Segmentation of Date Field on Bank Cheques

Louisa Lam[1,2], Rong Fan[1], Qizhi Xu[1], and Ching Y. Suen[1]

[1] Centre for Pattern Recognition and Machine Intelligence
Suite GM-606, Concordia University
1455 de Maisonneuve Blvd. West
Montréal, Québec H3G 1M8, Canada
[2] Department of Mathematics
Hong Kong Institute of Education
10 Lo Ping Road, Tai Po, Hong Kong

Abstract. This article describes a system being developed to segment handwritten date fields on bank cheque images. The handwritten information extracted from the date zone is segmented into day, month and year, and a hypothesis is also made on the writing style of the month (in word or digits). The system has been implemented and tested on cheque images. Subsequent modifications have also been designed and implemented to include contextual information in the determination of segmentation points. Results have shown that the system is effective; with continuing improvements, the system is expected to be a useful component for processing the date written on cheques.

1   Introduction

This paper describes a system being developed to segment the date information handwritten on bank cheques. The ability to automatically process handwritten dates is important in application environments where cheques cannot be cashed prior to the dates shown, since any delay would entail significant financial costs when large numbers of cheques are involved. On the other hand, developing an effective date processing system is very challenging due to the high degree of variability and uncertainty present in the dates handwritten on standard bank cheques, and the different recognizers required for processing the information. Perhaps for this reason, there has been no published work on this topic until recently, when work on the date fields of machine-printed cheques was reported [4], and this reference also considers date processing to be the most difficult target in cheque processing, given that it has the worst segmentation and recognition performance.

When *a priori* knowledge is available about the format or style used in representing the date, it is very difficult and computationally inefficient to develop a system to process the entire *date_zone* at the same time. Therefore, segmentation is introduced so that the problem can be reduced to the processing of separate components. The handwritten date_zone image is segmented into three subimages, and each subimage is assigned to the *Day*, *Month* or *Year* field. A decision is also made on the writing style of the *Month*.

This approach is different from that of [4], where the algorithm first identifies the bigrams '95', '96', '97', which are common to all date patterns (e.g. Jan 25, 1996, JANUARY 25 1996, 01/25/96. 01-25-96, etc.). It next examines the left neighbouring characters to determine whether they are the completion of year (i.e. 1996), or a delimiter for the day or month. A tool shell has been designed which supports both word spotting through inexact matching, and the use of wild card characters to search for formatted patterns such as **/**/**, **-**-**, etc. A sample of 500 valid *date* fields were segmented by a human operator and tested. The complete recognition performance with the automated field segmentation is estimated at 44% to 49% by using four commercial recognition devices.

2   Cheque Databases

During the design and development phase of an automatic cheque processing system, it was necessary to have access to a large quantity of bank cheques in order to gain insight into the various ways in which cheques can be written. However, such data did not exist at that time. Due to security and confidentiality considerations, it was also very difficult to have access to real cheques from banks or utility companies even for research purposes. For this reason, this research centre decided to create its own databases.

For the first database (Database 1), a blank cheque was carefully designed to have a size and layout similar to that of regular bank cheques. Its background is white, and all the lines are printed in special drop-out ink which is invisible to the scanner. This facilitates the extraction of written information, and also produces extracted images of better quality for the initial development of item extraction and recognition processes. The cheques were filled in by university students, after which they were scanned and stored as binary images, ready for the extraction and further processing of each item of written information [3]. Altogether 4564 cheque images were obtained this way.

For testing and further refinement of the cheque processing system, Database 2 was created. This consists of over 12,000 images from real-life standard cheques, on which the handwritten information had been completed by university students and staff, as well as employees of a major utility company. Results on both databases are contained in this article.

3   Writing Style Analyses

In North America, all bank cheques have a similar layout. The *date_zone* is always positioned at the upper right corner of each cheque. Machine-printed digits '1' and '9' appear near the right end of the *date_zone*, thus separating the *date_zone* into two parts. The part to the right of the printed "19" is intended for entering the last two digits of *Year*. The part on the left is intended for *Day* and *Month*, and it is completely blank, which implies there is no pre-defined position for *Day* or *Month*.

