# COURTESY AMOUNT RECOGNITION USING A FEEDBACK-BASED SEGMENTATION ALGORITHM

Wu Ding

A Thesis
In
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
At Concordia University
Montreal, Quebec, Canada

August 2008

*Your file Votre référence*
*ISBN: 978-0-494-45459-6*
*Our file Notre référence*
*ISBN: 978-0-494-45459-6*

# ABSTRACT


COURTESY AMOUNT RECOGNITION USING A FEEDBACK-BASED SEGMENTATION ALGORITIHM

Wu Ding

Paper checks play an important role in our modern commercial society. However, Reading paper checks and typing into computers manually is a time-consuming and labor-intensive task. Therefore, accurate and efficient Check Reading Systems (CRS) are in high demand by banks.

In this thesis, a CENPARMI CRS is introduced. Its courtesy amount recognition module is then presented in depth. The module makes use of a new feedback-based segmentation algorithm. The original segmentation algorithm can segment the inputted numeral string only once. The new algorithm can adjust the parameters of the segmentation algorithm according to the feedback information from the digit recognizer and can re-segment the inputted numeral string multiple times if necessary.

Two rejection strategies in the new segmentation algorithm are also presented. The first strategy is used for rejecting unreasonable segmentations, which can be accepted by the individual digit recognizer. The second strategy is used to avoid accepting a digit string as one digit.

Moreover, three algorithms in pre-processing and post-processing modules are proposed: a new border noise removal algorithm, an extra punctuations removal algorithm, and an implicit decimal points detection algorithm.

Finally, a Convolutional Neural Network (CNN) recognizer is integrated into the CRS to recognize "00" and "000" directly in order to avoid segmenting "00" and "000".

The experiment, based on the Bell Quebec check database, has shown that the courtesy amount recognition rate of the CRS has improved from 41.2% to 74.3%. The CRS can produce preliminary courtesy amount recognition results, which can be used for validation with the legal amounts to suppress the substitution rate.

# Acknowledgements

First and foremost, I would like to take this opportunity to express my sincerest thanks to my supervisors, Dr. Ching Y. Suen and Dr. Adam Krzyzak for providing a great research and study environment. I could not finish this work and my thesis without their support, encouragement and patience. I feel extremely fortunate to have been guided by them.

I specifically would like to thank Dr. Didier Guillevic, Dr. Jun Zhou, Mr. Wumo Pan Mr. Nicola Nobile, Ms. Shira Katz for their help and time. I learned a lot from the bi-weekly meetings with Dr. Didier Guillevic. Without the help from Dr. Jun Zhou, it would have been much more difficult to finish the CRS migration work. The discussions with Mr. Wumo Pan inspired me with many interesting ideas. Mr. Nicola Nobile took much time to find databases, legacy codes and documents for me. His efforts are very important for my research. Ms. Shira Katz made considerable efforts to proofread this thesis.

I would also like to thank my all other colleagues and friends: Chunlei He, Dr. Jianxiong Dong, Xiaoxiao Niu, Ying Li, Marleah Blom, Guiling Guo, Yun Li, Dr. Qizhi Xu, Dr. Javad Sadri, Izadi Nia Sara, Bo Jiao, Ya Tang, Liqiang Zhang, who gave me their help and shared the good times with me.

Finally, I would like to thank my parents, sister and brother for their selfless love. I feel very lucky to live in such a warm family. Without their continuous care and encouragement, it would have been impossible for me to finish my work.

# CONTENTS

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Optical Character Recognition (OCR) techniques have been developed for over 50 years. Although e-commerce, such as e-checks and e-forms, is becoming more and more popular, paper forms are still widely used and cannot be replaced completely in the near future. Therefore, robust OCR applications are still in high demand for commercial purposes. With the developments in the fields of artificial intelligence, pattern recognition, and image/video processing, many research and commercial OCR applications are becoming more powerful, and are gradually closing the gap between human beings and computers.

OCR applications can be divided into two main categories: on-line character recognition and off-line character recognition. On-line character recognition automatically converts the handwritings into text at the same time as they are written. The data obtained is regarded as a dynamic representation of handwriting. The information about the order and direction of strokes makes on-line character recognition results more accurate. Many computer technology corporations, such as Microsoft and

IBM, have released commercial products in online character recognition. These are used as input methods for tablet PCs, laptops, and desktops.

Off-line character recognition is much more complex because no information about the order and direction of strokes are available. Off-line character recognition can be divided into machine-printed character and handwritten character recognition. Machine-printed characters are produced in clear and standard ways and most of them are either not connected or connected in simple ways. Therefore, machine-printed character recognition is a relatively easy problem and has almost been solved. However, handwritten character recognition is a much more complex and difficult task due to enormous variations in writing styles and difficulty in segmenting touched characters. It has not yet been dealt with in a completely successful manner. Limiting the vocabulary of handwritten texts or setting the position of each character can make the recognition easier. There are many different applications in handwritten character recognition, such as mail address recognition [8], tax form recognition [9] [10], bank check recognition [11] [12] [13] [17], and census recognition [14]. Among the most important applications is the automatic bank Check Reading System (CRS), which has been used in many commercial banks.

The paper check is the most popular form for non-cash payment in North America. Nearly 68 billion checks are processed every year [13]. Reading the amounts on checks and entering the data into the computer manually (by human operators) is highly labor-intensive, time-consuming, and therefore expensive. The cost of processing one check manually is from 2.78 USD to 3.09 USD [15]. Therefore, the banking industry has serious need for a robust and efficient automatic bank check reading system, which has at least the same recognition rate as human operators with possibly lower error rate (<1%) than them.

A bank check reading system is a very complex system which uses a lot of image processing, pattern recognition, and artificial intelligence techniques in order to deal with immense variations in handwritten numeral strings and words. The shapes and formats of handwritten numeral strings and words could be considerably different according to different persons, countries and languages. Some complete bank check reading systems have been developed, such as A2iA CheckReader [11] [16], LIREC [43], Parascript [27], Unisys [52], and Orbograph [53]. The A2iA CheckReader is available in five languages (and could be adapted to different nations: English, French, Italian, Portuguese, and Spanish). There are also bank check reading systems for Korean checks [18], Chinese checks [19], French checks [11], U.S. checks [20], German checks [21], Brazilian checks [22], and Japanese checks [23].

There are two main items on one check which need to be recognized: the courtesy amount which is written in Arabic numeral expression form; and the legal amount which is written in words. These two parts are tightly related because the courtesy amount and the legal amount should indicate the same amount, or the check would be invalid. The main task of a bank check reading system is to recognize the courtesy amount and the legal amount, and then make sure they indicate the same amount if needed. Some bank check reading systems also recognize the date. Figure 1 is a sample of a bank check image, and the courtesy amount, legal amount, and date on the check are illustrated.

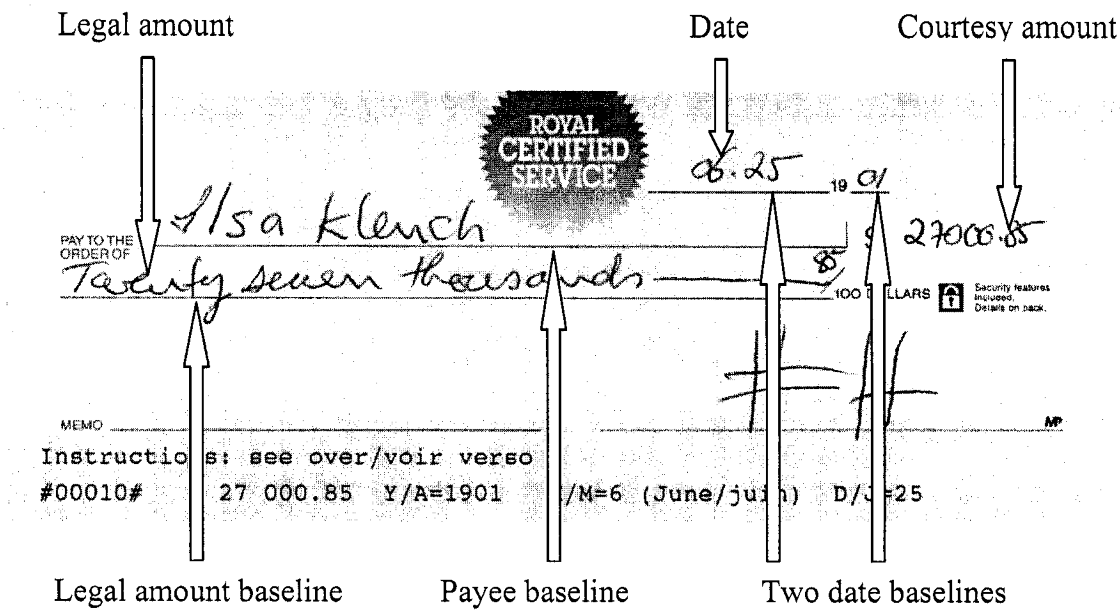



Figure 1. One sample of a bank check.

Although different bank check reading systems have different designs, they generally have the following basic modules: image scanning, pre-processing, extracting pertinent areas, legal amount recognition, courtesy amount recognition, post-processing, validation of legal amount recognition and courtesy amount recognition, acceptance or rejection of the recognition result. Figure 2 describes a typical structure of a bank check reading system. A paper check is scanned into a computer by a scanner; the check image is pre-processed, which includes noise removal, skew correction, slant removal, and binarization; then the item fields are extracted such as the legal amount, the courtesy amount, and the date; these items are recognized; the recognition results are post-processed; the recognition results of legal amount and courtesy amount are validated to generate a last result; the recognition result is accepted or rejected.

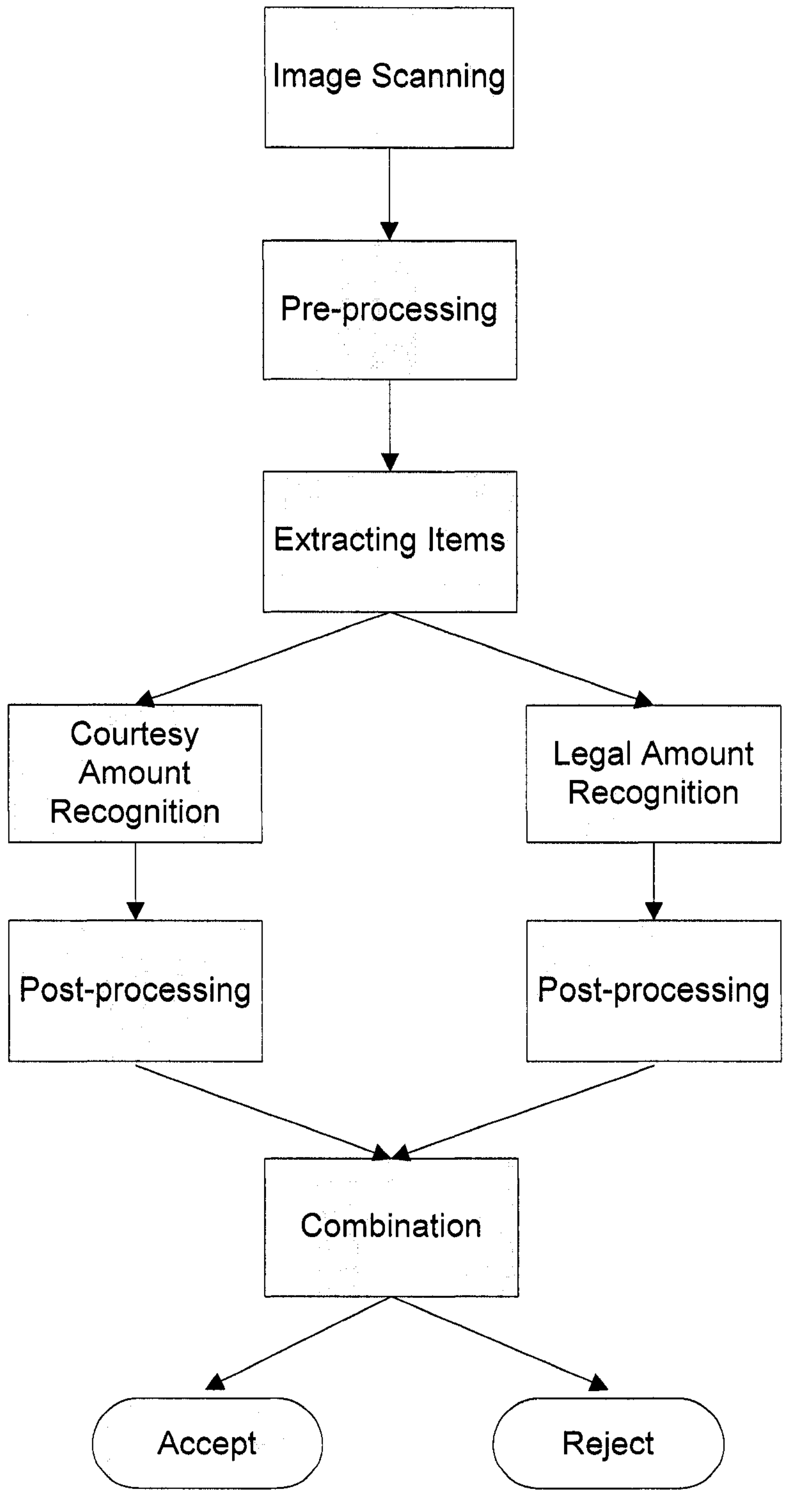



Figure 2. General structure of a bank check reading system.

## 1.1 The Challenge and the Motivation

Although the official value of a check is the legal amount which is written in words, the human operators in banks often only read the courtesy amount and ignore the legal amount because it is much easier to read the courtesy amount. They read the legal amount only when they are not sure of the courtesy amount. Similarly, many bank check reading systems mainly read the courtesy amount. Although the recognition rate of the legal amount is comparatively low, it can still be used to provide information to improve the recognition rate of the courtesy amount and to suppress its substitution rate, which is called the validation procedure.

Although recognition of the courtesy amount is not as difficult as that of the legal amount, it is still a challenging task. In a machine-printed numeral string, the shapes of the same digit are very similar; most of the digits are separate; the touching digits are connected in a simple way. In a handwritten numeral string, however, the situation becomes much more complex: sometimes the shapes of the same digit are quite different; most of the digits may be connected, and possibly connected in a complex way; some digits can be broken; and there maybe be some extra strokes in the string which are only used for connection. Moreover, a courtesy amount is more complicated than a common handwritten numeral string: some non-digit characters, such as "/", "×", and"_", can be seen in a courtesy amount; and the decimal point may be omitted. Multiple recognizers which are trained with a large amount of data can deal very well with the recognition of handwritten isolated numerals. Therefore, the main challenge of courtesy amount recognition lies in the correct segmentation of connected numeral strings into valid individual digits.

The digits in a courtesy amount can be connected in many different and complex ways and have different formats, which make segmentation difficult. Some samples of courtesy amounts are shown in Figure 3:

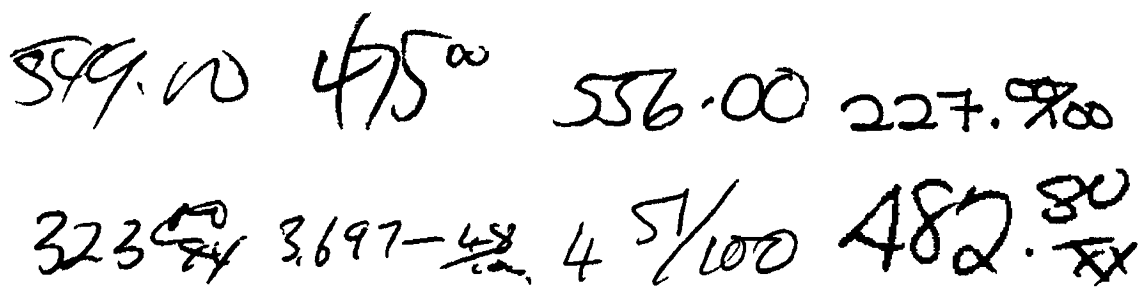


Figure 3. Some samples of courtesy amounts.

The original courtesy amount segmentation algorithm [1] used by CENPARMI check reading system works as follows: first, finding Significant Contour Points (SCPs) by analyzing the contour of a courtesy amount; then, finding possible segmentation paths between SCPs according to some pre-defined rules; next, evaluating these segmentation paths by using an individual digit recognizer; finally, generating a segmentation paths list in order of the confidence score of each path. However, if correct segmentation points are not included among SCPs or do not follow the pre-defined rules, this algorithm will fail.

This thesis focuses on how to improve the performance of the original courtesy amount segmentation algorithm, which is the main contribution of my thesis. Other problems which influence the performance of courtesy amount recognition of CENPARMI CRS are also discussed and corresponding algorithms are proposed to deal with them. Moreover, a new convolutional neural network recognizer for "00" and "000" numeral strings is presented to improve the recognition rate of "00" and "000".

## 1.2 State of the Art

In this section, we will review the state of the art of courtesy amount segmentation algorithms. Many researchers have developed algorithms to segment handwritten numeral strings or courtesy amounts and have made some progress.

C. R. Nohl [45] used a sliding window that moves along the image horizontally to get a sequence of rectangular sub-images, which were determined by an estimated number of digits and the image length. This method is an implicit segmentation algorithm. Obviously, this method cannot handle complex segmentation paths.

Choi and Oh [34] proposed a method to recognize handwritten numeral strings without segmentation. A neural network digit recognizer is trained with a database of two touching numerals, which include all permutations of two digits from 0 to 9. Its drawback is that it can only deal with two touching numerals. Moreover, because the number of images for each permutation is limited, the performance of the recognizer is not robust. Figure 4 shows some samples of two touching numerals in their training database:

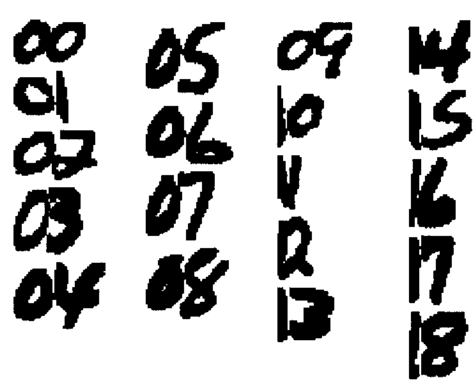


Figure 4. Some samples of two touching numerals in the training database of [34].

Suwa [44] proposed a method which makes use of the graph theory and some heuristic rules to find segmentation candidates. The handwritten numeral string is thinned and the result is represented as a graph in the form of vertices and edges. So segmentation becomes grouping the vertices and edges of the graph into disconnected sub-graphs. According to some heuristic rules and a score function, a single shared horizontal/vertical edge, or a shortest path between the deepest point of the upper valley and the highest point of the lower hill is determined as a segmentation path. A procedure for ligature elimination is also employed during segmentation. This method is an example of a recognition-free segmentation algorithm. The samples of a shared horizontal edge, a shared vertical edge, and a shortest path between the deepest point of the upper valley and the highest point of the lower hill are shown in Figure 5:


(a) Finding a shared horizontal edge

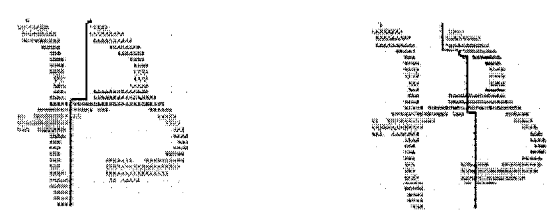
(b) Finding a shared vertical edge

(c) Finding the shortest path

Figure 5. Three samples of using the segmentation procedure of Suwa's method [44].

Congedo et al. [46] proposed a "drop-falling" segmentation algorithm based on the numeral string's contour. Its main idea is described as follows: first, find a starting point; then, simulate a "drop-falling" process to find a possible segmentation path, which acts as a drop of water that drops downward along the numeral string's contour from the starting point; finally, the path that a drop of water passes along is a possible segmentation path. There are four types of "drop-falling" procedures: descending-left algorithm, descending-right algorithm, ascending-left algorithm, and ascending-right algorithm. Descending/ascending and left/right are the movement directions of a drop of water. The essence of these four "drop-falling" procedures is the same. They only differ in the orientation of the starting points and the movement direction. The positions of starting points are very critical for these algorithms. One of the results produced by these four "drop-falling" procedures could be correct. Different results of the four drop-falling procedures are shown in Figure 6:

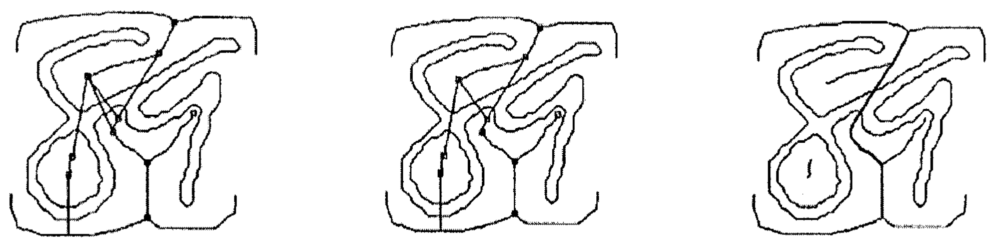
(a) Ascending-left (b) Ascending-right



(c) Descending-left (d) Descending-right

Figure 6. One Sample of different results of the four drop-falling procedures [46].

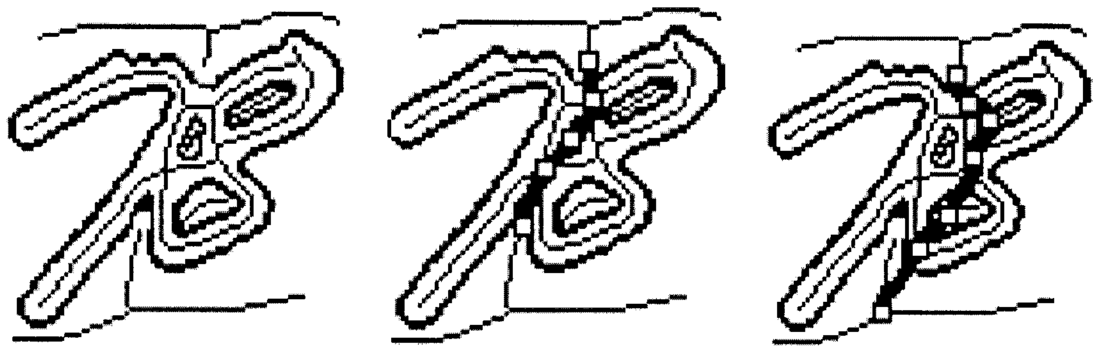
Lu et al. [47] proposed a two-digit numeral string segmentation algorithm which is based on the thinning of background regions. It first finds some feature points in the background skeleton of a numeral string image; then constructs segmentation path candidates between these feature points; then path candidates are preliminarily ranked by fuzzy rules; finally, top ranked path candidates are evaluated by a recognizer and the best one is chosen according to an acceptance criterion. One sample using this algorithm to segment a two-digit numeral string is shown in Figure 7:



(a) (b) (c)

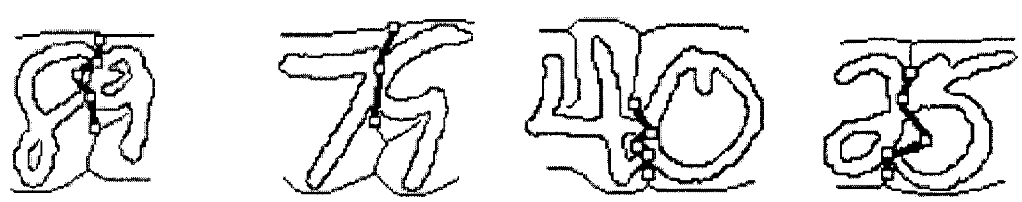
Figure 7. One sample using the algorithm of [47] to segment a two-digit numeral string: (a) All possible segmentation paths, (b) Top ranked segmentation paths, and (c) Final result.

