.

[47] T. Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin, 1984.

[48] T. Kohonen, K. Masisara, and T. Saramaki. Phonotopic maps–insightful representation of phonological features for speech representation. In *Proceedings of IEEE 7th International Conference on Pattern Recognition*, Montreal, Canada, 1984.

[49] H.T. Kung. The structure of parallel algorithms. In Marshall C. Yovits, editor, *Advances in Computers.* 19:65-112. Academic Press, New York, 1980.

[50] P.C.K. Kwok. A thinning algorithm by contour generation. *Communications of the ACM*, 31:1314–1324, 1988.

[51] L. Lam and C.Y. Suen. Structural classification and relaxation matching of totally unconstrained handwritten zip-code numbers. *Pattern Recognition*, 21:19–31, 1988.

[52] F.L. Lewis. *Optimal Estimation.* John Willey & Sons, New York, 1986.

[53] H.E. Lu and P.S.P. Wang. A comment on ' a fast parallel algorithm for thinning digital patterns'. *Communications of the ACM*, 29:239–242, 1986.

[54] R.F. Lyon and E.P. Loeb. Isolated digit recognition experiments with a cochlear model. In *Proceedings International Conference on Acoustics Speech and Signal Processing*, Dallas, Texas, 1987.

[55] J. Makhoul, S. Roucos, and H. Gish. Vector quantization in speech coding. In *IEEE Proceedings*, 73:1551-1588, 1985.

[56] J. Mantas. An overview of character recognition methodologies. *Pattern Recognition*, 19:425–430, 1986.

[57] J. Mantas. Methologies in pattern recognition and image analysis– a brief survey. *Pattern Recognition*, 20:1–6, 1987.

[58] P. Meer, C.A. Sher, and A. Rosinfeld. The chain pyramid: hierarchical contour processing. *IEEE Transactions on PAMI*, 12:363–376, 1990.

[59] A.R. Moller. *Auditory Physiology.* Academic Press, New York, 1983.

[60] P. Mueller and J. Lazzaro. A machine for neural computation of acoustical patterns with application to real-time speech recognition. In *AIP Conference Proceedings 151, Neural Networks for Computing,*, Snowbird Utah, 1986.

[61] N.J. Naccache and R. Shinghal. Spta: A proposed algorithm for thinning binary patterns. *IEEE Transactions on System, Man and Cybernetics*, 14:409–418, 1984.

[62] H. Nagahashi and M. Nakasuyama. A pattern description and generation method for structural characters. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 8:112–118, 1986.

[63] A. Namane and M.A. Sid-Ahmed. Character scaling by contour method. *IEEE Transactions on PAMI*, 12:600–606, 1990.

[64] D.B. Parker. A comparison of algorithms for neuron-like cells. In *AIP Conference Proceedings 151, Neural Networks for Computing*, Snowbird Utah, 1986.

[65] T. Parsons. *Voice and Speech Processing.* McGraw-Hill, New York, 1986.

[66] T. Pavlidis and F. Ali. Computer recognition of handprinted numerals by polygonal approximation. *IEEE Transactions on System, Man and Cybernetics*, 5:610–614, 1975.

[67] S.M. Peeling, R.K. Moore, and M.J. Tomlinson. The multi-layer perceptron as a tool for speech pattern processing research. In *Proceedings of IoA Autumn Conference on Speech and Hearing*, 1986.

[68] I. Pitas and A.N. Venetsanopoulos. Morphological shape decomposition. *IEEE Transactions on PAMI*, 12:38–45, 1990.

[69] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, 1985.

[70] A. Rosenfeld and J. L. Pfaltz. Sequential operations in digital picture processing. *Journal of ACM*, 13(4):471–494, 1966.

[71] D.E. Rumelhart and G.E. Hinton. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations.* MIT Press, 1986.

[72] J.P. Sage, K. Thompson, and R.S. Withers. An artificial neural network integrated circuit based on mnos/cd principles. In *AIP Conference Proceedings 151, Neural Networks for Computing*, Snowbird Utah, 1986.

[73] A. Samal. A survey of 3-d representation systems in computer vision. a Talk, 1984.

[74] S. Seneff. A computational model for the peripheral auditory system: Application to speech recognition research. In *Proceedings of International Conference on Acoustics Speech and Signal Processing*, 1986.

[75] M. Shridhar and A. Badreldin. High accuracy character recognition algorithm using fourier and topological descriptors. *Pattern Recognition*, 17:515–523, 1984.

[76] M. Shridhar and A. Badreldin. A high accuracy syntactic recognition algorithm for handwritten numerals. *IEEE Transactions on System, Man and Cybernetics*, 15:152–158, 1985.

[77] M. Shridhar and A. Badreldin. Recognition of isolated and simply connected handwritten numerals. *Pattern Recognition*, 19:1–12, 1986.

[78] D. Sinha and C.R. Giardina. Discrete black and white object recognition via morphological functions. *IEEE Transactions on PAMI*, 12:275–293, 1990.

[79] R. Stefanelli and A. Rosenfeld. A some parallel thinning algorithms for digital pictures. *Journal of ACM*, 18(2):255–264, 1971.

[80] S.L. Tanimoto. A pyramidal to parallel processing. In *International Symposium on Computer Architecture.* 372-378. IEEE Computer Society Press, 1983.

[81] D.W. Tank and J.J. Hopfield. Simple 'neural' optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit. *IEEE Transaction on Circuits Systems*, 33:533–541, 1986.

[82] C.C. Tappert, C.Y. Suen, and T. Wakahara. The state of the art in on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:787–808, 1990.

[83] D.J. Wallace. Memory and learning in a class of neural models. In *Proceedings of the Workshop on Lattice Gauge Theory*, 1986.

[84] B. Widrow and S.D. Stearns. *Adaptive Signal Processing.* Prentice-Hall, New York, 1985.

[85] K.D. William. Theory and implementation of network representation of knowledge - application to character recognition. Ph.D. Thesis, University of Berkeley, 1979.

[86] C.L. Wilson and M.D. Garris. *Handprinted character database.* National Institute of Standards and Technology, 1990.

[87] H.J. Wolfson. On curve matching. *IEEE Transactions on PAMI*, 12:483–489, 1990.

[88] W. Xu and C. Wang. Cgt: a fast thinning algorithm implemented on a sequential computer. *IEEE Transactions on Systems, Man and Cybernetics*, 17(5):847–851, 1987.

[89] M. Yoshida and M. Eden. Handwritten chinese character recognition by an analysis by synthesis method. *Proceedings of 1st International Conference on Pattern Recognition*, 197-204, 1973.

[90] T.Y. Zhang and C.Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27:236–239, 1984.