In addition, there is no restriction on the writing style. *Month* can be written in either digit or word form, while punctuations (period ('.'), comma (','), slash ('/') and hyphen ('-')) or a space can be used to identify the end of a field. Hence, great flexibility exists for a writer when entering the *Day* and *Month* fields, and dates can be represented by a large variety of writing styles, some of which are shown in Fig. 1.

Fig. 1. Examples of *date_zone* images

As shown in Fig. 1, the contents of *date_zone* can be either pure digits when *Month* is written in digits, or a combination of digits and cursive scripts when *Month* is written as a word. *Month* can be placed either before or after *Day*. In general, the writing styles of *date_zones* extracted from bank cheques can be expressed in the following 4 patterns, with possible additions of punctuations or suffixes (such as 'st', 'nd', etc.):

$$dd\ mm\ yy$$
$$mm\ dd\ yy$$
$$dd\ MM\ yy$$
$$MM\ dd\ yy$$

In the above, *dd* designates *Day* written in digits, *mm* designates *Month* written in digits, *MM* designates *Month* written in word form, and *yy* designates *Year* written in digits.

To represent a certain date, for example "February 26, 1997", each of the following 8 variations can be used:

February(,) 26$^{(th)}$(,) **1997**	(0)2/26(/) **1997**
Feb(.) 26$^{(th)}$(,) **1997**	(0)2-26(-) **1997**
26$^{(th)}$ February **1997**	26/(0)2(/) **1997**
26$^{(th)}$ Feb(.) **1997**	26-(0)2(-) **1997**

In the above list, items shown within parentheses denote entities which may or may not be present. "**19**" in bold are the printed digits which appear on all bank cheques representing the century. Suffixes such as 'st', 'nd', 'rd', or 'th' can be written either as superscripts or at the same horizontal position as the rest of the *date_zone* information.

3.1   Statistics on Punctuations

On handwritten date images, no uniform spacing or separator is present between *Day* and *Month*. To separate these two fields, punctuations are often used by cheque writers, and different punctuations are used. Studies on our cheque databases show that period and comma are used mostly when *Month* is represented in word form, while slash and hyphen are mostly written when *Month* is represented by digits.

Among 4564 such cheques of Database 1, 84.09% have *Month* written in word form, and 15.91% have *Month* written in numerals. Among the 3837 images where *Month* is in word form, punctuations appear on 765 images (about 19.94%). Detailed statistics about punctuation usage in this case are shown in Table 1. Among the 726 *date_zone* images where *Month* appears in digits, only 17 of them do not contain any punctuations. Table 2 illustrates the punctuation usage when all three fields of a *date_zone* are written in numerals.

Table 1. Use of punctuations when *Month* is in word

	No. of images	Percentage (%)
Total	765	100.00
Period '.' only	537	70.20
comma ',' only	154	20.13
Both *period* & *comma*	38	4.97
Others	36	4.70

Table 2. Use of punctuations when *Month* is in digits

	No. of images	Percentage (%)
Total	709	100.00
Slash '/' only	423	59.67
Hyphen '-' only	234	33.00
Both *slash* & *hyphen*	2	0.28
Others	50	7.05

"Others" in Table 1 refer to those *date_zone* images where *Month* is written in word, but punctuations used are slash or hyphen. "Others" in Table 2 refers to cases where *Month* is written in digits, but punctuations used are period or comma.

In this database, 3089 images contain no punctuation at all, when *Month* is almost always written in word (only 0.55% of such images have *Month* written in numerals).

As discussed previously, if punctuations and suffixes are not considered, *date_zone* writing styles can be classified into 4 categories. Pertinent statistics have also been gathered on these aspects. As shown in Table 3, when *Month*

is written in word, the *Month* word appears before *Day* in about half of the samples, and the reverse is true for the rest. However, when *Month* is written in numerals, more people tend to write it after *Day*. In addition, on 237 *date_zone* images, both *Month* and *Day* are written in numerals, but the sequence cannot be determined since both fields have values not exceeding 12.