Chen et al. [48] proposed a segmentation algorithm which is similar to that of Lu et al. [47], but it is based on both background analysis and foreground analysis. It first thins the background and foreground of the image of a connected numeral string; then, feature points are extracted from the background and foreground skeletons; then, some possible segmentation paths are constructed and useless strokes are removed; finally, the best segmentation path is determined by using a mixture of the Gaussian probability function to analyze the geometric properties of each segmentation path. One sample of finding possible segmentation paths is shown in Figure 8. Some samples of applying this algorithm on images of connected numeral strings are shown in Figure 9:



(a) (b) (c)

Figure 8. An example of finding possible segmentation paths in [48]: (a) Image after foreground and background thinning, (b) A possible segmentation path, and (c) Another possible segmentation path.



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

Figure 9. Some samples of separating numeral strings [48].

Strathy [1] proposed a segmentation algorithm which is applied to find Significant Contour Points (SCPs) based on the analysis of the outer contours of the connected handwritten numeral string. SCPs are a set of likely entry/exit points on the outer contour which are located on the discontinuities in the outer contour, which can be found by a corner-finding procedure. This algorithm assumes that a separating path might enter and exit the connected numeral string through any two SCPs. Therefore, segmentation path candidates are generated across two SCPs and ranked in a paths list by an evaluation function based on the features of SCPs. Finally, the paths on the top of the paths list will be evaluated again by an individual digit recognizer to determine the best path. Some samples of applying this algorithm are shown in Figure 10:

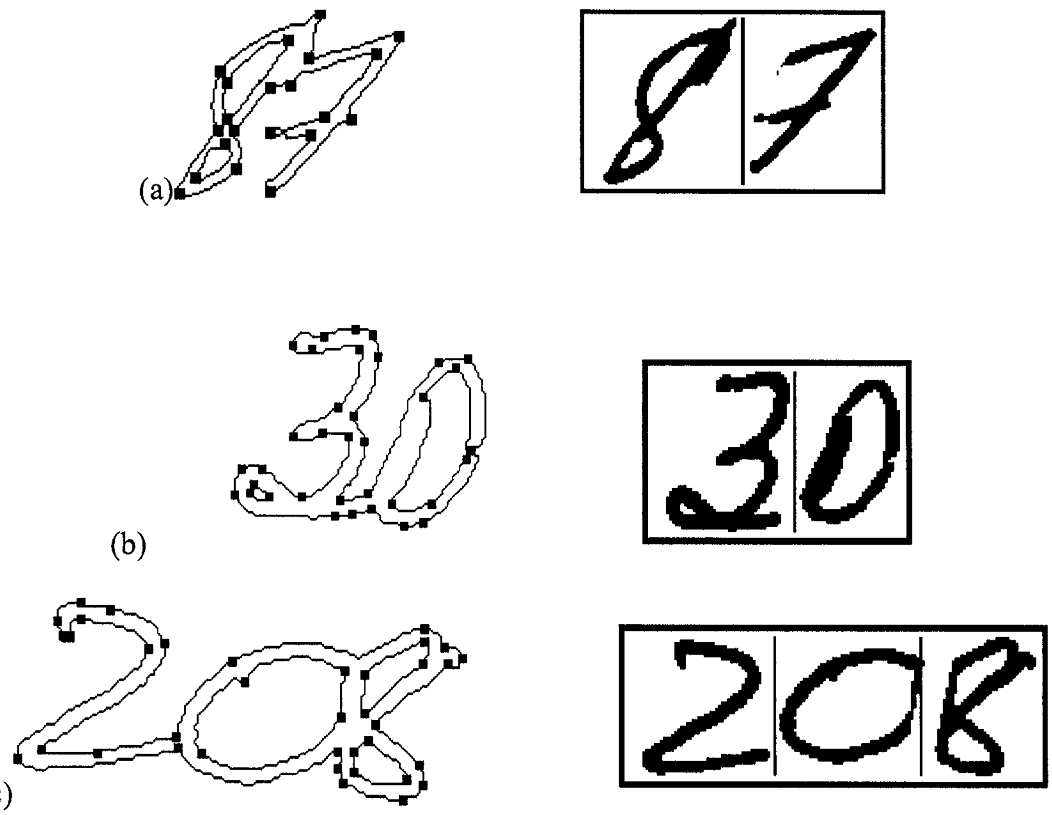

(a) (b) (c)

Figure 10. Some samples of SCPs and final segmentation results in [1].

Other researchers have used methods that combine multiple segmentation algorithms together to improve the segmentation performance: Dimauro et al. [12] combined the drop-falling algorithm and SCPs-based algorithm; Palacios et al. [13] combined three segmentation algorithms: a Min-Max contour analysis, Hybird Drop Fall, and Extended Drop Fall algorithms.

In this thesis, a new feedback-based segmentation algorithm which is based on the Strathy's SCP-based algorithm is proposed. A detailed description of this algorithm will be provided in chapter 3.

## 1.3 Outline of Thesis

This thesis is divided into 7 chapters:

- In Chapter 1, the motivation and objectives of this thesis have been described. The state of the art in the courtesy amount segmentation and Check Reading System (CRS) have also been reviewed.
- In Chapter 2, the history of CENPARMI CRS and its constituted modules are introduced. Then, the migration work of CENPARMI CRS from Unix to Windows is reviewed. Finally, the three main modules: courtesy amount recognition, legal amount recognition and date recognition are illustrated step by step.
- In Chapter 3, the original segmentation algorithm in CENPARMI CRS is firstly reviewed. Then, a new feedback-based segmentation algorithm, which is the main contribution of my thesis, is proposed. Three other methods used in the segmentation algorithm are also proposed: a method for rejecting unreasonable segmentations, a method for rejecting digit strings, and a method for dealing with non-connected digit "5".
- In Chapter 4, the methods used in pre-processing and post-processing modules are proposed: a method for removing border noise, a method for removing extra punctuations in the recognition results, and a method for detecting implicit decimal points.

- In Chapter 5, firstly, a method for integrating the original recognizer for "00" into CENPARMI CRS is described. Then, a new CNN recognizer for "00" and "000" is proposed, whose performance is much better than that of the original "00" recognizer.
- In Chapter 6, some experiments are conducted to test the performance of new CENPARMI CRS. Recognition errors of the CRS are categorized and analyzed.
- In Chapter 7, the thesis is summarized and potential future work is discussed.

# Chapter 2

# CENPARMI Check Reading System

## 2.1 Original CENPARMI Check Reading System

Every day bank checks are processed in commercial banks. Processing bank checks manually is very expensive. In order to save check-processing costs, a Check Reading System (CRS) is used to automatically read bank checks in place of human operators. The design of a good CRS is very difficult because it should read checks written by different people in diverse writing styles with a high reliability and a low substitution rate (less than 1%). Some commercial products, such as A2iA CheckReader, Ocbograph, and Unisys, have been used in commercial banks.

The CENPARMI CRS was originally developed in the 1990s by many researchers, including Nicholas W. Strathy [1], Ke Liu [2], X. Y. Ye [3], J.H. Kim [4], and D. Guillevic [5]. It was maintained and enhanced by Jun Zhou [6] and Qizhi Xu [7]. Its functionalities include Legal Amount Recognition (LAR), Courtesy Amount Recognition (CAR), and Date Recognition (DR).

The CENPARMI CRS consists of seven modules: image scanning, pre-processing, legal amount recognition (LAR), courtesy amount recognition (CAR), date recognition (DR), post-processing, accept or reject the recognition results. Its entire structure is shown in Figure 11:

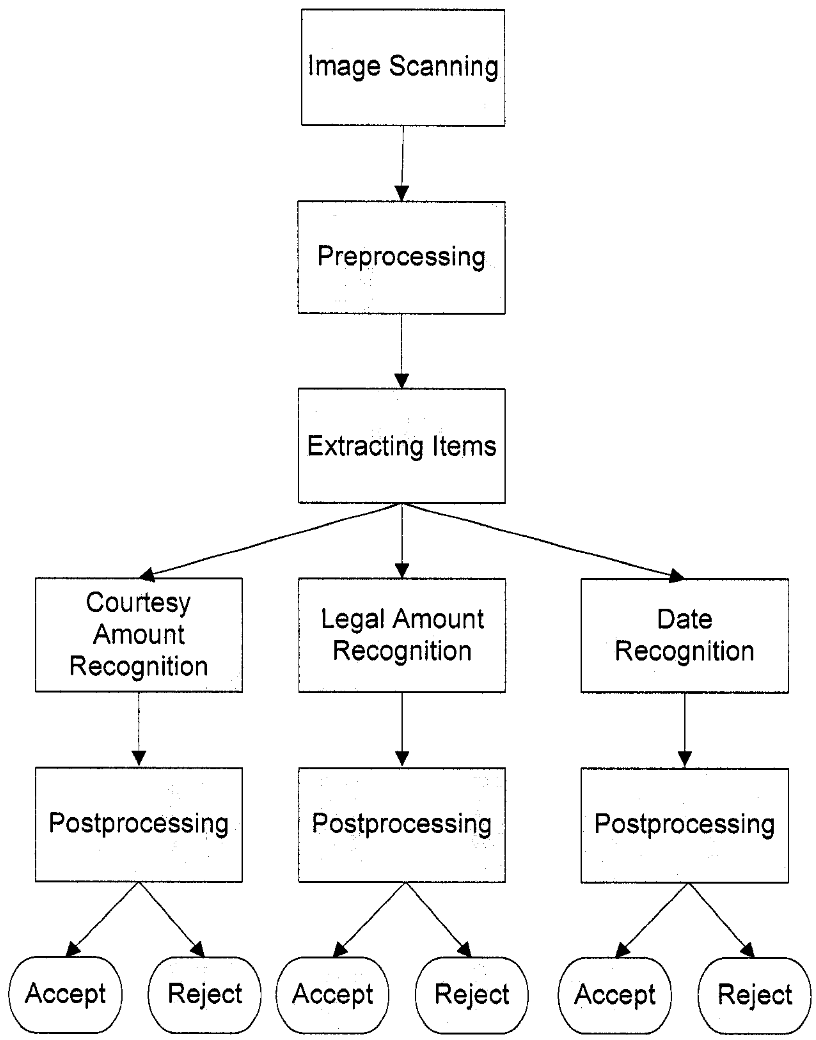



Figure 11. Structure of the CENPARMI CRS.

The CRS's CAR [1] module is implemented by one segmentation algorithm which is based on finding significant contour points [1], and one artificial neural network digit recognizer [32]. Its LAR module [6] is implemented by one segmentation algorithm which decides on a threshold for inter-word gaps, and one artificial neural network character recognizer. The DR module [7] is implemented by a date layout recognition algorithm, and a hybrid recognizer, which is composed of two separate recognizers: one artificial neural network recognizer and one HMM recognizer.

In the following chapters of this thesis, the/our CRS/recognizer is referred to CENPARMI CRS/recognizer. The thresholds used in the algorithms at this thesis are determined by experiments. The explanation of the format of courtesy amount, legal amount, and date recognition results of CENPARMI CRS are referred to Appendix A.

## 2.2 Migration Work

The original CENPARMI CRS was implemented on a Sun Unix workstation. In 2005, it was decided to migrate it from the Unix platform to a Windows platform in order to enhance its application in the contemporary data processing environment.

Five main problems were encountered during migration: (1) a new Graphic User Interface (GUI) needed to be developed because Windows platform uses a totally different GUI library and style; (2) source codes had to be rewritten for using a new scanner and corresponding scanner driver; (3) conversion from big-endian data to little-endian data, which requires reversing the order in which a sequence of bytes are stored in a computer's memory; (4) various errors due to incompatibility between Unix and Windows platforms had to be corrected; and (5) bugs in the original codes, which sometimes caused the system breakdown, had to be located and corrected.

## 2.3 New CENPARMI CRS

In this section, the GUI (Graphics User Interface) of the new CRS and its three main modules: legal amount recognition, courtesy amount recognition, and date recognition, will be described and illustrated respectively.

### 2.3.1 GUI of New CENPARMI CRS

The main menu of the new CENPARMI CRS is illustrated in Figure 12:

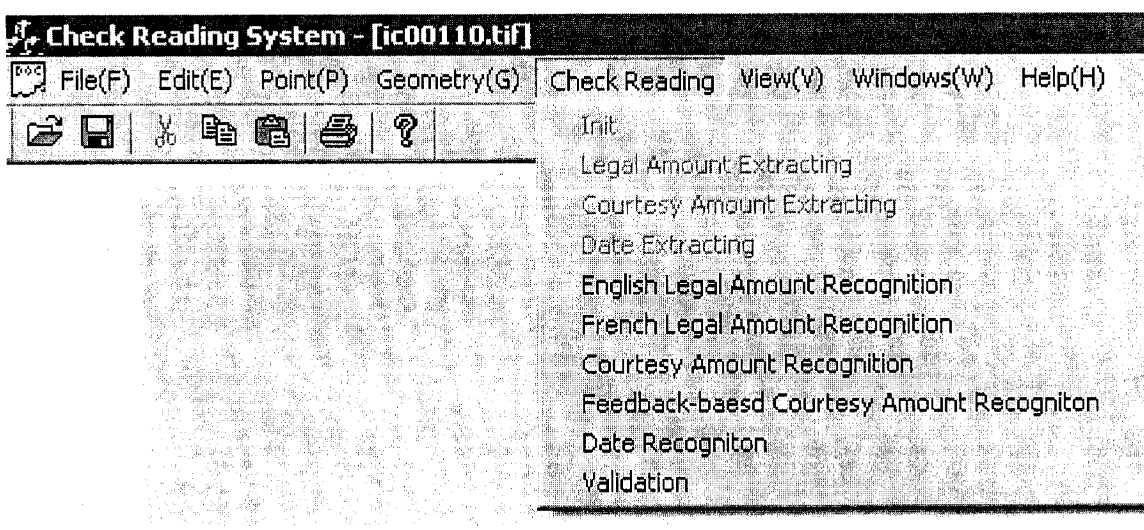



Figure 12. New CRS GUI.

After clicking any recognition operation from the menu, such as courtesy amount recognition, legal amount recognition, or date recognition, the recognition results will be shown on the interface immediately.

A snapshot of the recognition results of courtesy amount, legal amount and date are illustrated in Figure 13:

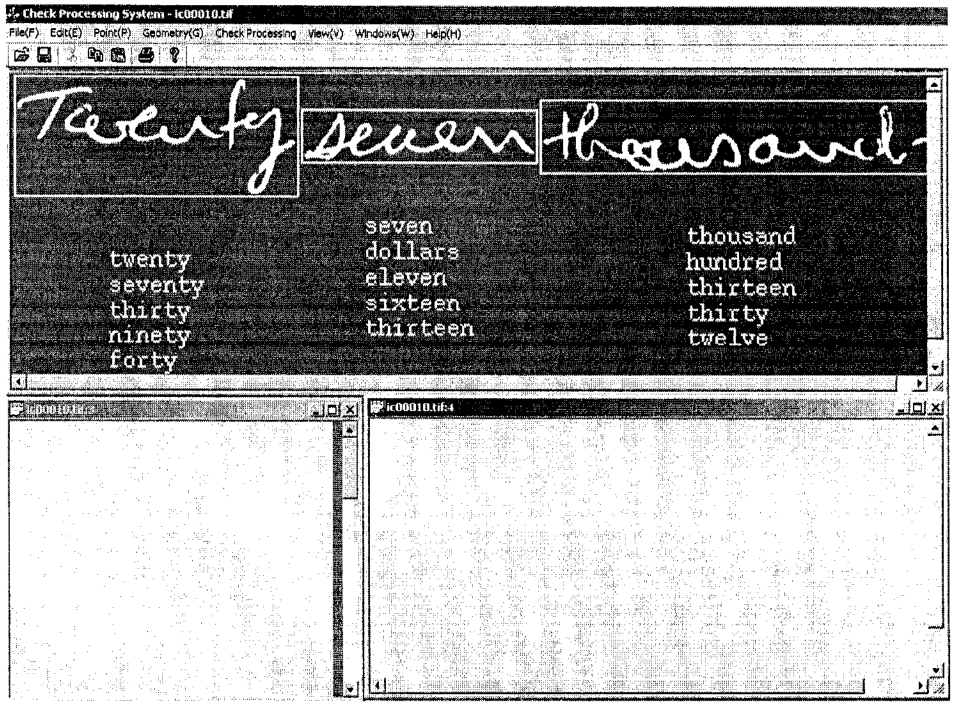



Figure 13. Snapshot of check recognition result of new CRS.

### 2.3.2 Three Main Modules of New CENPARMI CRS

In this section, the procedures of three main modules in the new CENPARMI CRS will be described and illustrated in detail: legal amount recognition, courtesy amount recognition, and date recognition. Each module consists mainly of two steps: (1) item extraction; (2) segmentation and recognition.

- **Legal amount recognition [2][6]**

1. Item extraction [2]

    a. The legal amount is extracted as the following procedure: (1) locate the legal amount baseline; (2) locate the payee baseline; (3) extract the region which is bounded by the legal amount and the payee baselines as shown in Figure 14:

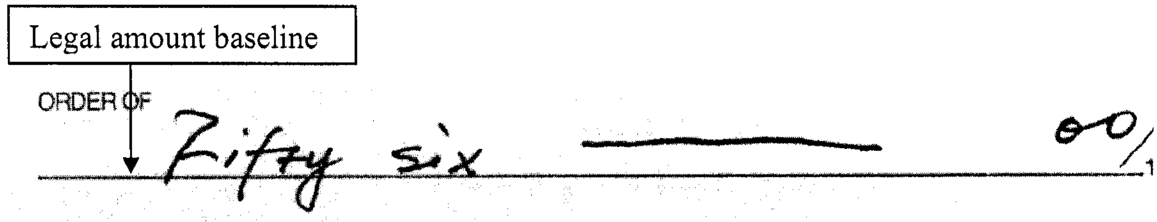



Figure 14. Legal amount region.

b. The legal amount baseline is removed as shown in Figure 15:

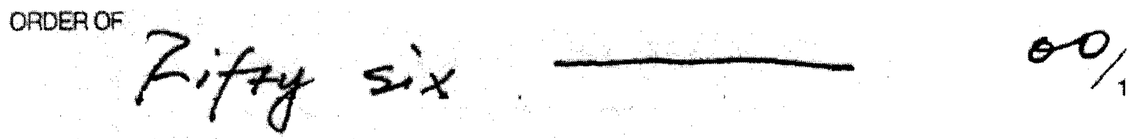



Figure 15. Legal amount after removing the legal amount baseline.

c. The legal amount is binarized with result shown in Figure 16:

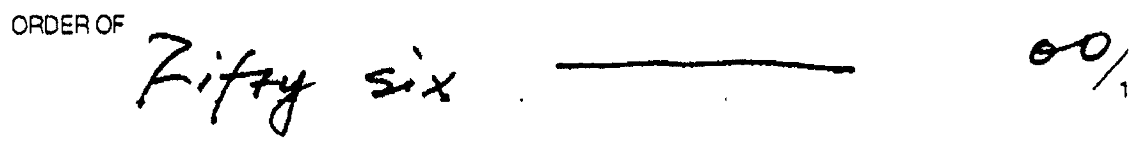



Figure 16. Legal amount after binarization.

d. Noise and some printed characters such as "ORDER OF", are removed as shown in Figure 17:

Figure 17. Legal amount after removing printed characters.

e. Finally, the cents part of the image is removed to result in the final legal amount item in handwritten English words as shown in Figure 18:

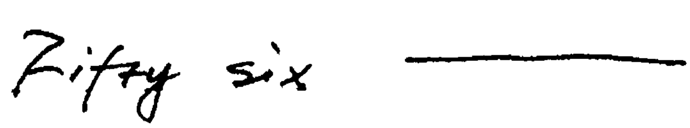


Figure 18. Legal amount after removing its cent part.

2. Segmentation and recognition [6]

a. The image is smoothed, and the handwritten horizontal lines (shown in Figure 18) at the beginning and the end of the string are removed with result shown in Figure 19:

Figure 19. Legal amount after removing the handwritten horizontal lines.

b. The slant is corrected, and the image is cropped as the bounding box of the legal amount as shown in Figure 20:

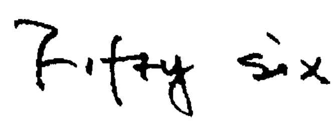


Figure 20. Legal amount after slant correction and cropping operation.

c. The segment, re-segment, and recognition operations are performed:

(1) the legal amount is under-segmented according to an inter-word gap threshold, which is selected as a portion of the largest Euclidean distance between the connected components; (2) the under-segmented result of the legal amount is re-segmented according to the feedback of a neural network word recognizer and the analysis of the inter-word and inter-character gaps; (3) the words segmented from the legal amount are recognized by the neural network word recognizer. The recognition results of each handwritten word, which are listed below it in decreasing order, are shown in Figure 21:

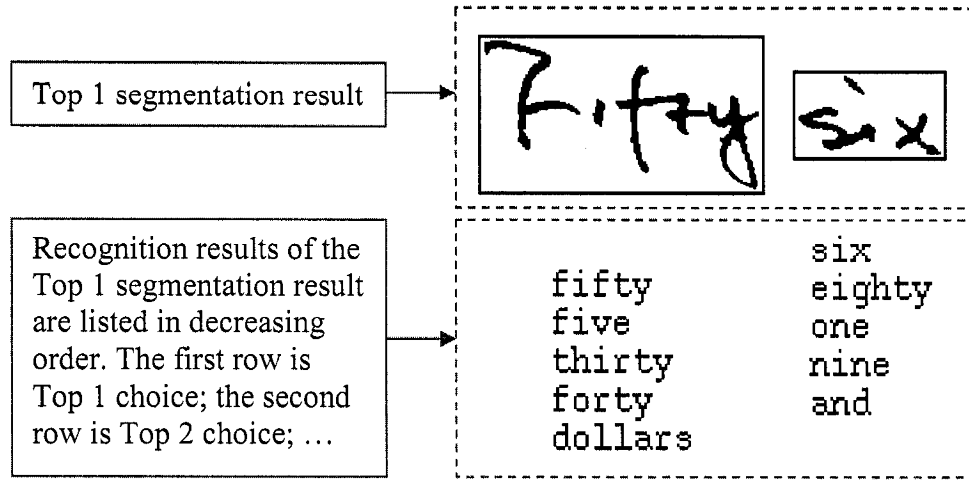



Figure 21. Recognition results of the legal amount.