Table 3. Statistics about *date_zone* writing patterns

Writing style pattern	No. of images
MM dd yy	1936
dd MM yy	1901
dd mm yy	442
mm dd yy	46

4   Feature Description

The input of the automatic date processing system consists of binary *date_zone* images, each of which is decomposed into a set of connected components, which are analyzed according to a set of features designed to detect the four types of punctuations as well as printed digits '1' and '9'.

In general, two categories of features, shape features and spatial features [1], can be considered. Shape features deal with the geometric aspects of each connected component, particularly its appearance and measurements. Spatial features deal with the contextual aspects of each connected component, which provide important information especially when the objective is to process a text line. They are used to describe the location of each connected component with respect to the entire *date_zone* image as well as its neighbouring components.

The shape features used to describe punctuations are *high_density*, *narrow*, *flat*, *slope*, *small*, *simple_curve* and *no_innerloop*; while the spatial features used consist of *exceed_neighbour*, *at_middlezone*, *mid_to_neighbour*, *below_lowerhalf* and *low_to_left* [2]. The selection of features for detecting each punctuation was based on experimentation with all the developed shape and spatial features. The following four sets of features were chosen to characterize the punctuations:

- Slash: *narrow* and *exceed_neighbour*;
- Hyphen: *high_density*, *flat*, *at_middlezone* and *mid_to_neighbour*;
- Period: *high_density*, *small* and *below_lowerhalf*;
- Comma: *narrow*, *small* and *below_lowerhalf*.

5   Segmentation Strategies

The purpose of *date_zone* image segmentation is to divide the entire image into three subimages representing *Day*, *Month* and *Year* respectively, and also to generate a hypothesis on how *Month* is written so that it can be processed using the appropriate recognizer.

Since machine-printed digits '1' and '9' are present on all standard bank cheques, it would be more reliable to start segmentation by searching for the printed "19", because this is a more stable part of the image than the rest of the *date_zone* which contains completely unconstrained handwritten information. To detect these printed characters, the connected components of the *date_zone* image are examined in sequence from the right until a printed '1' is found or the left end of the date image is reached. The '1' is identified by the *slope*, *narrow* and *high_density* features since the stroke is mostly vertical, relatively narrow and occupies most of its bounding box. Once this is located, its right neighbour is examined to see if it is the printed digit '9'. If this component contains exactly one inner loop, and it has approximately the same height as its left neighbour, these two adjacent connected components are considered to be the printed digits "19". These components are then removed, resulting in two separate subimages, *Day&Month* and *Year*. The subimage on the left is assumed to be the *Day&Month* subimage, and it will be further segmented into *Day* and *Month* subimages.

5.1   Observations on Day and Month Segmentation

As observed previously, four types of punctuations are frequently seen in *date_zone* images. The position of a punctuation can mark the end of *Day* or *Month* field, or a division between *Day* and *Month*. Based on the statistics presented in Tables 1 and 2, the type of punctuation written after *Month* can suggest how this field is written. The presence of slash or hyphen strongly implies that *Month* is written in digits, while the presence of period or comma implies a word. Therefore, once the *Day&Month* subimage is scanned for these four types of punctuations, a segmentation between *Day* and *Month* is suggested.

It has been observed that a higher proportion of *date_zone* images do not contain punctuation(s); in these cases, *Month* is almost always written as a word, and most people tend to leave a gap between *Day* and *Month*. The locating of an interword gap can then provide a suitable segmentation point between *Day* and *Month*.

For the detection of an interword gap, many algorithms can be used to compute the distances between pairs of connected components [5,6]. In this work, we consider the maximum gap to occur where the maximum distance between neighbouring components occurs on the largest number of scan lines. This method is completely independent of threshold values, and it is effective and computationally efficient. Statistics show that among those cheques where no punctuation is written between *Day* and *Month*, or only one punctuation is written at the end of *Day&Month* subimage, 80.11% of the maximum gaps are correctly detected by this method.

After punctuations have been detected on the *Day&Month* subimage, a set of empirical rules are applied to obtain the result of segmentation and hypotheses. The segmentation process separates the *Day&Month* subimage into two parts, while hypotheses are made on which part represents *Day* and which represents

Month, and whether the *Month* field is written in word or digits. The rules used at this step are based on:

– the number of punctuations detected;
– locations of the punctuations, and;
– types of punctuations detected.