- **Courtesy amount recognition [1][2]**

1. Item extraction [2]

    a. Courtesy amount is extracted as the following procedure: (1) locate the legal amount baseline; (2) locate the payee baseline; (3) locate the dollar sign $ on the right of the legal amount and payee baselines; (4) extract the region which is on the right of the dollar sign $ as shown in Figure 22:

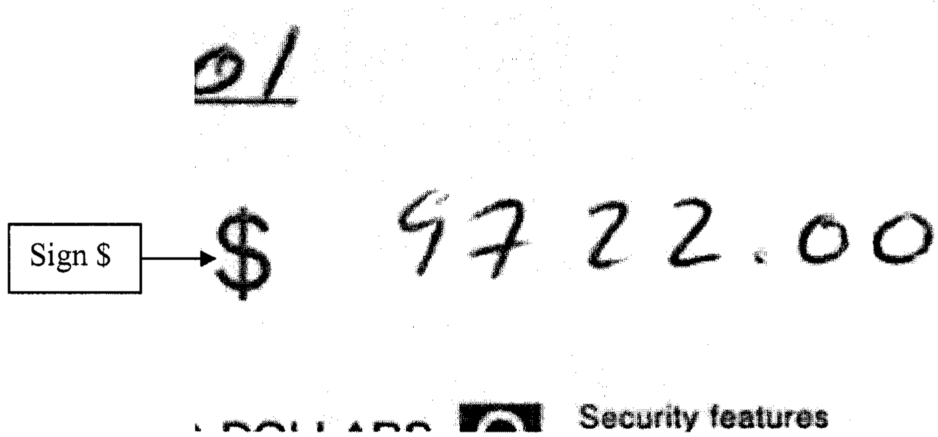



Figure 22. Courtesy amount region.

    b. Then, the courtesy amount is binarized and cropped as the following procedure: (1) determine a threshold for the courtesy amount; (2) binarize the courtesy amount region; (3) the image is cropped as the bounding box of the courtesy amount as shown in Figure 23:

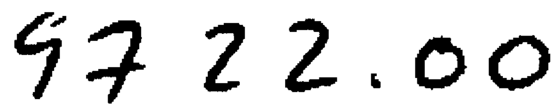


Figure 23. Courtesy amount after binarization and cropping operation.

2. Segmentation and recognition [1]

   a. Segmentation and recognition operations are performed.

   (1) the courtesy amount is segmented into connected components; (2) some connected components within a pre-defined neighborhood are merged to create some new groups; (3) different segmentation paths are found for each connected component or group; (4) each segmented unit, connected component, and group are recognized by a neural network digit recognizer; (5) according to their confidence scores, the courtesy amount recognition results are searched from left to right and a final results list is generated. One example of the procedure is shown in Figures 24-26:

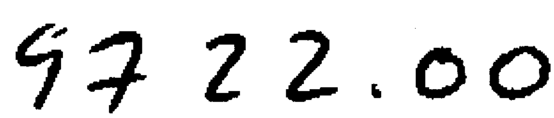


Figure 24. Extracted courtesy amount.

Figure 25. Eight connected components of the courtesy amount.

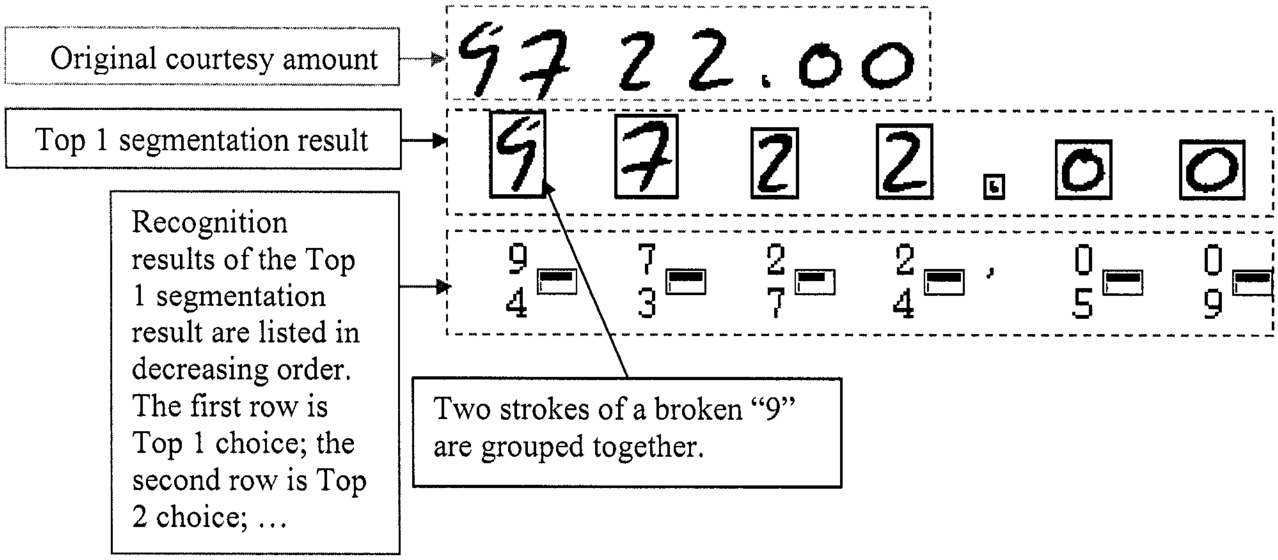



Figure 26. Recognition results of the courtesy amount. Note: (1) the recognition results of each digit are listed below it in decreasing order; (2) the first connected component (which looks like the left part of numeral 9) and second connected component "," are grouped together and recognized as "9".

- **Date recognition [2][7]**

1. Item extraction [2]

    a. The date is extracted as the following procedure: (1) locate two date baselines; (2) extract the region which is around two date baselines as shown in Figure 27:

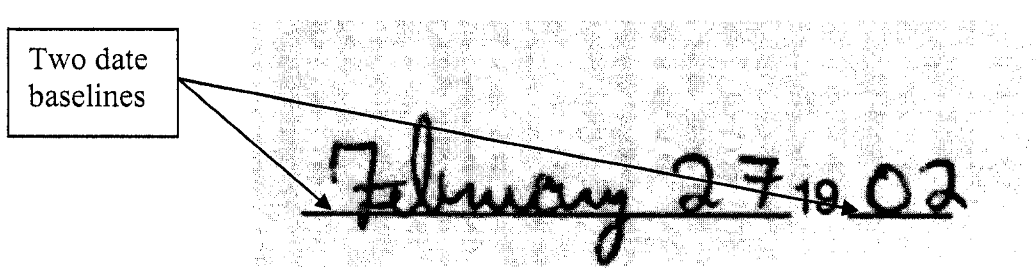



Figure 27. Date region.

    b. Two date baselines are removed as shown in Figure 28:

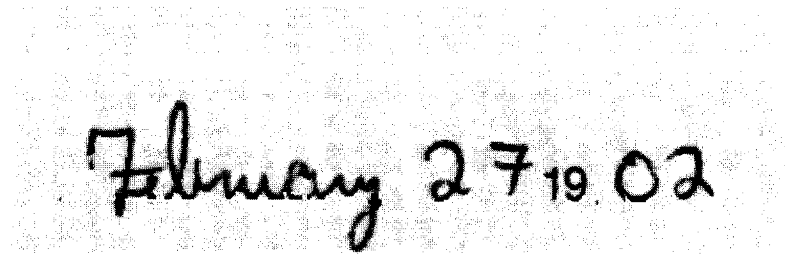


Figure 28. Date after removing two date baselines.

c. The image is binarized and its background noise is removed as shown in Figure 29:

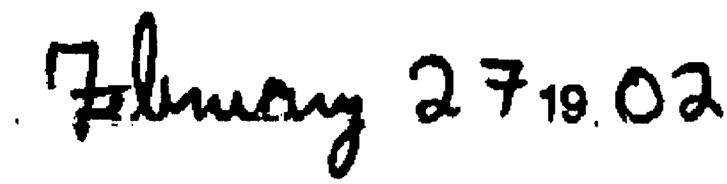


Figure 29. Date after noise removal and binarization.

2. Segmentation and recognition [7]

a. The “year” part of the date is extracted as shown in Figure 30:

Figure 30. The “year” of the date.

b. The “day” and the “month” parts are extracted as shown in Figure 31:

Figure 31. The “day” and the “month” of the date.

c. The form of year, month, and day parts are analyzed and processed by the appropriate recognizers. If both a neural network recognizer and an HMM recognizer are used, their results will be combined using a modified Product rule [7]. The results are shown in Figure 32:

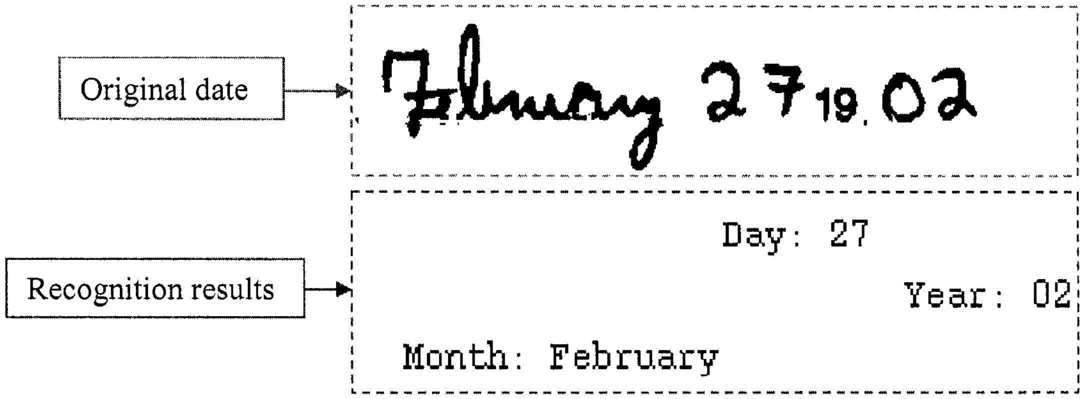



Figure 32. Recognition results of the date.

# Chapter 3

# Courtesy Amount Segmentation and Recognition

## 3.1 CENPARMI Courtesy Amount Recognition Algorithm

The courtesy amount is a handwritten numeral string which represents the amount of the check. Compared with legal amount recognition, courtesy amount recognition is a relatively easy problem. However, since the courtesy amount can have enormous variability in format, size and writing style, its recognition is still a difficult task.

Generally, there are two types of recognition procedures for numeral strings (e.g. courtesy amounts): segmentation-free recognition ([34] [35]) and segmentation-based recognition ([13] [33]) as illustrated in Figure 33:

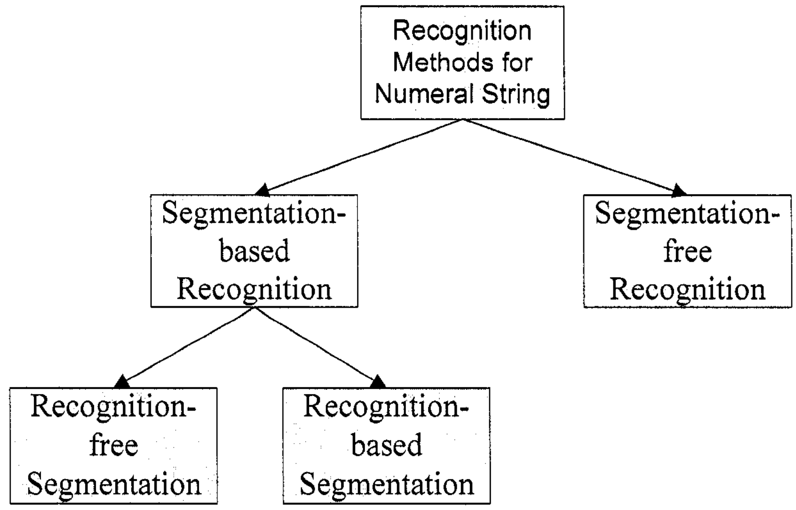



Figure 33. Classification of recognition methods for numeral strings.

With the procedure of segmentation-free recognition, a pair of touching numerals would be processed as one pattern and recognized by a holistic recognizer without segmentation. The drawback is that there are too many combinations of two numerals and it is not easy to collect enough data for each pair of touching numerals to train the recognizer.

On the other hand, with segmentation-based recognition ([11] [13]), the image of a numeral string is firstly segmented into individual digits; then the individual digits may be recognized or verified by a recognizer. The segmentation algorithm is a very crucial part of the procedure. It could be classified into recognition-free segmentation and recognition-based segmentation. With recognition-free segmentation [36], a numeral string is segmented into individual digits directly by a set of predefined rules without using a recognizer to verify the segmentation result; whereas for recognition-based segmentation [11], a numeral string is analyzed by the segmentation algorithm to generate some segmentation points. The candidate segmentation points are then used to generate segmentation paths, which are then verified by a recognizer to identify one optimal segmentation path.

In [37] and [38], recognition-free segmentation and recognition-based segmentation algorithms are combined in order to obtain better segmentation results.

The original CENPARMI courtesy amount recognition procedure [32] is segmentation-based recognition algorithm in which segmentation and recognition are combined. Given a grey level numeral string image, the procedure can be described as follows:

1) Binarization;
2) Removal of noise;
3) Extraction of the connected components by tracing contours using the algorithms in [39];
4) Removal of the spurious connected components by analyzing their positions and shapes;
5) Sorting of the connected components according to the average horizontal coordinates of all points on the contour;
6) Creation of groups of connected components in a pre-defined neighborhood;
7) Recognition of each connected component and group;
8) For each unrecognized connected component and group, splitting them to generate a possible segmentation results list and recognizing each result;
9) From left to right, search the connected components, groups, and their segmentation results list to generate the recognition results list with their confidence scores;

10) Post-processing, which includes syntactic analysis and the confidence score adjustments according to some pre-defined rules;

11) Sorting of the recognition results according to their confidence scores, in descending order;

12) Output the recognition results.

## 3.2 The Original Segmentation Algorithm

The original segmentation algorithm of courtesy amounts in the CENPARMI check reading system is based on the feature points, which are called Significant Contour Points (SCPs) [1]. SCPs are determined by the analyses of the outer contour of the connected numeral string and some pre-defined rules. One segmentation path is generated across two SCPs to separate the leftmost digit from the string. Given a set of SCPs, obviously there could be many segmentation path candidates between two SCPs in this set. In order to determine potential good pairs of SCPs, the importance score of each pair of SCPs is calculated by a function which has 9 feature measurements. A genetic algorithm is used to train 9 coefficients corresponding to these 9 feature measurements.

Given each connected component or group, the digit splitter tries to segment and recognize its leftmost digit. If recognition is successful, then it attempts to recursively segment and recognize the rest. The original segmentation algorithm of the courtesy amount is described as follows:

**Algorithm:** OriginalSegmentCourtesyAmount( image )

**for** (each connected component and group, from left to right, in the numeral string)

```
try to recognize it as a single digit;
if ( recognition is successful )  continue;
 set segmentation parameters;
 try to find SCPs and generate a list of segmentation paths;
 for ( each segmentation path in the list )
       split the image into two sub-images;
       try to recognize the left sub-image;
       if (  recognition is not successful )  continue;
       try to recognize the right sub-image;
       if (  recognition is not successful )
              OriginalSegmentCourtesyAmount( right sub-image ) ;
       save segmentation path;
   end for
end for
```

The key steps of the algorithm are shown in the flowchart of Figure 34:

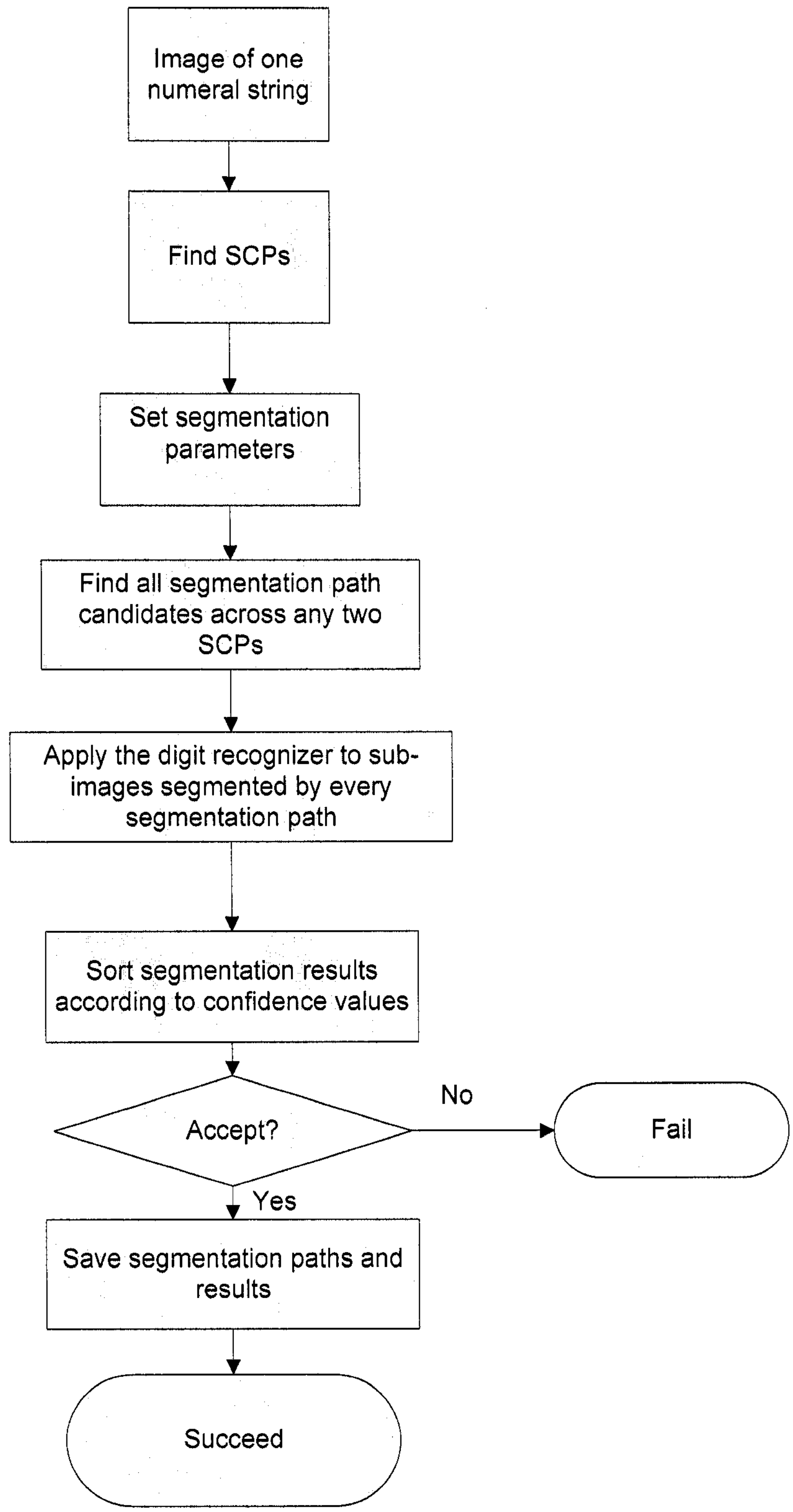



Figure 34. Flowchart of the original segmentation algorithm.

## 3.3 New Feedback-based Segmentation Algorithm [54]

According to Figure 34, the segmentation algorithm is used just once regardless of whether segmentation results are accepted or not. In fact, the segmentation algorithm could fail with one set of segmentation parameters and succeed with another set of parameters. So, it is necessary to re-segment with a different set of segmentation parameters when the segmentation algorithm fails with the first set. I propose a new feedback-based segmentation algorithm, which can adjust some parameters of the original segmentation algorithm based on the recognition results of segmentation path candidates. The algorithm can re-segment the connected numeral string multiple times until all combinations of segmentation parameters have exhausted or an accepted segmentation path candidate is found. The feedback-based segmentation algorithm is described as follows:

**Algorithm:** FeedbackBasedSegmentCourtesyAmount( image )

  **for** (each connected component and group, from left to right, in the numeral string)

      try to recognize it as a single digit;

      **if** ( recognition is successful ) **continue**;

      **for** ( each possible segmentation parameter combination in the list )

         try to find SCPs and generate a list of segmentations;

         **for (** each segmentation path in the list )

            split the image into two sub-images;

            try to recognize the left sub-image;

            **if** ( recognition is not successful ) **continue**;

            try to recognize the right sub-image;

```
            if (  recognition is not successful )
                  FeedbackBasedSegmentCourtesyAmount ( right sub-image ) ;
             save segmentation path;
         end for
          if ( segmenting is successful ) break;
     end for
end for
```

The description of key steps of the new feedback-based segmentation algorithm is shown in the flowchart of Figure 35:

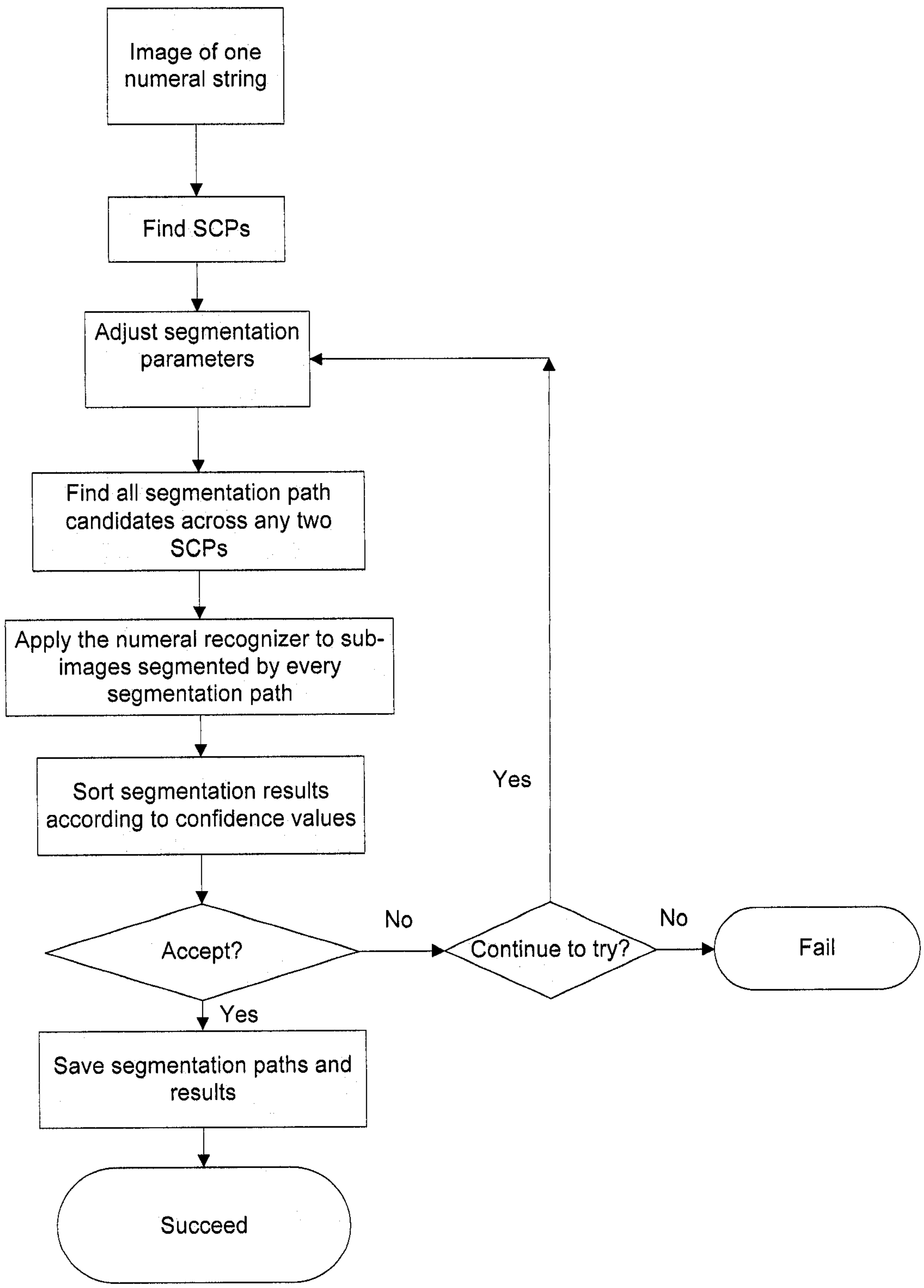



Figure 35. Flowchart of the new feedback-based segmentation algorithm.

There are three kinds of parameters which could be adjusted during the feedback phase:

1. The minimum ratio value of the height of the split sub-image and the height of the original image
2. The minimum complexity of the segmentation path
3. The maximum number of digits in the numeral string

By adjusting these three parameters, we can re-segment the connected numeral string in different ways and obtain different segmentation path candidates.

In order to show the difference between the original and new segmentation algorithms, these two algorithms have been applied to the same input numeral string in Figure 36, and Figure 37, respectively. Figure 36 shows how the original segmentation algorithm fails to correctly segment one input numeral string. Figure 37 shows how to re-segment the same string by the new feedback-based segmentation algorithm which adjusts the parameter "the maximum number of digits in the numeral string".

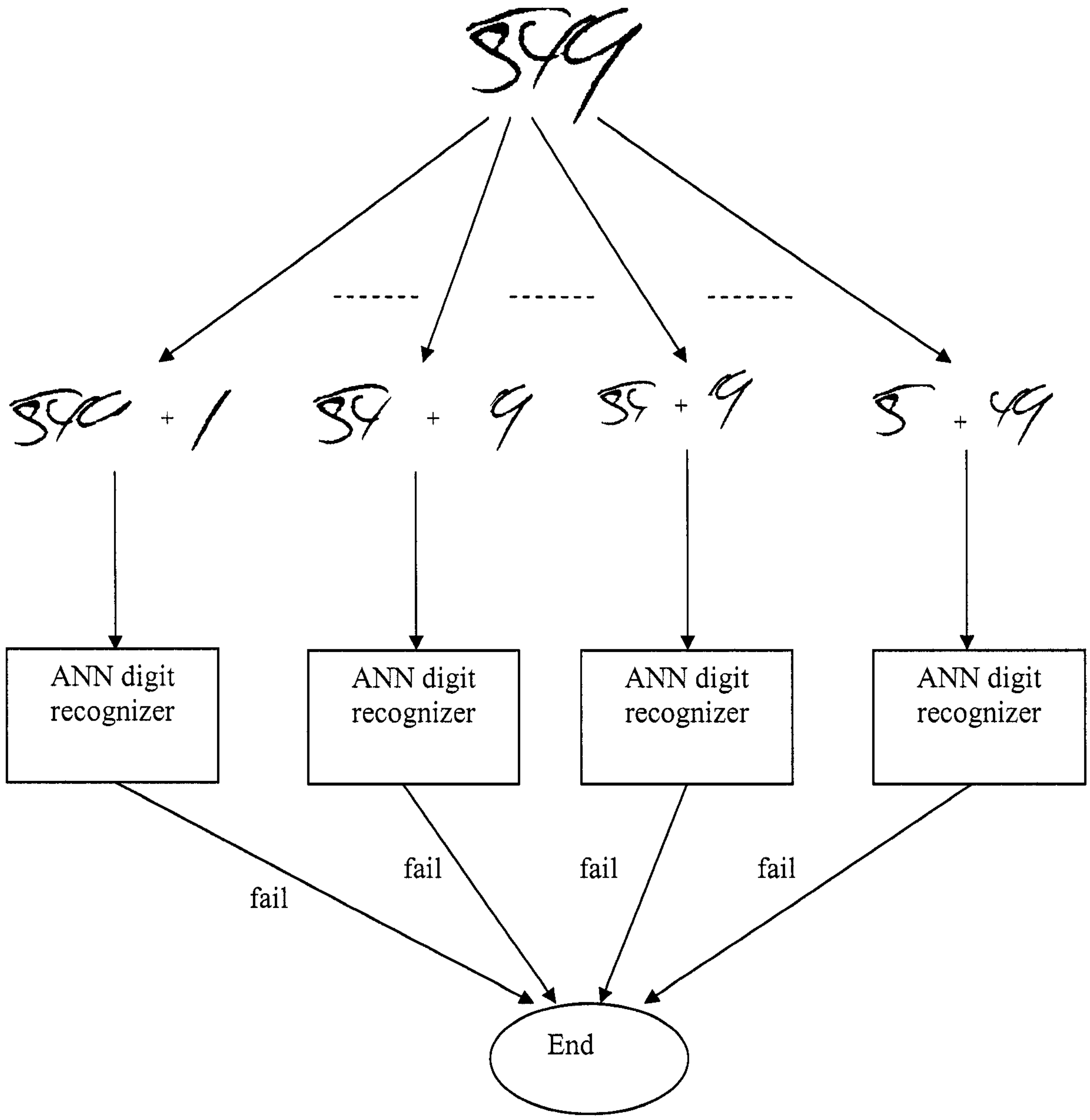



Figure 36. One sample of segmenting the numeral string by the original segmentation algorithm ("--------" indicates other possible segmentation candidates).

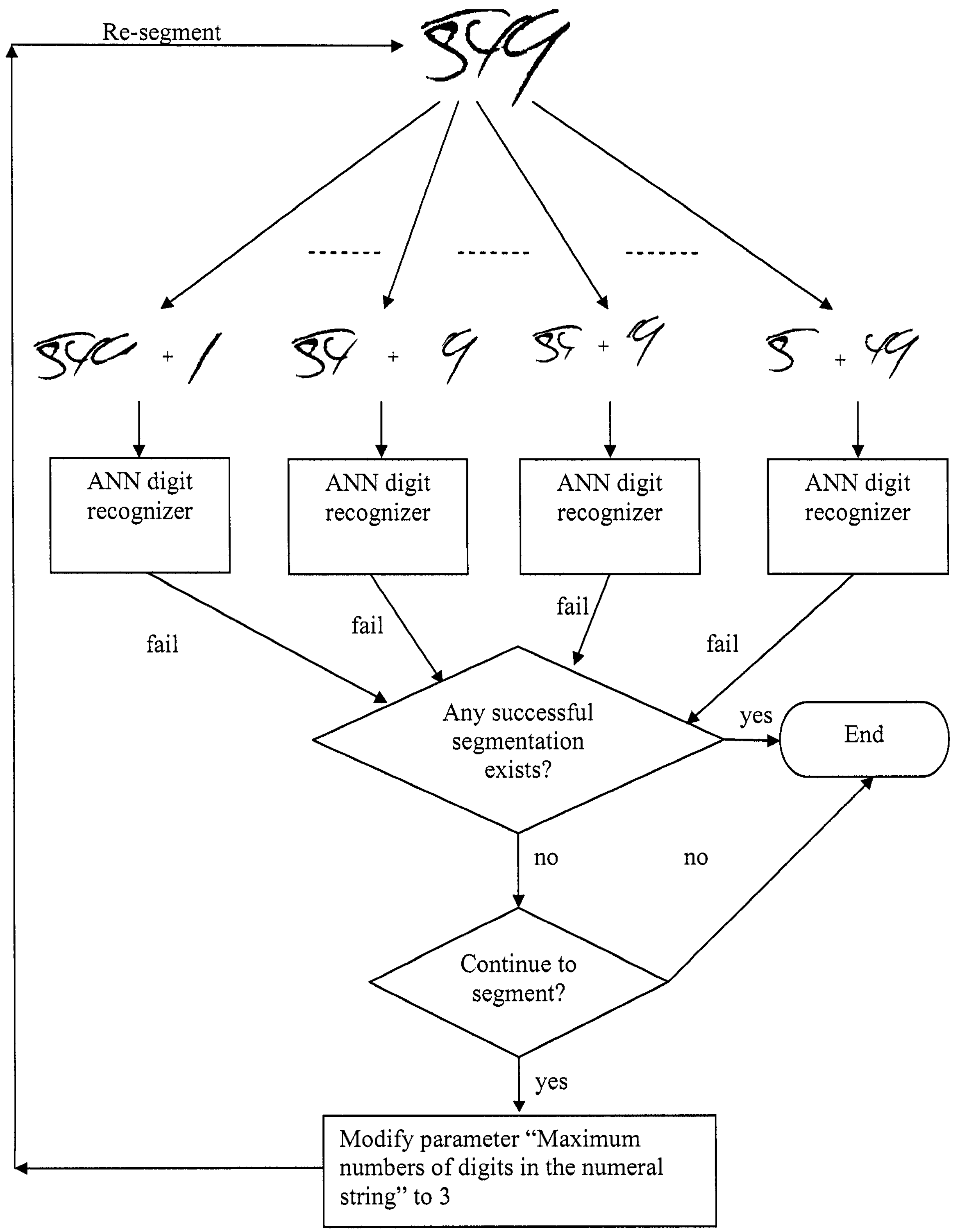



Figure 37. (Continue on next page)

(Continue: Re-segment)

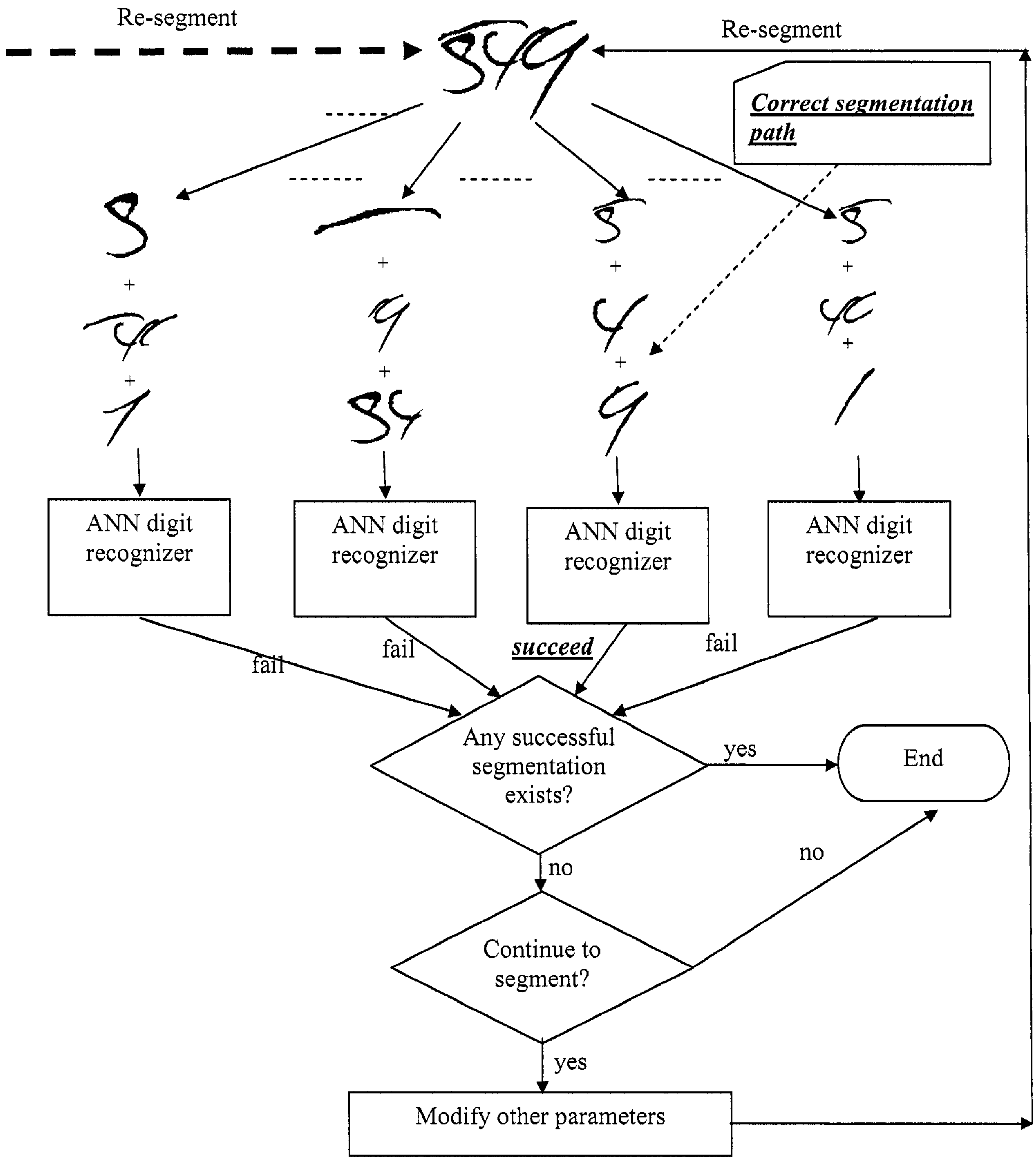
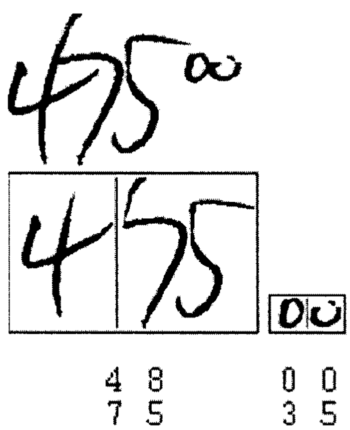
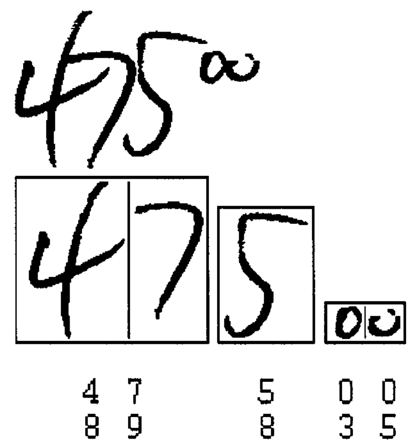



Figure 37. One sample of applying the new feedback-based segmentation algorithm ("--------" indicates other possible segmentation candidates).

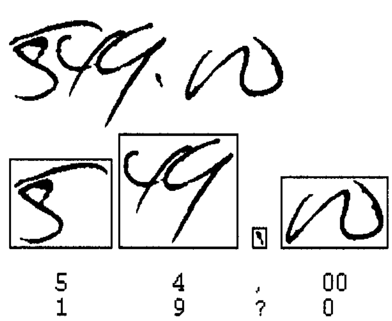
The new algorithm can deal with many difficult segmentation tasks. The experiment on a large real check database shows that the recognition rate of the new feedback-based algorithm improves considerably. The experimental results will be discussed in depth in Chapter 6. Some segmentation and recognition results of the original algorithm and new algorithm are shown in Figure 38:



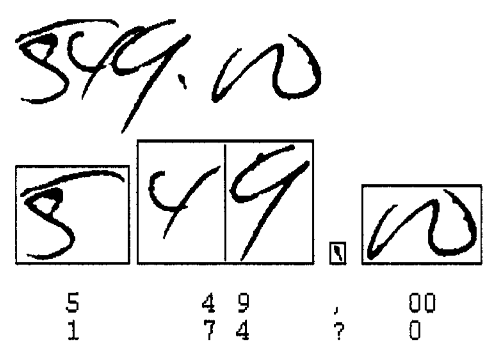

(a) Original algorithm



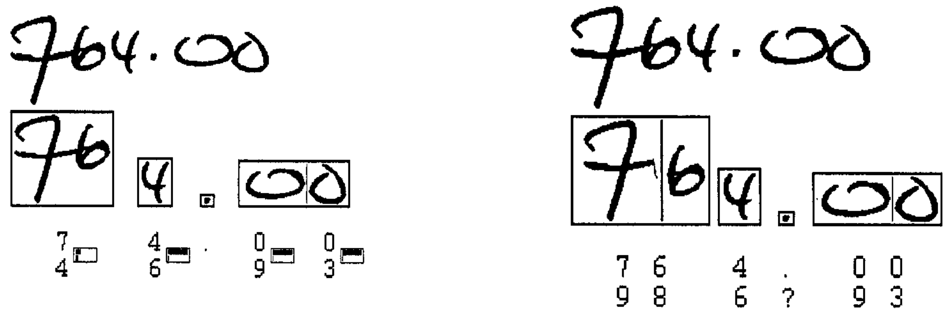

(b) New algorithm




(c) Original algorithm



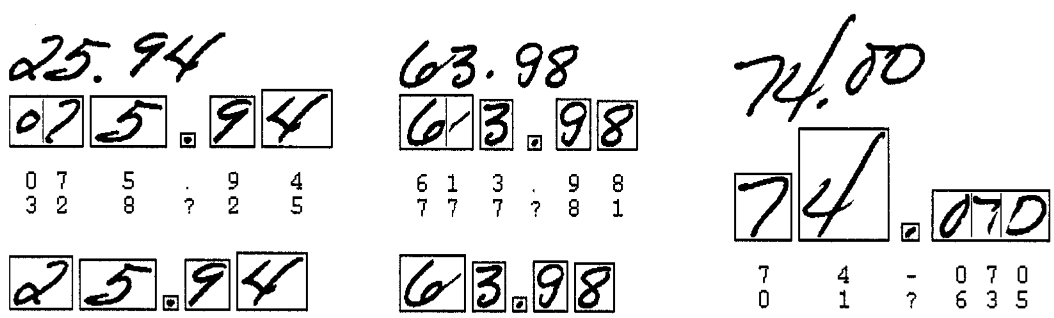

(d) New algorithm



(e) Original algorithm (f) New algorithm

Figure 38. Comparison of the original segmentation algorithm and new feedback-based segmentation algorithm.

The new segmentation algorithm sometimes will cause the segmentation errors when adjusting the segmentation parameters. Some segmentation errors are shown in Figure 39:



(a) Sample 1 (b) Sample 2 (c) Sample 3

Figure 39. Some segmentation errors of new segmentation algorithm

In Figure 39 (a) the first segmentation parameter is adjusted so small that "2" is segmented into "0" and "7"; in Figure 39 (b) the second segmentation parameter is adjusted so small that "6" is segmented into "6" and "1"; in Figure 39 (c) the third segmentation parameter is adjusted so big that "00" is segmented into "070".

## 3.4 Other Improvements in Segmentation

Other improvements in segmentation include the implementation of two rejection strategies in the new segmentation algorithm. The first strategy aims to reject unreasonable segmentations while the second strategy is used to avoid accepting a digit string as one digit. Both strategies are implemented by analyzing the positions of segmented sub-images. The problem of how to deal with a broken part of digit "5" during segmentation is also discussed.

### 3.4.1 Rejection Strategies

In this section, the term "rejection" means refusing to accept a wrongly segmented digit or a digit string as one digit. The ability to accept a correctly segmented digit and the ability to reject a wrongly segmented digit or a digit string are equally important. Rejecting the image which is not a complete individual digit will give the segmentation algorithm correct feedback and make it continue to search for new segmentation paths; conversely, accepting this image will give the segmentation algorithm wrong feedback, will make it stop searching for new segmentation paths and will accept a wrong recognition result. However, the rejection performance of CENPARMI ANN numeral recognizer is not good. In the following two sub-sections, two rejection strategies which are independent of the recognizer will be discussed: the algorithm for rejecting unreasonable segmentations and the algorithm for rejecting digit strings.

#### 3.4.1.1 Algorithm for Rejecting Unreasonable Segmentations

In segmenting the connected components, although the recognition of segmentation results is sometimes successful, some results are not reasonable and should be rejected. For example, "8" could be segmented as "00", "9" could be segmented into "01", and "6" could be segmented into "10". These three examples are shown in Figure 40:

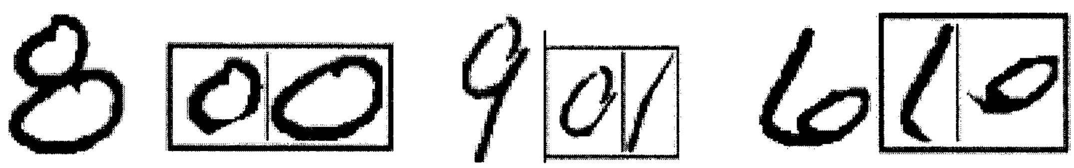


Figure 40. Wrong segmentation paths.

Obviously, these segmentation paths should be rejected, even though the segmentation results are accepted by CENPARMI digit recognizer. In order to reject this kind of segmentation path, the positions of two segmented sub-images should be examined to decide whether the segmentation path is reasonable before applying the recognizer to them. The algorithm for deciding whether the segmentation path is reasonable is described as follows:

**Algorithm**: RejectUnreasonableSegmentation( path )

get the coordinates of top-left corner and bottom-right corner of the left sub-image and right sub-image;

calculate the horizontal overlap ratio of the left sub-image and right sub-image: *horizontal_ratio*; ( horizontal overlap ratio means the ratio value of the width overlap of two sub-images and minimum width of two sub-images ) calculate the vertical overlap ratio of the left sub-image and

right sub-image: *vertical_ratio*; ( vertical overlap ratio means the ratio value of the height overlap of two sub-images and minimum height of two sub-images )

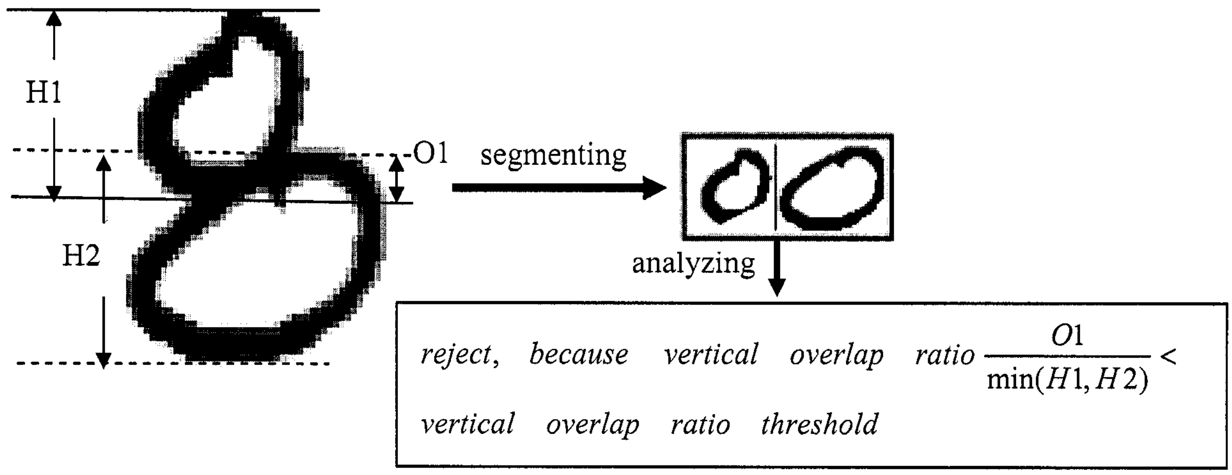
**if** ( *horizontal_ratio > pre-defined threshold 1*

|| *vertical_ratio < pre-defined threshold 2* )

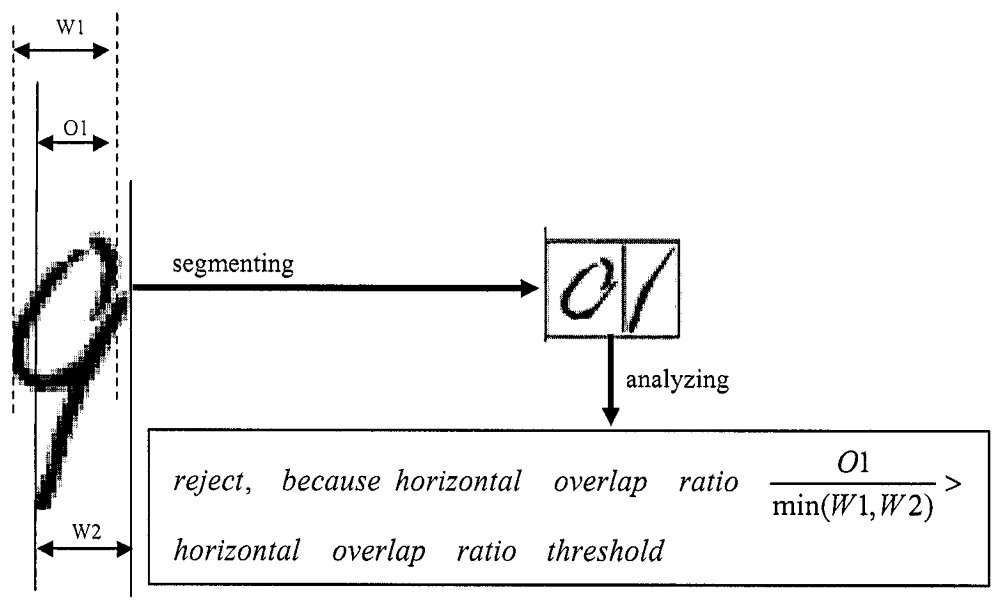
return 0;

**else** return 1;

Figure 41 shows some examples of the application of this algorithm. In Figure 41 (a), "8" could be segmented as two "0"s which are accepted by the numeral recognizer. However, the two "0"s are overlapped vertically and the vertical overlap ratio is smaller than a pre-defined threshold so that the segmentation result is rejected; in Figure 41 (b), "9" could be segmented as "0" and "1" which are accepted by the numeral recognizer. However, the "0" and "1" are overlapped horizontally and the horizontal overlap ratio is bigger than a pre-defined threshold so that the segmentation result is rejected.




(a) Sample 1




(b) Sample 2

Figure 41. Two examples of applying the algorithm for rejecting unreasonable segmentations.

#### *3.4.1.2 Algorithm for Rejecting Digit strings*

Although the recognition rate of CENPARMI individual digit recognizer can be up to 99.07%, the rejection performance of the recognizer is not good, especially for images of digit strings, which are generally generated by grouping two or more connected components together in the segmentation algorithm. For example, in CENPARMI individual digit recognizer, the digit pair of "14" is recognized as "1"; the digit pair of "49" is recognized as "4", and the digit string of "788" is recognized as "7". The image of these digit strings are shown in Figure 42:

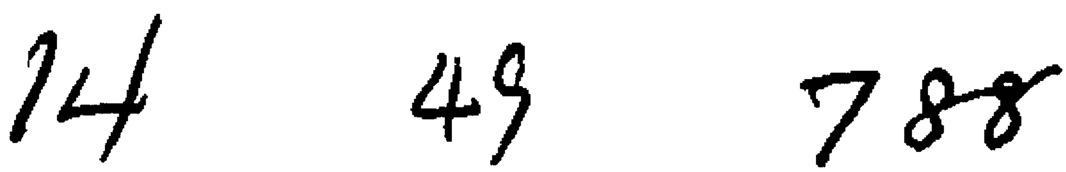


Figure 42. Some samples of digit strings.

These kinds of recognition errors will influence the segmentation performance because the segmentation algorithm is based on whether the segmentation path is accepted or not by the recognizer. So besides using the recognizer to determine whether the segmentation path is accepted or not, we also use one kind of structure analysis algorithm to help the recognizer to reject a digit string, that is, to avoid recognizing it as one digit. The structure analysis algorithm is described as follows:

**Algorithm**: IsMorethanOneDigit( image )

get the connected components in the image;

**if** ( only one connected component in the image ) return 0;

**for** ( any two connected components in the image )

calculate the height ratio of the two connected components: *height_ratio*;

**if** ( *pre-defined threshold 3* > *height_ratio* > *pre-defined threshold 4* ) calculate the vertical overlap ratio of the two connected

components: *vertical_ratio*;

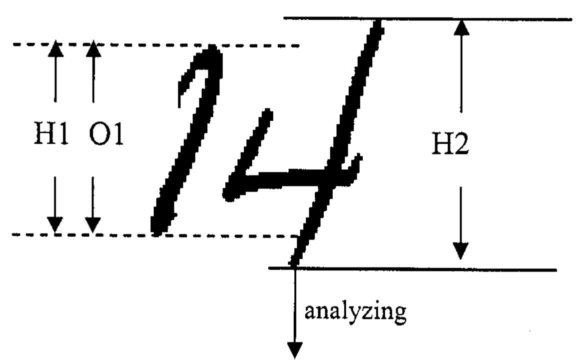
**if** ( *vertical_ratio* > *pre-defined threshold 4* )

**return** 1;

**end for**

**return** 0;

In Figure 43, one example is shown to illustrate the application of the algorithm:




Reject, because $\frac{O1}{\min(H1,H2)}$ > *vertical ratio threshold* and

*Max height ratio threshold* > $\frac{H1}{H2}$ > *Min height ratio threshold*

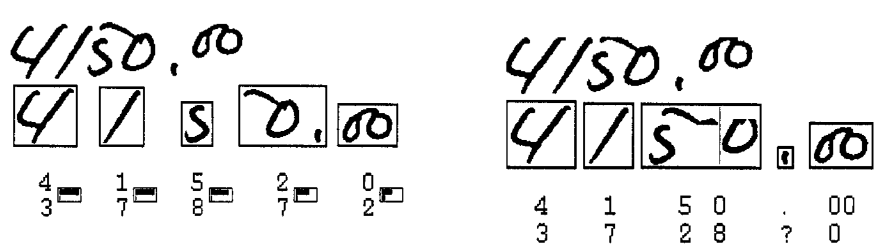
Figure 43. One example of applying the algorithm for rejecting a digit string.

### 3.4.2 The Problem about Non-connected Digit '5'

Sometimes the digit "5" is written with the horizontal stroke "-" detached. In this case, the number "5" could be recognized as two parts: the horizontal stroke "-" and the rest of "5". The incomplete digit "5" without the horizontal stroke could be recognized as "5" with a very high confidence score by CENPARMI digit recognizer. Some examples of this are shown in Figure 44:

Figure 44. Some examples of digit pairs containing digit "5".

In these cases, even though the left connected components of the images are recognized as "5", it is still necessary to group the left connected component with the right connected component and then segment the group. The original algorithm [1] was modified to address this issue: when the recognition result of one connected component is "5", grouping it with its right connected component in its neighborhood and segmenting the group will be performed. The modified algorithm deals very well with this kind of issue. Some results of applying the original algorithm and modified algorithm are shown in Figure 45:



(a) original algorithm (b) modified algorithm

Figure 45. The comparison of the original algorithm and modified algorithm.

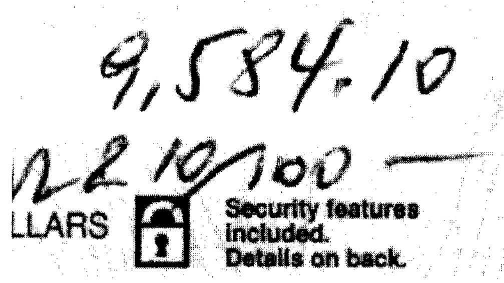
# Chapter 4

# Pre-processing and Post-processing

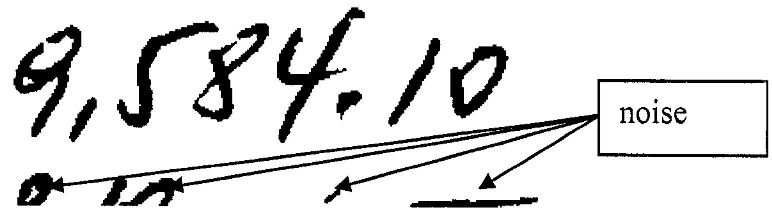
Pre-processing and post-processing modules are both very important modules for the CRS. Some functions of these modules will be discussed in this chapter, such as border noise removal in the pre-processing module, and the detection of the implicit decimal point in the post-processing module.

## 4.1 Pre-processing

In the pre-processing module, there is a procedure called border noise removal. Its functionality is to remove some useless strokes, which don't belong to the courtesy amount but have been extracted from the background or other areas of the check. These kinds of strokes cannot be removed by the common pepper noise removal algorithm because they are much larger in size than pepper noise. If they are not removed before the image is sent to the segmentation and recognition modules, they could be recognized as digits. One sample with this kind of noise is shown in Figure 46(b).

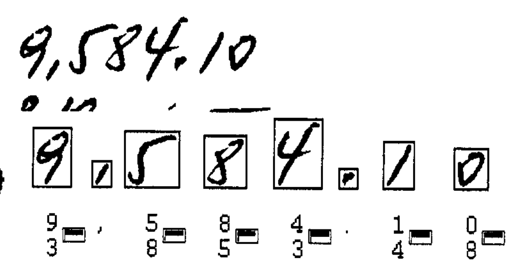



(a) Original image from the check




(b) Courtesy amount extracted with noise


(c) Courtesy amount after border noise removal




(d) Recognition results after border noise removal

Figure 46. Original border noise removal algorithm.

The original border noise removal algorithm of CENPARMI CRS views a stroke touching the border as noise when its absolute length is smaller than a threshold. The procedure of the original algorithm is described as follows:

**Algorithm**: OriginalCleanBorderNoise( image )

finding the strokes which touch the top or bottom border;

**for** ( each stroke found in the border )

**if** ( it protrudes less than a threshold distance into the image

&& its shape is narrow and long )

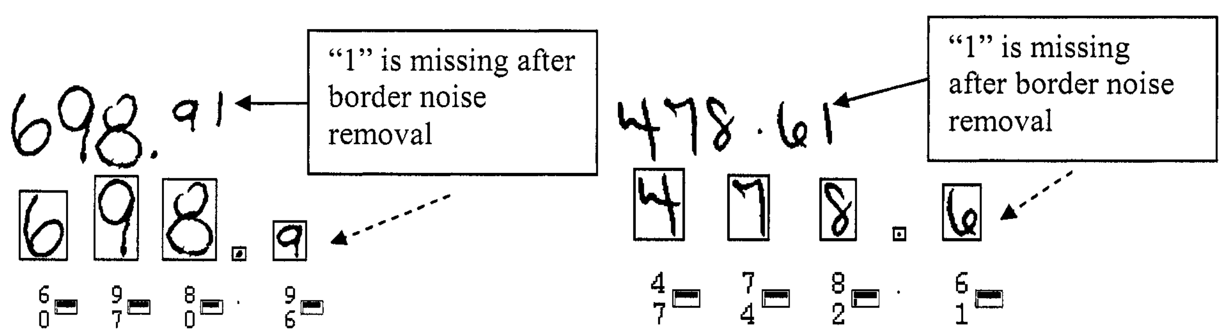
remove it from the image;

**end for**

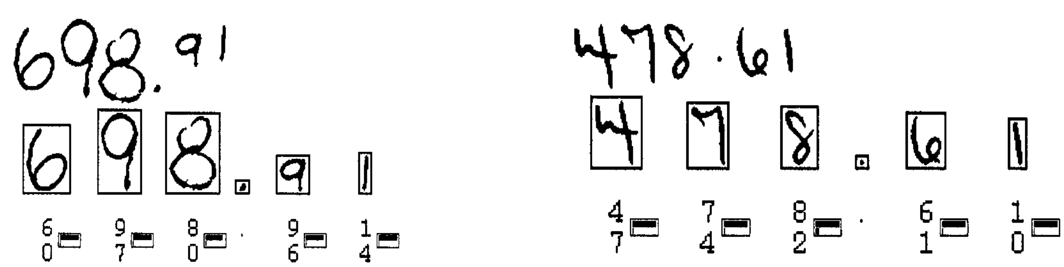
After applying the original border noise removal algorithm to the sample in Figure 46 (b), the result is shown in Figure 46(c).

There is a drawback in the original border noise removal procedure: it only considers the absolute length of a stroke touching the border and sometimes will erroneously remove some part of the cent amount when it is relatively small and touches the top/right border. Some examples of this are shown in Figure 47:

(a) Courtesy amount extracted




(b) Recognition results using original border noise removal algorithm



(c) Recognition results using new border noise removal algorithm

Figure 47. Recognition results using the original and new border noise removal algorithms.

Therefore, a new algorithm is proposed which views a stroke touching the border as noise when its relative length is smaller than a threshold, compared with the length of another stroke next to it, which doesn't touch the border. The procedure of the new algorithm is described as follows:

**Algorithm**: NewCleanBorderNoise( image )

finding the strokes which touch the border;

**for** ( each stroke (*stroke1*) found in the border )

**if** ( *stroke1* protrudes less than a threshold distance into the image

&& *stroke1's* shape is narrow and long )

find the stroke (*stroke2*) next to it which doesn't touch the border;

**if** (length of stroke1 and stroke2 are not close)

remove stroke1 from the image;

**end for**

After applying the new border noise removal algorithm to the samples shown in Figure 47(a) and recognizing them, results are shown in Figure 47(c). The new algorithm's flowchart is shown in Figure 48:

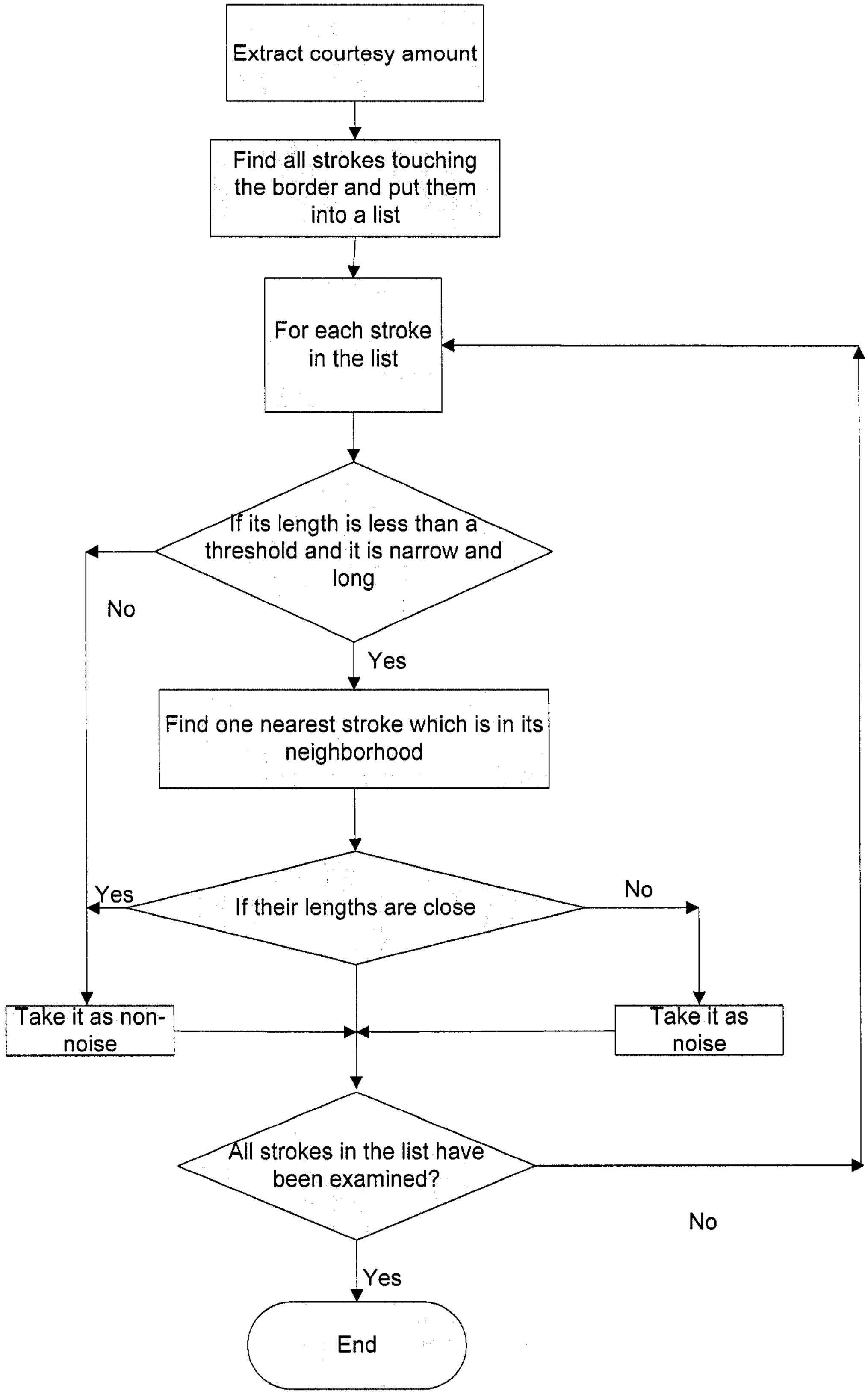



Figure 48. Flowchart of new border noise removal algorithm.

## 4.2 Post-processing

After finishing courtesy amount recognition, the raw recognition results from courtesy amount recognition will be further processed by the post-processing module. Post-processing module is very important for the CRS because the raw recognition results may include some extraneous characters. It is necessary to identify and remove redundant characters or errors in the raw recognition results. Two kinds of post-processing procedures will be discussed in this section.

### 4.2.1 Removal of Horizontal Stroke '-', Decimal Point '.', and Comma ','

Sometimes people write some horizontal strokes '-' at the two sides of a courtesy amount for the delimitation of the courtesy amount. The decimal point '.' or comma ',' could appear in the positions of a decimal point and thousands separators of a courtesy amount. Moreover, some noises could be recognized as horizontal stroke '-' or point '.'. Obviously, those redundant horizontal stroke '-', point '.', and comma ',' noises in the wrong positions should be ignored. Some samples are shown in Figure 49:

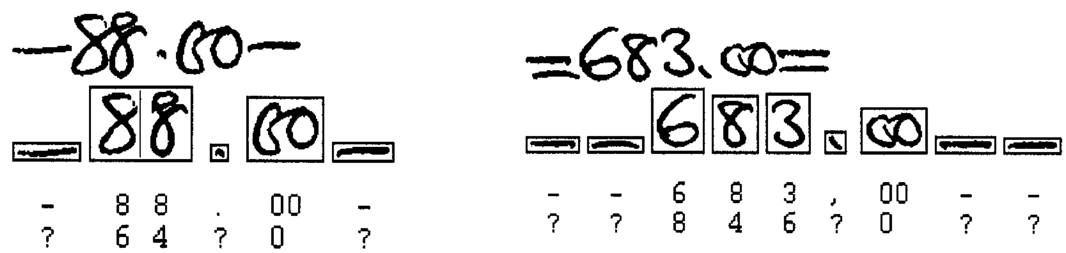



Figure 49. Some samples of courtesy amounts with horizontal strokes '-', points '.', or commas ','.

The procedure for removing the horizontal stroke '-', point '.', and comma ',' is described as follows:

a. Remove leftmost and rightmost nondigit characters until a digit is encountered.

b. Set the characters of horizontal stroke '-', point '.', and comma '.' at the position of the decimal point (that is, the last third position) to a special character 'P'

c. Remove all other characters of horizontal stroke '-', point '.', and comma '.'

d. Restore the special character 'P' to a decimal point

### 4.2.2 Detection of Implicit Decimal Point

Sometimes people write the cent part of a courtesy amount without prefixing it with a decimal point, but its cent part is relatively smaller or in a higher place than its dollar part. In original CENPARMI CRS system, this kind of writing style is not considered and no corresponding post-processing procedure is used to deal with it. So the implicit decimal point could be ignored and the courtesy amount could become 100 times bigger than its original amount. For example, 7758.76 could be recognized as 775876. Some samples in this kind of writing style are shown in Figure 50:

Figure 50. Some samples with an implicit decimal point.

One procedure is proposed to detect implicit decimal points in the post-processing. The procedure is described as follows:

a. Consider last two digits as a potential cents part, calculate their average height (*centHeight*) and baseline position (*centXbottom*).

b. Consider other digits in the courtesy amount as a potential dollar part, calculate their average height (*dollarHeight*) and baseline position (*dollarXBottom*).

c. If *centHeight / dollarHeight* < *threshold1* or *(dollarXBottom – centXbottom)/ dollarHeight* > *threshold2*, then there could be an implicit decimal point between the cent part and the dollar part.

d. If no decimal point exists in the recognition result and there could be an implicit decimal point derived from step c, then we can assume that there is an implicit decimal point between the cent part and the dollar part.

One example is given to illustrate the application of this algorithm in Figure 51:

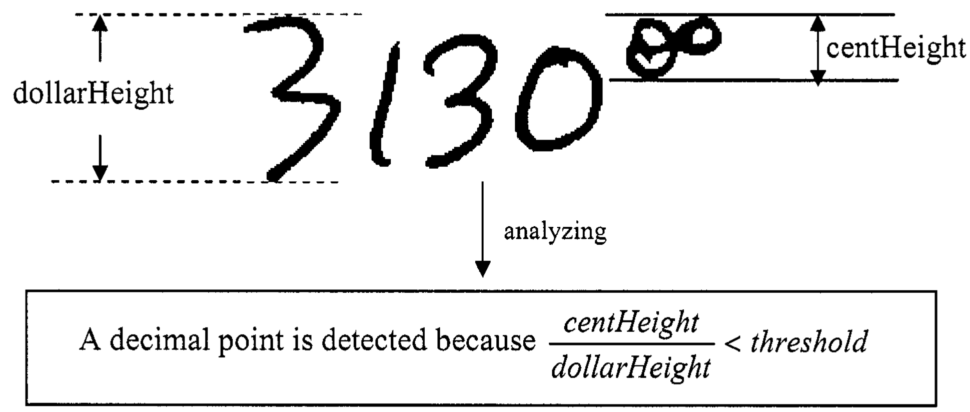



Figure 51. One example of applying the algorithm to detect the implicit decimal point.

In some case, the procedure cannot detect the decimal point in the digit string. One sample is shown in Figure 52:

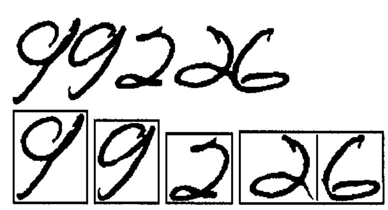


Figure 52. One Sample of detecting decimal points error

In Figure 52, the procedure fails to detect the decimal point because the average height and position of the last two digits in the string are similar to those of the other digits.

# Chapter 5

# Convolutional Neural Networks Recognizer for "00" and "000"

The common digit recognizer is used only for the recognition of individual digits from '0' to 9, not for pairs of digits. Any digit could be recognized only if it could be segmented correctly from the connected numeral string, because the courtesy amount recognition in CENPARMI CRS is a segmentation-based recognition algorithm. The drawback of this method is that it depends too much on the performance of the segmentation algorithm. Since segmentation of a connected numeral string is a very complex problem, error is inevitable in some difficult cases. So, if a recognizer for some combinations of multiple digits which occur frequently in courtesy amounts and are difficult to segment, is integrated into CENPARMI CRS system, the performance of the CRS could be improved. Double zero pairs "00" and triple "000" strings are examples of such difficult cases.

While "00" and "000" numeral strings may appear very simple for recognition. Their variability in writing is enormous and some extraneous strokes are often inserted between the "0"s. It is very difficult to segment "00" or "000" numeral strings directly, remove useless strokes, and recognize individual '0's. Some samples of "00" and "000" numeral strings in different writing styles are shown in Figure 53:

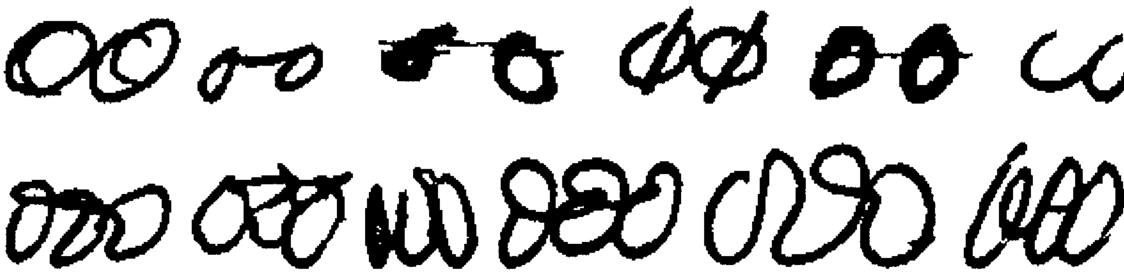


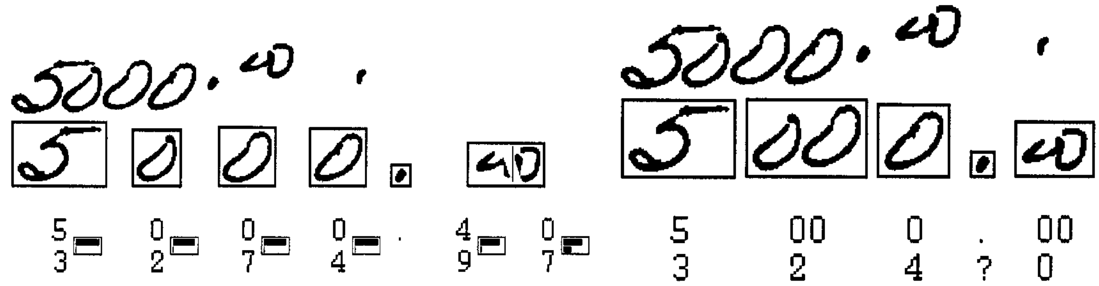
Figure 53. Some samples of "00" and "000" numeral strings.

## 5.1 Integration of Original "00" Pair Recognizer

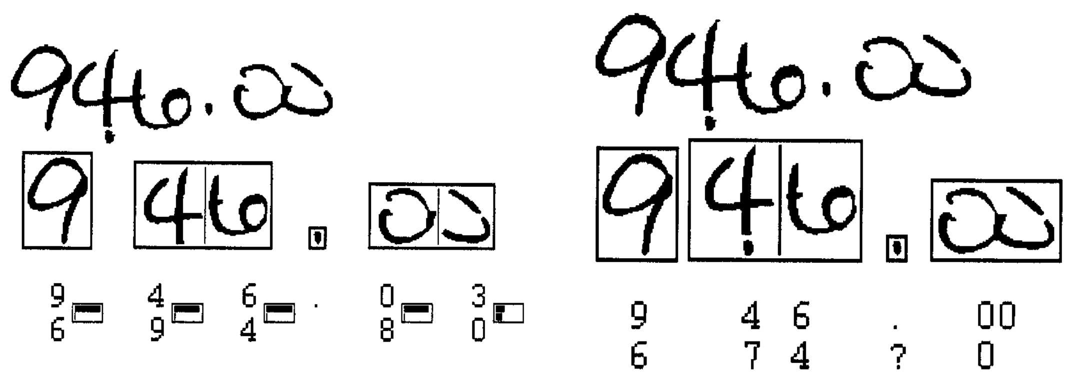
A neural network recognizer for the "00" pair was developed in the past for courtesy amount recognition at CENPARMI [40]. It has been integrated into new CENPARMI CRS. This original "00" pair recognizer was trained with only 500 samples of "00" pairs and 50,000 isolated digits [40]. Because the number of "00" pairs was too small, the performance of the "00" pair recognizer was comparatively weaker than that of the individual digit recognizer.

Fortunately, there is a simple method to improve the ability of new CENPARMI CRS to recognize "00": decrease the confidence score threshold for accepting a digit string as "00" while using the original "00" pair recognizer. Based on experiments, an optimal threshold is determined. In this way, the performance of new CENPARMI CRS can be improved.

Some examples before and after integration of the original "00" neural network recognizer are shown in Figure 54:




(a) Without the "00" pair recognizer (b) With the "00" pair recognizer




(a) Without the "00" pair recognizer (b) With the "00" pair recognizer

Figure 54. Recognition comparison between with and without the original "00" pair recognizer.

## 5.2 New CNN Recognizer for "00" and "000"

In order to improve the performance of the CRS for the recognition of "00" and "000", a new Convolutional Neural Network(CNN) recognizer for "00" and "000" is used. CNN is one kind of feed-forward network which can automatically extract topological properties from an input image and can have a relatively small amount of weights for training, while a generalized feed-forward neural network will ignore topological properties [49][50] and have a very large amount of weights for training.

CNN is also more suitable for visual document tasks and can achieve an especially good performance for a limited amount of training data of "00" and "000" numeral strings.

### 5.2.1 The Structure of the New CNN Recognizer

The structure of the new CNN [51] is shown in Figure 55:

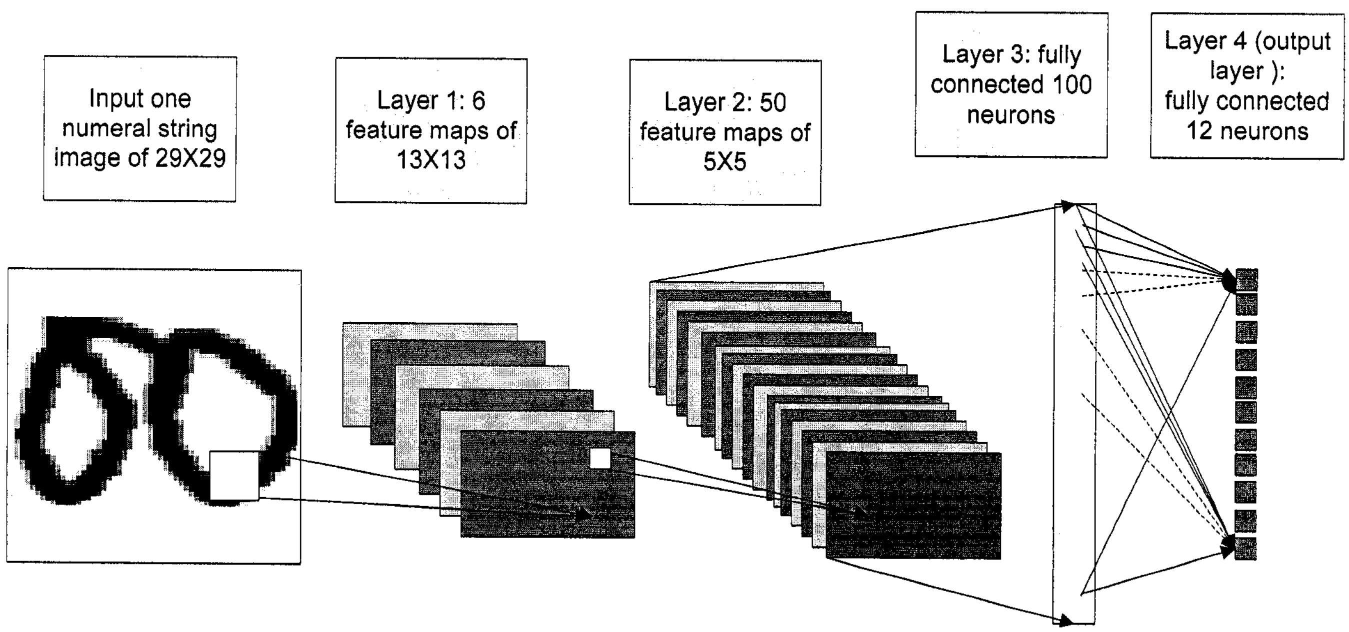



Figure 55. Structure of CNN recognizer for "00" and "000".

The input of the CNN is the grayscale image of one handwritten numeral string of size $29 \times 29$. The CNN consists of 4 layers:

Layer 1 is a convolutional layer which is composed of six feature maps of $13 \times 13$. Each pixel of one feature map is connected to a $5 \times 5$ sub-image of the inputted image with $5 \times 5 = 25$ weights and 1 bias, which are used for convolutional computation. Therefore, there are $(5 \times 5+1) \times 6 = 156$ weights.

Layer 2 is also a convolutional layer which is composed of 50 feature maps of $5 \times 5$. Each pixel of one feature map is connected to the $5 \times 5$ sub-maps of 6 feature maps of layer 1 with $5 \times 5 \times 6 = 150$ weights and $1 \times 6 = 6$ bias, which are used for convolutional computation. Therefore, there are $(150+6) \times 50 = 7800$ weights.

Layer 3 is a traditional fully-connected hidden layer with 100 neurons. It has 125100 weights.

Layer 4 is a traditional fully-connected output layer with 12 neurons, which corresponds to individual digits from '0' to '9', "00", and "000", respectively. It has 1212 weights.

### 5.2.2 How to Collect Samples for "00" and "000"

Samples of "00" and "000" were collected from the Quebec Bell check database. There are four sets in the database: English training set, English test set, French training set, and French test set. The number of checks in each set is shown in Table 1.

Table 1. The number of checks in Quebec Bell check data set

| | English training set | English test set | French training set | French test set |
|---|---|---|---|---|
| Number of checks | 2813 | 2787 | 3993 | 3392 |

It is very time-consuming and infeasible to only manually extract "00" and "000" from the Quebec Bell check database. Therefore, we extracted the samples "00" and "000" in two steps semi-manually, as follows: first, use the original "00" recognizer to extract the coarse results without setting any rejection threshold; then, manually pick up

"00" and "000", respectively, from the coarse results, to obtain the final results. This method is very efficient. The number of samples in the coarse results and final results are shown in Table 2:

Table 2. The number of samples in the coarse results and final results

| | English training set | English test set | French training set | French test set |
|---|---|---|---|---|
| Number of checks in coarse results | 1728 | 1820 | 2387 | 2037 |
| Number of checks in "00"final results | 787 | 827 | 1111 | 942 |
| Number of checks in "000"final results | 59 | 74 | 46 | 36 |

Because the samples from the English test set should be used only for testing, all samples from the other three data sets could be used for training. Moreover, 267 samples of "00" from one unknown database were also used for training. So, we had 787+1111+942 +267= 3107 samples of "00" and 59+46+36=141 samples of "000" for training. These samples were used together to train the new CNN with the MNIST handwritten digit database. Compared with the MNIST handwritten digit database, which consists of 60,000 samples of individual digits from 0 to 9, the total number of the samples of "00" and "000" for training were too small. Because there are about 6,000 images for each digit in the MNIST handwritten digit database, it was reasonable to expand the number of the samples of "00" and "000" for training to about 6000.

There are various ways to expand the sample set, such as using affine transformation, bilinear interpolation, and elastic deformation. Here, elastic deformation [50] was used to expand the "00" and "000" sample set.

Elastic deformation generates new images by deforming the original image with the random displacement fields in x and y coordinates, respectively, as follows: (1) initialize the random displacement fields $\Delta x$ and $\Delta y$ with a uniform distribution between 0 and 1; (2) they are convolved with a Gaussian of standard deviation $\sigma$; (3) normalize them and then multiply them by a scaling factor $\alpha$, which controls the intensity of the deformation, to obtain the final displacement fields. Here, deviation $\sigma$ is an elasticity coefficient. An intermediate value $\sigma$ is set to achieve good elastic deformation effects.

By the elastic deformation algorithm, the "00" sample set is doubled in size, while the "000" sample set has its size multiplied by 41. Thus, after elastic transformation, the size of the "00" sample set becomes $2 \times 3107 = 6214$, while the size of the "000" sample set becomes $41 \times 141 = 5781$.

### 5.2.3 Experimental Results for CNN "00" and "000"Recognizers

First, the original "00" recognizer and CNN recognizer were tested on 827 "00" samples collected from the English test set of the Quebec Bell check database. The experimental results are shown in Table 3.

From Table 3, we can see that the recognition rate has improved considerably when all test data are "00" samples.

Table 3. The recognition rates of the original "00"recognizer and the CNN recognizer

| | Correct | Error | Recognition rate |
|---|---|---|---|
| Original "00" recognizer | 622 | 205 | $622/827 \approx 0.752$ |
| New CNN "00" recognizer | 812 | 15 | $812/827 \approx 0.982$ |

Experimental results about rec(n) of the original "00" recognizer and the CNN recognizer for "00" are shown in Table 4. Here, rec(n) means the percent of cases for which the top n recognition results include the correct courtesy amount.

Table 4. The rec(n) of the CNN recognizer for "00"

| Recognizer | Top1 | Top2 | Top3 | Top4 | Top5 |
|---|---|---|---|---|---|
| Original "00" recognizer | 75.2% | 75.5% | 75.5% | 75.5% | 75.5% |
| New CNN "00" recognizer | 98.2% | 98.3% | 98.3% | 98.3% | 98.3% |

From Table 4, we can see that the rec(n)s of the CNN recognizer for "00" are much higher than those of the original "00" recognizer.

Moreover, in order to compare these two recognizers' reliability rates, the CNN "00" recognizer and the original "00" recognizer have been respectively integrated into the courtesy amount recognition module of CENPARMI CRS and tested on the English test set of the Quebec Bell check database. In the module, these two recognizers have been used to recognize the connected components from courtesy amounts. If they could recognize one connected component as "00", then no segmentation would be needed on the connected component. The experimental results are shown in Table 5:

Table 5. Reliability rates of the original "00" recognizer and CNN recognizer in CRS

| | Connected components which are recognized as "00" | Correct | Error | Reliability rate |
|---|---|---|---|---|
| Original "00" recognizer | 1558 | 389 | 1169 | $389/1558 \approx 0.25$ |

| | | | | |
|---|---|---|---|---|
| CNN "00" recognizer | 879 | 780 | 99 | 780/879≈0.89 |

As for the performance of the CNN recognizer for "000", similar experiments were conducted. Based on 74 "000" samples from the English test set, the experimental results regarding the recognition rate of the CNN recognizer are shown in Table 6:

Table 6. The recognition rates of the CNN recognizer for "000"

| Total | Correct | Error | Recognition rate |
|---|---|---|---|
| 74 | 67 | 7 | 67/74≈0.905 |

Compared with the recognition rate of the CNN recognizer for "00", the recognition rate of the CNN recognizer for "000" is relatively low. That is because the amount of original "000" data, 141 samples, is too small and there are too many artificially generated "000" data by the elastic deformation algorithm.

Experimental results about rec(n) of the CNN recognizer for "000" are shown in Table 7:

Table 7. The rec(n) of the CNN recognizer for "000"

| n | Top1 | Top2 | Top3 | Top4 | Top5 |
|---|---|---|---|---|---|
| Rec(n) | 90.5% | 100% | 100% | 100% | 100% |

In Table 7, we can see that rec(n) can reach 100% when n >= 2.

In order to test the CNN recognizer's reliability rate for "000", it was also integrated into the courtesy amount recognition module of CENPARMI CRS to recognize the connected components from courtesy amounts. Based on the English test set, the reliability rates are shown in Table 8:

Table 8. Reliability rate of the CNN recognizer for "000" in CRS

| Connected components which are recognized as "00" | Correct | Error | Reliability rate |
|---|---|---|---|
| 86 | 60 | 26 | $60/86 \approx 0.70$ |

Based on the English test set, the courtesy amount recognition rates of CENPARMI CRS, with the original "00" recognizer or with the CNN recognizer for "00" and "000" are shown in Table 9:

Table 9. Performance comparison of the CRS with two different recognizers

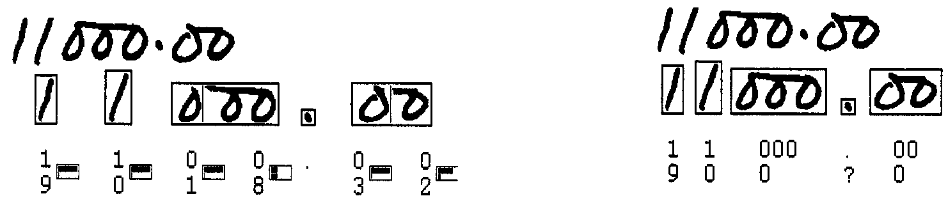
| CRS system | Correct | Error | Courtesy amount recognition rate |
|---|---|---|---|
| CRS with the original "00" recognizer | 1983 | 804 | $1983/2787 \approx 0.712$ |
| CRS with the CNN recognizer for "00" and "000" | 2075 | 712 | $2075/2787 \approx 0.745$ |

From Table 9, we can see that by the integration of the CNN recognizer to handle the recognition of "00" and "000" strings, the performance of the CRS has improved from 71.2% to 74.5%.

Some courtesy amount recognition results of CENPARMI CRS with the original "00" recognizer and with the CNN recognizer for "00" and "000" are shown in Figure 56:


(a) Original "00" recognizer

(b) CNN recognizer



(c) Original "00" recognizer (d) CNN recognizer

Figure 56. Courtesy amount recognition results of CENPARMI CRS with the original "00" recognizer and with the CNN recognizer.

From the above experiments, we can conclude that the recognition rate of the CNN recognizer for "00" has improved from 75.2% to 98.2%. The recognition rate of the CNN recognizer for "000" reaches 90.5%, which is also acceptable and could improve considerably if more "000" data is collected. By using the CNN recognizer for "00" and "000", the courtesy amount recognition rate of CENPARMI CRS has also improved from 71.2% to 74.5%.

# Chapter 6

# Experimental Results

In this chapter, experimental results regarding courtesy amount recognition of CENPARMI CRS will be first discussed. Then, the courtesy amount recognition errors will be classified and analyzed in detail.

## 6.1 Database

The courtesy amount recognition module of new CENPARMI CRS is tested on the Quebec Bell check database. The checks in the database are real checks which have been used in Montreal. They are from different banks, designed in different colors, fonts, layout formats, and backgrounds. Moreover, the checks have been written by different people in a real-life environment and scanned into the computer at 300 DPI in grey level. So these real checks are appropriate for testing the robustness and performance of new CENPARMI CRS.

One sample from the database is shown in Figure 57:

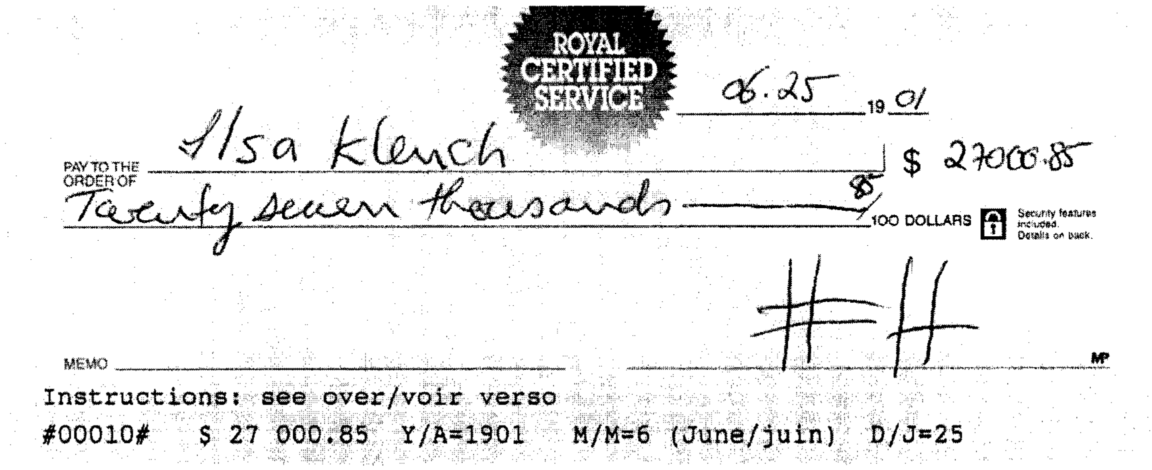



Figure 57. A sample from the Quebec Bell check database.

There are four different sets in the database: English training set, English test set, French training set, and French test set. The number of checks in each set is as follows: 2813, 2787, 3993, and 3392, respectively.

The Quebec Bell check database was chosen for the following reasons:

1. The number of checks used in the Quebec Bell check database is much larger than databases used by other researchers at CENPARMI ( in [40], Zhang's test database has 645 check images; in [6], Zhou's test database has 680 images).
2. The checks in the Quebec Bell check database are real checks from different banks.
3. The checks in the Quebec Bell check database were written by many different people with different writing styles and habits, which could reflect the complexity of recognizing checks with a check reading system in a real-life commercial environment.

Therefore, the results obtained from the experiments based on the Quebec Bell check database can indicate the real performance of CENPARMI check reading system more precisely and can inform us about the challenging problems that remain.

## 6.2 Working Environment

The check reading system has been implemented on Windows XP, developed in Visual C++ 2005, and run on a PC which has 3.5G memory and an Intel Pentium D 3.4GHZ CPU.

## 6.3 Recognition Results of the Database

The courtesy amount recognition module of CENPARMI CRS has been tested on the 4 data sets of the Quebec Bell Check database, respectively. Here, rejection is not considered, that is, recognition rate + substitution rate = 100%. The recognition rates for each data set are shown in Table 10:

Table 10. Courtesy amount recognition rates of original CRS and new CRS

| Test Sets | Total image number | Original CRS | New CRS |
|---|---|---|---|
| English training set | 2813 | 42.1% | 73.2% |
| English test set | 2787 | 40.6% | 74.5% |
| French training set | 3993 | 41.3% | 74.7% |

| | | | |
|---|---|---|---|
| French test set | 3392 | 40.9% | 74.7% |
| All | 12985 | 41.2% | 74.3% |

From Table 10, we can see that the overall courtesy amount recognition rate of the new CRS has improved from 41.2% to 74.3%. Its substitution rate has decreased from 58.8% to 25.9%. In [40], the courtesy amount recognition rate of Zhang's algorithm can reach up to 69.8%. However, this experimental result is based only on 400 checks, which may not sufficient to reflect the complexity of courtesy amount recognition.

Because there is no standard check database, it is difficult to compare CENPARMI CRS directly with CRSs developed by other researchers or companies. But their results can still be used to indicate the performance of CENPARMI CRS to some extent. Other CRS's experimental results for courtesy amount recognition are shown in Table 11:

Table 11. Courtesy amount recognition rates of other Check Reading Systems

| | %correct | %error | Source/Date |
|---|---|---|---|
| E.Lethelier et al | 60% | raw | [42]/1995 |
| Mitek | 55 | 1 | [41]/1996 |
| ParaScript | 47% | 1 | [27]/1997 |
| Lucent | 44% | 1 | Advertisement/1998 |
| A2iA | 49% | 1 | [11]/2001 |
| Luiz S. Oliveira et al | 57.17% | 0.5 | [33]/2002 |
| Unisys-SoftCAR+ | 79% | 1 | [52]/2008 |
| Orbograph | 70%~80% | 0.4 | [53]/2008 |

From Table 11, we can see that the courtesy amount recognition rate of the new CRS is comparable to that of other CRSs, although it does not consider how to reduce the substitution rate. However, it can still produce preliminary recognition results which can be used for validation with the legal amounts to suppress the substitution rate.

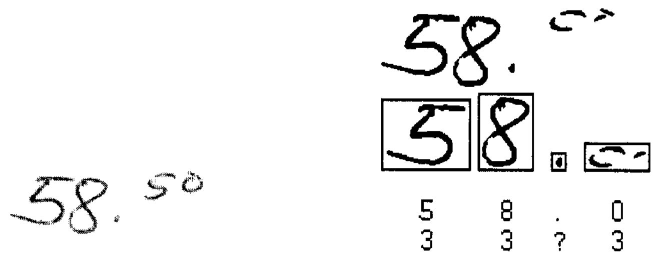
## 6.4 Error Analysis of Courtesy Amount Recognition

It is necessary to analyze the errors generated by new CENPARMI CRS because the information collected from error analysis could help us to understand the advantages and disadvantages of the new CRS and to improve it.

A careful examination of the courtesy amount recognition errors generated by the new CRS show that the errors can be classified into the following categories:

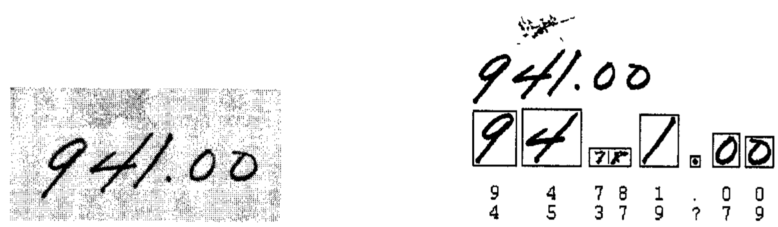
### 6.4.1 Item Extraction Errors

Due to the complexity of different backgrounds and layouts of different bank checks, it was possible to either remove some part of the courtesy amount or retain some background patterns during pre-processing and courtesy amount extraction procedures. Some examples of these are shown in Figure 58:

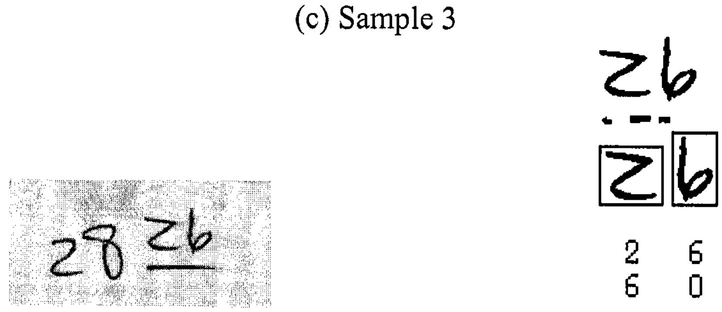



(a) Sample 1

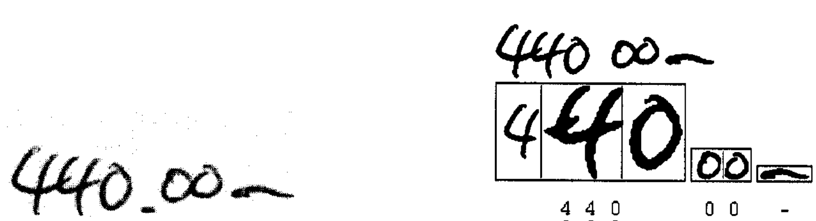

(b) Sample 2




(c) Sample 3




(d) Sample 4



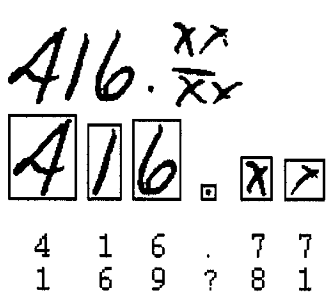

(e) Sample 5

Figure 58. Some examples of item extraction errors. Note that the left images are original courtesy amount at checks while the right images are the corresponding courtesy amounts extracted.

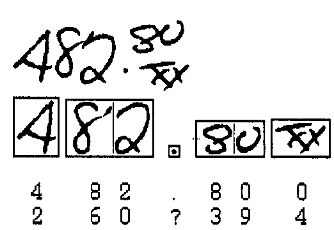
In the above Figure: in (a), some part of "50" is missing; in (b), the "$" sign is not removed; in (c), the background pattern is not removed; in (d), part of the numeral string is missing; and in (e), the decimal point is missing. Possible ways to reduce this kind of error could include: improving the item extraction algorithm, and validating courtesy amount recognition results with legal amount recognition results.

### 6.4.2 Non-digit Character Errors in Recognition

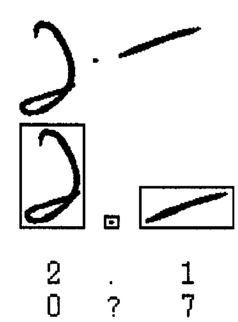
There are some non-digit characters in the courtesy amount, such as ",", ".", "/", "X". In CENPARMI CRS, the performance of recognizing "/" and "X" is not good. Some examples of misrecognizing "X" and "/" are shown in Figure 59. These "X"s are easily misrecognized as digits and "/"s can be misrecognized as "1".



(a) Sample 1



(b) Sample 2



(c) Sample 3

Figure 59. Some examples of recognition errors in "X" and "/".

### 6.4.3 Segmentation Errors

Although the courtesy amount recognition performance is improved considerably by the new feedback-based segmentation algorithm, there are still some segmentation errors because of the rejection performance of CENAPRMI individual digit recognizer and the complexity of the writing styles. Segmentation errors are classified into three classes: errors caused by rejection performance of CENPARMI individual digit recognizer, errors caused by overlooking decimal points, and errors caused by some writing styles. They are discussed in detail as follows:

#### 6.4.3.1 Errors Caused by Rejection Performance of CENPARMI Individual Digit Recognizer

The new feedback-based segmentation algorithm is based on recognition. It starts segmenting one connected component if CENAPRMI individual digit recognizer cannot recognize the connected component with a high enough confidence score; on the other hand, it will stop segmenting (that is, accept the segmentation path) or not even segment the connected component if the recognizer can recognize the connected component with a high enough confidence score. Therefore, the performance of the segmentation algorithm depends on both the recognition and rejection performance of the recognizer. The recognition performance of the recognizer is good if the connected component is a complete digit. However, its rejection performance is mediocre if the connected component is a digit string. In other words, its ability to avoid accepting a digit string as a digit is not ideal. This weakness often gives the wrong feedback to the segmentation algorithm and results it errors.

Some examples in this case are given in Figure 60:

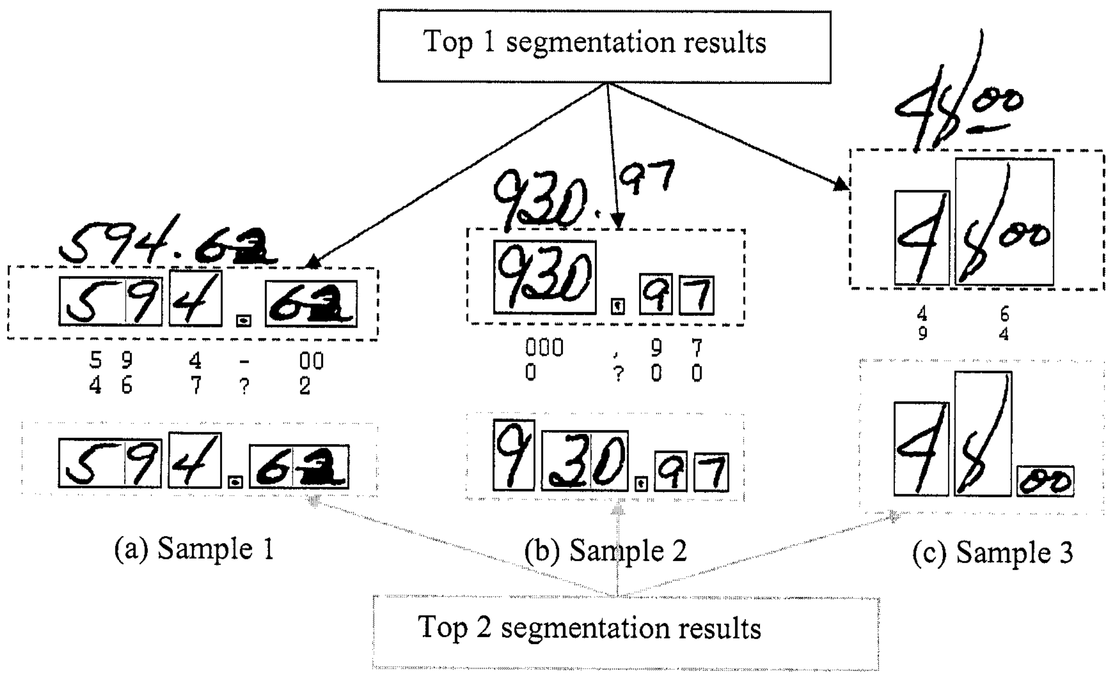
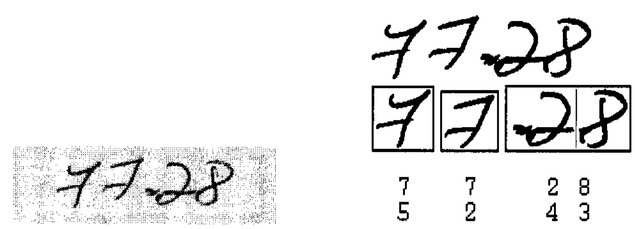



Figure 60. Some examples of segmentation errors caused by the rejection performance of CENPARMI individual digit recognizer. Note that Top 2 segmentation results (at the bottom of each image) of the above examples are correct, but suppressed by Top 1 segmentation results.

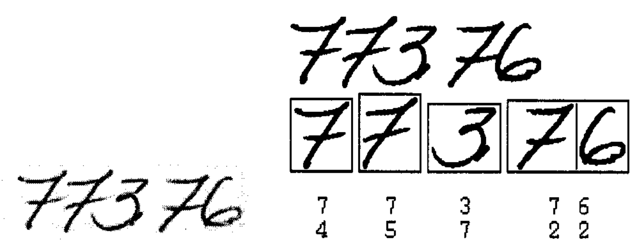
In the above Figure: in (a), the connected digit string "62" is misrecognized as "00", so its correct segmentation result becomes Top 2 choice; in (b), digit string "930" is misrecognized as "000", so its correct segmentation result becomes Top 2 choice; in (c), digit string "800" is misrecognized as "000", so its correct segmentation result becomes Top 2 choice. One possible means of solving this problem is to validate the courtesy amount recognition result with the legal amount recognition result.

#### 6.4.3.2 Errors Caused by Overlooking Decimal Points

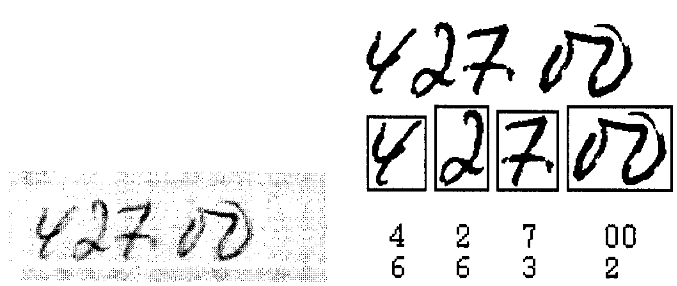
Sometimes a decimal point touches another digit in the numeral string. It can be so small that it may seem like one part of that digit and cannot be detected by the segmentation algorithm of CENPARMI CRS. Some examples in this case are shown in Figure 61. Again, these examples prove that it is necessary to validate the courtesy amount recognition results with the legal amount recognition results.




(a) Sample 1




(b) Sample 2

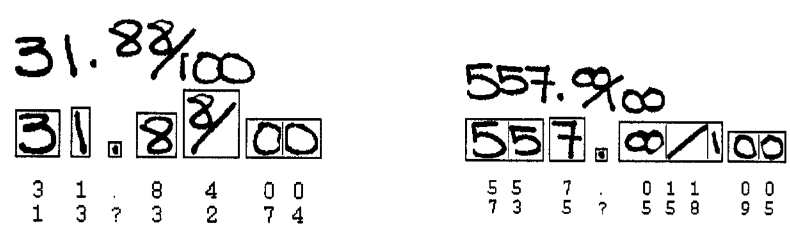



(c) Sample 3

Figure 61. Some examples of errors caused by overlooking decimal points from numeral strings.

### 6.4.3.3 Errors Caused by Some Writing Styles

Some courtesy amounts are written in special forms such as "00/100". CENPARMI CRS lacks the functionality to analyze the layout of these kinds of writing styles. Therefore, the segmentation fails with these kinds of writing styles. Some examples of these cases are shown in Figure 62:



(a) Sample 1 (b) Sample 2

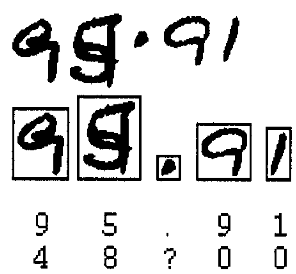
Figure 62. Some examples of errors caused by some writing styles.

### 6.4.4 Individual Digit Recognition Errors

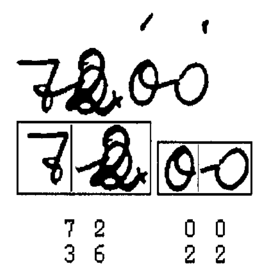
Individual digit recognition errors occur when one digit is extracted from one check image or segmented from a numeral string correctly, but still cannot be recognized correctly by the recognizer. Although the recognition rate of the digit neural network recognizer is very high (99.07%), it still makes occasional mistakes due to the poor quality of the digit or other reasons. There are three kinds of errors: completely overlapping digits, ambiguous digits, and recognizer's mistakes. They will be discussed in detail as follows:

### 6.4.4.1 Completely Overlapping Digits

Occasionally some digits on checks overlap. It is difficult to recognize them correctly. Some examples of completely overlapping digits and corresponding recognition results are shown in Figure 63:




(a) Sample 1



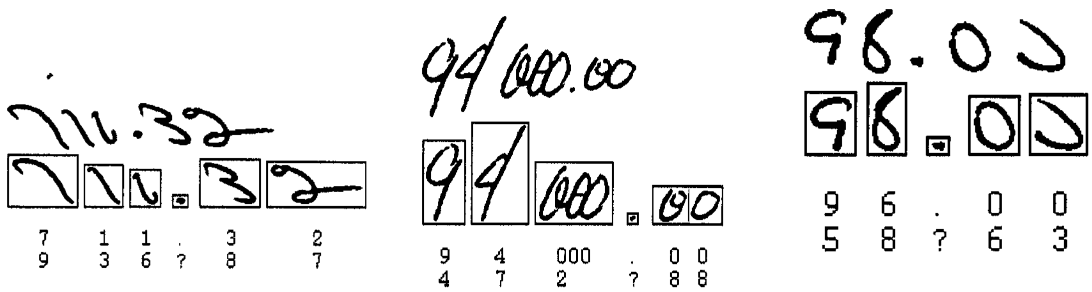

(b) Sample 2

Figure 63. Two examples of digits of poor quality.

In the above Figure: in (a), the digit "5" and the digit "4" completely overlap because one digit is written on top of the other digit, which was written earlier. In this case, even a human cannot tell which one is correct without looking at the legal amount recognition result; in (b), similar things happen to digits "2" and "0". Again, these examples prove that it is necessary to validate the courtesy amount recognition results with the legal amount recognition results.

### 6.4.4.2 Ambiguous Digits

Sometimes a digit on checks is written in an ambiguous way and even a human cannot tell which one is correct. Some examples of ambiguous digits and corresponding recognition results are shown in Figure 64:




(a) Sample 1 (b) Sample 2 (c) Sample 3

Figure 64. Three examples of ambiguous digits.

In the above Figure, both “6” and “1” in (a), both “9” and “4” in (b), and both “8” and “6” in (c), are possibly the real digits. The only way to distinguish them is to refer to the legal amount recognition results. Again these examples prove that it is necessary to validate the courtesy amount recognition results with the legal amount recognition results.

### 6.4.4.3 Recognizer’s Mistakes

Occasionally CENPARMI individual digit recognizer cannot recognize a digit of good quality on checks correctly. One example due to CENPARMI recognizer’s mistakes are shown in Figure 65:

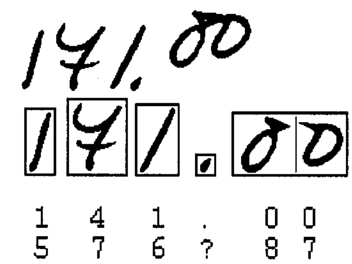
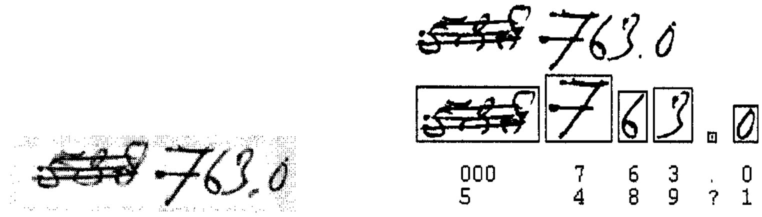



Figure 65. Example of recognizer's mistakes.

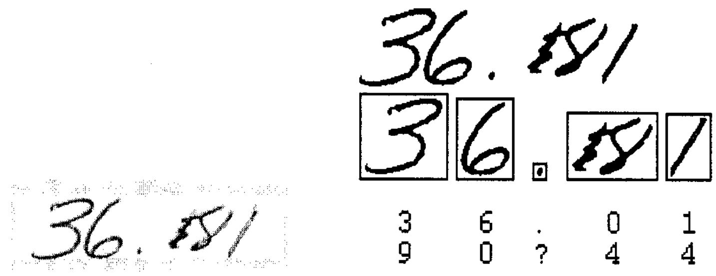
In the above Figure, the digit "7" is recognized as the digit "4" with top 1 choice, and as the digit "7" with top 2 choice. However, the digit "7" is very clear in this example and should not be recognized as a wrong digit. The problem arises from the performance limitation of CENPARMI digit neural network recognizer and can be fixed by using multiple recognizers or by validating the courtesy amount recognition result with legal amount recognition result.

### 6.4.5 Errors about Wiping Digits

Sometimes people can write a wrong courtesy amount on a check, then wipe it out and write a new one to replace it. CENAPRMI check reading system lacks the ability to recognize the area where some digits are wiped out. Some examples in this case are shown in Figure 66:




(a) Example 1




(b) Example 2

Figure 66. Some examples of errors about wiping digits.

Future work can focus on dealing with the errors described above. Some advices about the future works will be given in Chapter 7.2.

# Chapter 7

# Conclusion

## 7.1 Summary

In this thesis, the structure of CENPARMI CRS has been described, its constituted modules have been introduced, and the functionalities of its three main modules have been shown step by step: courtesy amount recognition, legal amount recognition, and date recognition.

Additionally, a new feedback-based segmentation algorithm for courtesy amount, which is the main contribution of my thesis, has been proposed in detail. The original segmentation algorithm can segment the input numeral string only once. If the segmentation results cannot be accepted by the digit recognizer, the segmentation algorithm will fail. Therefore, a feedback-based segmentation algorithm has been proposed, which can adjust the parameters of the segmentation algorithm and re-segment the input numeral string when the segmentation algorithm fails. The experiments based on the Quebec Bell check database have shown that a considerable improvement on the recognition rate can be achieved through this process.

Two rejection strategies in the segmentation algorithm were also presented. The first strategy was the algorithm for rejecting unreasonable segmentations which can be accepted by the individual digit recognizer. The second strategy was the rejection algorithm used to avoid accepting a digit string as one digit. Both of them are implemented by an analysis of the positions of segmented sub-images.

Because an incomplete digit "5" was accepted by CENPARMI individual digit recognizer with a high confidence score, another method specifically designed to deal with digit "5" with a detached stroke was presented. Using the method, any digit which was accepted as "5" would still be grouped with its right connected component and segmented.

Moreover, some improvement in the pre-processing and post-processing modules was proposed: firstly, a new border noise removal algorithm was proposed to avoid removing strokes of numeral strings erroneously; secondly, a semantics-based algorithm was proposed to remove extra commas, periods, and horizontal strokes "-" in the recognition results; thirdly, an algorithm for detecting implicit decimal points was proposed, which was based on the analysis of the positions of the last two digits in the numeral string.

Finally, a convolutional neural network (CNN) recognizer for "00" and "000" was presented. It was integrated into the CRS to recognize "00" and "000" directly in order to avoid having to segment "00" and "000" into separate digits. Although one of the advantages of CNN was that it had few parameters so that it did not need too much data for training, we still needed to collect training data in order to achieve a good performance. A two-step coarse-to-fine method was used to collect "00" and "000 data as follows: first, use a program to collect "00" and "000" data from the Quebec Bell check database coarsely; then, pick up "00" and "000" data manually; finally, an elastic deformation algorithm was used to enlarge the data set of "00" and "000". According to the experimental results, the recognition rate of the CNN for "00" was much higher than that of the traditional "00" neural network recognizer. The recognition rate of the CNN for "000" was also good.

The experiments based on the Bell Quebec check database showed that the courtesy amount recognition rate of the new CENPARMI CRS was almost twice that of the original CENPARMI CRS and was comparable to that of CRSs developed by other researchers or companies. However, new CENPARMI CRS did not consider how to suppress the substitution rate. Further improvements could be made by validating the courtesy amount recognition results with the legal amounts. It is expected that the validation process would achieve improved results in new CENPARMI CRS.

## 7.2 Future Work

Although the performance of CENPARMI CRS has improved by the feedback-based segmentation algorithm, the new CNN recognizer for "00" and "000", and other algorithms in the pre-processing and post-processing modules, further work is still needed to improve its reliability rate and suppress its substitution rate. Based on the error analyses in Chapter 6, future research may include the following problems:

1. Validation procedure

   In order to improve the reliability and robustness of CENPARMI CRS, a validation procedure should be integrated into CENPARMI CRS. The validation procedure involves validating recognition results of the legal amount with those of courtesy amount to improve the final recognition rate of the CRS. Although the recognition rate of the legal amount is relatively low, the recognition results of the legal amount can help to verify ambiguous courtesy amount recognition results or to correct errors in courtesy amount recognition results.

2. Layout analysis of courtesy amount and the recognition of characters "X" and "/"

   There are some special layouts for courtesy amounts on checks, such as "XX/XX", "XX/00", "00/100". CENPARMI CRS currently lacks the ability to analyze these layouts and recognize characters "X" and "/". Therefore, a better layout analysis of courtesy amounts and the recognition of characters "X" and "/"could improve the performance of CENPARMI CRS.

3. Enlarge training data

   The performance of the digit recognizer in CENPARMI CRS is very important for the recognition-based segmentation algorithm. Although the recognition rate of CENPARMI digit recognizer is high (99.07%), there are still some cases where the digit recognizer fails to recognize correct individual digits. In order to improve its performance further, a larger numeral database is still needed for training. It is a good idea to further categorize the numeral database by the country of origin, because people from different countries may have different writing habits.

4. Rejection performance of the individual digit recognizer

   In the segmentation algorithm of CENPARMI CRS, it is very important to reject the wrong segmentation paths or groups and to accept correct ones. Therefore, it demands that the recognition performance and rejection performance of the individual digit recognizer are both excellent. Currently, the rejection performance of the recognizer is very weak, which causes some segmentation errors. So improving its rejection performance will be very helpful for the courtesy amount recognition.

5. Courtesy amount extraction

   From the error analyses, we can see that there are some courtesy amount extraction errors in CENPARMI CRS. If the courtesy amount is extracted erroneously, there is no way to correctly recognize it. Therefore, it is important to correct this kind of errors.

# Appendix A. Explanation of the Format of Recognition Results of CENPARMI CRS

The format of courtesy amount, legal amount, and date recognition results of CENPARMI CRS are explained respectively as follows:

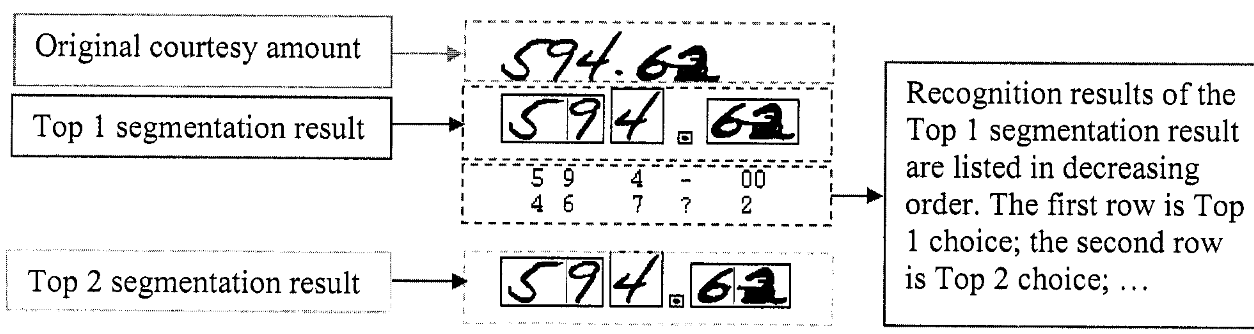



Figure 67. Explanation of the format of courtesy amount recognition results

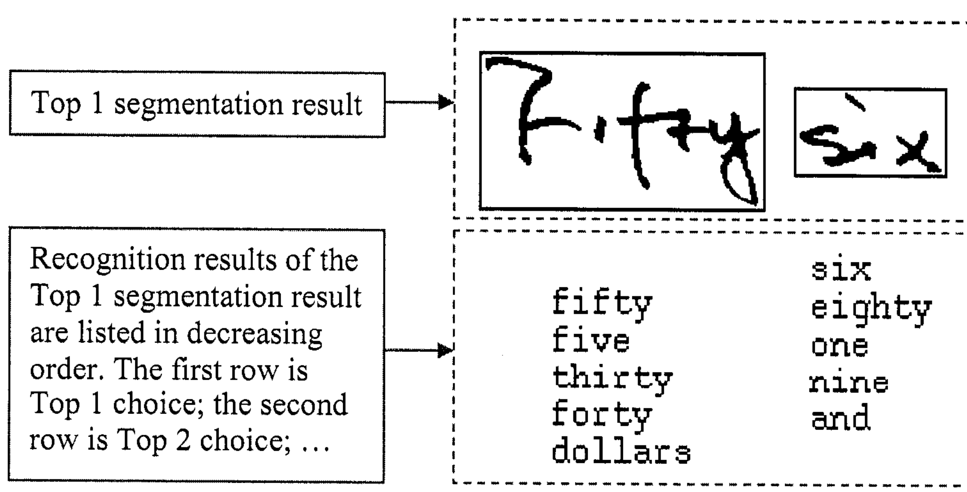



Figure 68. Explanation of the format of legal amount recognition results

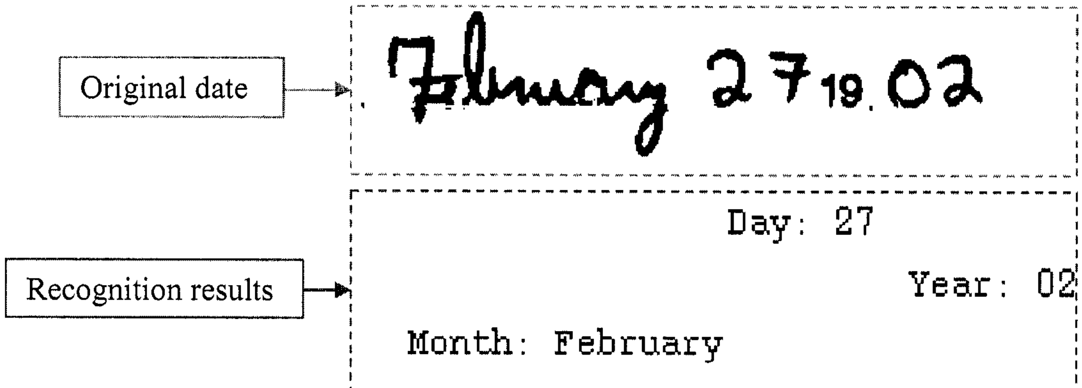



Figure 69. Explanation of the format of date recognition results

# References


[1] N. W. Strathy, "A Method for Segmentation of Touching Handwritten Numerals," Master Thesis, Department of Computer Science, Concordia University, Montreal, Canada, 1993.

[2] K. Liu, C.Y. Suen, M. Cheriet, J. N. Said, C. Nadal, and Y.Y Tang, "Automatic Extraction of Baselines and Data from Check Images," Special Issue of International Journal of Pattern Recognition and Artificial Intelligence on Automatic Bankcheck Processing, Vol. 11, No. 4, pp. 675-697, 1997.

[3] X. Ye, M. Cheriet, C. Y. Suen, and K. Liu, "Extraction of Bankcheck Items by Mathematical Morphology," International Journal of Document Analysis and Recognition, Vol. 2, No. 2-3, pp. 53-66, 1999.

[4] J. H. Kim, K. K. Kim, and C. Y. Suen, "An HMM-MLP Hybrid Model for Cursive Script Recognition," Pattern Analysis and Applications, pp. 314-324, 2004.

[5] D. Guillevic, "Unconstrained Handwriting Recognition Applied to the Processing of Bank Cheques," Doctoral thesis, Department of Computer Science, Concordia University, Montreal, Canada, 1995.

[6] J. Zhou, "Segmentation of Legal Amounts on Bankchecks," Master Thesis, Department of Computer Science, Concordia University, Montreal, Canada, 2001.

[7] Q. Xu, "Automatic Segmentation and Recognition System for Handwritten Dates on Cheques," Doctoral thesis, Department of Computer Science, Concordia University, Montreal, Canada, 2002.

[8]S. N. Srihari, and E. J. Keubert, “Integration of Hand-Written Address Interpretation Technology into the United States Postal Service Remote Computer Reader System,” Proceedings of the fourth International Conference on Document Analysis and Recognition, Ulm, Germany, 1997.

[9] ABBYY software company, “State Tax Service of Russia Uses ABBYY Technology in Form Input Project,” http://www.abbyy.com/company/case_studies.asp?param=32188, 2007.

[10] P. K. Kim, H. Lim, K. Y. Rho, K. Y. Lee, and S. M. Kim, “Automatic Tax-sheet Input System Using Handwritten Numeral Recognition,” IEEE TENCON - Speech and Image Technologies for Computing and Telecommunications, pp. 527-530, 1997.

[11] N. Gorski, V. Anisimov, E. Augustin, O. Baret, and S. Maximov, “Industrial Bankcheck Processing: the A2iA CheckReader,” International Journal of Document Analysis and Recognition, Vol. 3, No. 4, pp. 196–206, 2001.

[12] G. Dimauro, S. Impedovo, G. Pirlo, and A. Salzo, “Automatic Bankcheck Processing: a New Engineered System,” International Journal of Pattern Recognition and Artificial Intelligence, S. Impedovo et al., Eds., pp. 467-503, 1997.

[13] P. Rafael and G. Amar, “A System for Processing Handwritten Bank Checks Automatically,” MIT Sloan Working Paper No. 4346-02; Eller College Working Paper No. 1028-05, 2002.

[14] T. Breuel, “Segmentation of Hand-printed Letter Strings Using a Dynamic Programming Algorithm,” Proceedings of the Sixth International Conference on Document Analysis and Recognition, pp. 821-826, Seattle, USA, 2001.

[15] Verifone Company, "Check Imaging White Paper," http://bvcontent.verifone.com/VeriFone/Attachment/20040623/check_imaging_white_paper.pdf, 2004.

[16] N. Greco, D. Impedovo, M. G. Lucchese, A. Salzo, and L. Sarcinella, "Bankcheck Processing System: Modifications Due to the New European Currency," Proceedings of the Seventh International Conference on Document Analysis and Recognition, pp. 343-348, Edinburgh, Scotland, 2003.

[17] S. Impedovo, P. S. P. Wang and H. Bunke, Eds., "Automatic Bankcheck Processing," World Scientific Publication, Singapore, 1997.

[18] T. C. Jee, E. J. Kim, and Y. Lee, "Error Correction of Korean Courtesy Amounts in Bank Slips Using Rule Information and Cross-referencing," Proceedings of the Fifth International Conference on Document Analysis and Recognition, pp. 95–98, Bangalore, India, 1999.

[19] H. Su, B. Zhao, F. Ma, S. Wang, and S. Xia, "A Fault-tolerant Chinese Check Recognition System, " S. Impedovo, P. S. P. Wang and H. Bunke, Eds., "Automatic Bankcheck Processing," pp. 109-132, World Scientific Publication, Singapore, 1997.

[20] K. Hussein, A. Agarwal, A. Gupta, and P. S. P. Wang, "A Knowledge-based Segmentation Algorithm for Enhanced Recognition of Handwritten Courtesy Amounts," Pattern Recognition, Vol. 32, No. 2, pp. 305–316, 1999.

[21] G. Kaufmann and H. Bunke, "Automated Reading of Cheque Amounts," Pattern Analysis and Applications, Vol. 3, pp. 132–141, 2000.

[22] C. O. de Almendra Freitas, A. el Yacoubi, F. Bortolozzi, and R. Sabourin, "Brazilian Bankcheck Handwritten Legal Amount Recognition," Proceedings of XIII Brazilian

Symposium on Computer Graphics and Image Processing, pp. 97–104, Gramado, Rio Grande do Sul, Brazil, 2000.

[23] K. Ueda and K. Matsuo, "Automatic Seal Imprint Verification System for Bankcheck Processing," Proceedings of the Third International Conference of Information Technology and Applications, Vol. 1, pp. 768–771, Sydney, Australia, 2005.

[24] C. Kim, K. M. Kim, and C. Y. Suen, "Word Separation in Handwritten Legal Amounts on Bankchecks Based on Spatial Gap Distances," R. Orchard et al., Eds., "Innovations in Applied Artificial Intelligence," IEA/AIE 2004, LNAI 3029, pp. 453-462, 2004.

[25] A. Agarwal, A. Gupta, and K. Hussein, "Bankcheck Analysis and Recognition by Computer," H. Bunke and P. S. P. Wang, "Handbook of Character Recognition and Document Image Analysis," pp. 623-651, World Scientific, Singapore, 1997.

[26] S. Knerr, V. Anisimov, O. Baret, N. Gorski, D. Price, and J. C. Simon, "The A2iA Intercheques System: Courtesy Amount and Legal Amount Recognition for French Checks," International Journal of Pattern Recognition and Artificial Intelligence, S. Impedovo et al., Eds., Vol. 11, No. 4, pp. 505-548, 1997.

[27] G. Dzuba, A. Filatov, D. Gershuny, I. Kil, and V. Nikitin, "Check Amount Recognition Based on the Cross Validation of Courtesy and Legal Amount Fields," International Journal of Pattern Recognition and Artificial Intelligence, S. Impedovo et al., Eds., Vol. 11, No. 4, pp. 639-655, 1997.

[28] G. Kim and V. Govindaraju, "Bankcheck Recognition using Cross Validation Between Legal and Courtesy Amounts," International Journal of Pattern Recognition and Artificial Intelligence, S. Impedovo et al., Eds., Vol. 11, No. 4, pp. 657–674, 1997.

[29] C. Y. Suen, Q. Xu, and L. Lam, "Automatic Recognition of Handwritten Data on Cheques—Fact or Fiction?," Pattern Recognition Letters, Vol. 20, pp. 1287–1295, 1999.

[30] A2iA Company, http://www.a2ia.com//Web_Bao/Applications082A2iA000nbsp075CheckReader-Eng.aspx, 2007.

[31] Q. Xu, L. Lam, and C. Y. Suen, "Automatic Segmentation and Recognition System for Handwritten Dates on Canadian Bank Cheques," Proceedings of the Seventh International Conference on Document Analysis and Recognition, pp. 704-708, Edinburgh, Scotland, 2003.

[32] L. Lam, C. Y. Suen, D. Guillevic, N. W. Strathy, M. Cheriet, K. Liu, and J. N. Said, "Automatic Processing of Information on Cheques," Proceedings of the International Conference on Systems, Man & Cybernetics, pp. 2353-2358, Vancouver, Canada, 1995.

[33] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Automatic Recognition of Handwritten Numerical Strings: A Recognition and Verification Strategy," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 24, No. 11, pp. 1438-1454, 2002.

[34] S. M. Choi and I. S. Oh, "A Segmentation-Free Recognition of Handwritten Touching Numeral Pairs Using Modular Neural Network," International Journal of Pattern Recognition and Artificial Intelligence, Vol. 15, No. 6, pp. 949-966, 2001.

[35] X. Wang, V. Govindaraju, and S. Srihari, "Holistic Recognition of Touching Digits," Pattern Recognition, Vol. 33, No.12, pp. 1967–1973, 2000.

[36] U.l. Pal, A. Belad, and C. Choisy, "Touching Numeral Segmentation Using Water Reservoir Concept," Pattern Recognition Letters, Vol. 24, No. 1, pp. 261-272, 2003.

[37] K. K. Kim, J. H. Kim, and C. Y. Suen, "Recognition of Unconstrained Handwritten Numeral Strings by Composite Segmentation Method," Proceedings of the 15th International Conference on Pattern Recognition, Vol. 2, pp. 594-597, Barcelona, Spain, 2000.

[38] T. M. Ha, M. Zimmermann and H. Bunke, "Off-line Handwritten Numeral String Recognition by Combining Segmentation-based and Segmentation-free Methods," Pattern Recognition, Vol. 31, No. 3, pp. 257-272, 1998.

[39] N. W. Strathy, F. Said and C. Y. Suen, "A Sequential Method of Extracting Contour Chains from an Image," Proceedings of the 12th International Conference on Pattern Recognition, pages, Vol. 2, pp. 580–582, Jerusalem, Israel, 1994.

[40] L. Zhang, "Recognition of Courtesy Amounts on Bank Checks Based on a Segmentation Approach," Master Thesis, Department of Computer Science, Concordia University, Montreal, Canada, 2001.

[41] G. F. Houle, D. B. Aragon, and R. W. Smith, "A Multi-Layered Corroboration-Based Check Reader," Proceedings of IAPR Workshop on Document Analysis Systems, pp. 495-543, Malvern, USA, 1996.

[42] E. Lethelier, M. Leroux, M. Gilloux, "An Automatic Reading System for Handwritten Numeral Amounts on French Checks," Proceedings of the Third International Conference on Document Analysis and Recognition, pp. 92–97, Montreal, Canada, 1995.

[43] L. Heutte, P. Barbosa-Pereira, O. Bougeois, J. V. Moreau, B. Plessis, P. Courtellemont, and Y. Lecourtier, "Multi-bank Check Recognition System: Consideration on the Numeral Amount Recognition Module," S. Impedovo, P. S. P.

Wang and H. Bunke, Eds., "Automatic Bankcheck Processing", pp. 133-156, World Scientific, Singapore, 1997.

[44] M. Suwa, "Segmentation of Connected Numerals by Graph Representation," Proceedings of the Eighth International Conference on Document Analysis and Recognition, pp. 750-754, Seoul, Korea, 2005.

[45]C. R. Nohl, C. J. C. Burges, and J. I. Ben, "Character-based Handwritten Address Word Recognition with Lexicon," Proceeding of USPS ATC, Vol. 1, pp. 167-180, USA, 1992.

[46] G. Congedo, G. Dimauro, S. Impedovo, and G. Pirlo, "Segmentation of Numeric String," Proceedings of the third International Conference on Document Analysis and Recognition, Vol. 2, pp. 1038-1041, Montreal, Canada, 1995.

[47] Z. Lu, Z. Chi, W. C. Siu, and P. Shi, "A Background-thinning-based Approach for Separating and Recognizing Connected Handwritten Digit Strings," Pattern Recognition, Vol. 32, pp. 921-933, 1999.

[48] Y. K. Chen and J. F. Wang, "Segmentation of Single- or Multiple-touching handwritten Numeral String Using Background and Foreground Analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 11, pp. 1304–1317, 2000.

[49] F. Lauer, C. Y. Suen and G. Bloch, "A Trainable Feature Extractor for Handwritten Digit Recognition," Pattern Recognition, Vol. 40, pp. 1816-1824, 2007.

[50] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," Proceedings of the Seventh

International Conference on Document Analysis and Recognition, pp. 958-962, Edinburgh, Scotland, 2003.

[51] M. O'Neill, "Neural Network for Recognition of Handwritten Digits," http://www.codeproject.com/KB/library/NeuralNetRecognition.aspx, 2006.

[52] Unisys company, "SoftCAR+ Diamond Capabilities Guide," http://www.unisys.com/products/payment_systems/amount_recognition/specifications.htm, 2008

[53] Orbograph company, "OrboCAR Accura Accuracy Achievement Release," http://orbograph.com/data/uploads/Accura%20PR.pdf, 2008

[54] Wu Ding, C. Y. Suen, A. Krzyzak, "A New Courtesy Amount Recognition Module of a Check Reading System," Proceedings of the 19th International Conference on Pattern Recognition, Tampa, USA, 2008. (Accepted)