6   Results of Segmentation

Table 4 shows the performance of *date_zone* image segmentation on the images of Database 1, where "correct" means that the cutting positions are properly located and hypotheses are correctly generated, so that each field can be sent to the appropriate recognizer (digit or cursive word).

Table 4. Performance of *date_zone* segmentation on Database 1

Total no.	Correct (%)	Reject (%)	Error (%)
4564	74.96	7.82	17.22

In Table 4, the rejection of a *date_zone* image can be due to one of the following reasons:

(a) Improper binarization causes too many (broken) components to be created in the image. Such images are simply rejected and excluded from further processing.
(b) Some *date_zone* images contain only one connected component representing the entire *date_zone*. Since this provides no clues to segmentation, these images are rejected.
(c) Printed digits "19" cannot always be detected because they are almost completely overwritten by other fields, or touching the rest of the image. Sometimes '1' and '9' are distorted so that their features can not be detected properly. As locating "19" is the first step in *date_zone* segmentation, these images are rejected.
(d) The *Day&Month* subimage contains only one connected component, which provides no clues for segmentation.

Punctuation detection is also a critical step in *Day&Month* subimage segmentation as described above, and therefore their correct detection is significant for the performance of this process. Table 5 gives the performance of each punctuation detection algorithm on Database 1. It shows that the methods proposed by this research work are effective for this purpose.

However, errors do occur during this step, in that components of other fields are misinterpreted as punctuations. For example, broken strokes (most probably due to binarization) may be mis-interpreted as punctuations, and the digit '1' or letter '*l*' can be identified as slash '/'. For a more accurate determination of punctuations and segmentation points, a further strategy has been developed and described below.

Table 5. Performance of punctuation detection on Database 1

	Slash (%)	Hyphen (%)	Period (%)	Comma (%)
Correct	94.72	94.02	92.80	93.63

7   Segmentation with Confirmation

Given that *date_zone* image segmentation depends heavily on the accurate determination of a separator between the *Day* and *Month* fields, it is important to improve the accuracy in the detection of this separator, which may be a punctuation or an interword gap. This separator can be determined from:

(a) Presence of a significant gap or punctuation within the *Day&Month* subimage, or
(b) A transition between digits and letters.

Consequently, we develop and add a confirmation procedure to our segmentation strategy, so that a two-level strategy is implemented. Using this strategy, more candidates for punctuation and gap are determined than before. However, these candidates are considered to be separators only if they satisfy more stringent conditions than used previously. Otherwise the confirmation procedure is applied at the second level by considering the contextual information or the nature of the subimage on either side of the candidate.

For example, locating the interword gap in the *Day&Month* subimage can be a difficult task because this gap may not always appear as the widest gap observed in the *Day&Month* subimage when users write the date freely. However, if it can be determined that a gap occurs at the transition between numeric and alphabetic fields, then this gap can be considered to be gap_{DM}, the gap between the *Day* and *Month* fields. Similarly, when a candidate for slash is considered, this candidate can be confirmed as slash when subimages on both sides of the candidate show high likelihoods of being numeric, and not confirmed as slash when both subimages are highly unlikely to be numeric. (In the former case, we apply our experimental knowledge that slashes are often used when both *Day* and *Month* are represented numerically, and also the *a priori* condition that each such field should contain at most two digits, so that a symbol appearing between two numeric images should be considered a separator rather than digit '1').

For our purpose, the likelihood of a subimage being numeric is determined by a combination of the following information:

(a) the confidence value and the number of digits in the subimage returned by a connected digit recognizer [7], and
(b) structural features of the subimage. These consist of the maximum number of horizontal runs, the sum of the numbers of peaks and valleys, and the number of inner loops in the subimage. These numbers should be below certain thresholds for numeric subimages, which are usually simpler than alphabetic ones. The thresholds are determined through experimentation with the training set.

Based on the above information, a measure can be obtained that represents the likelihood of a subimage being numeric, and this measure is used in the confirmation process described above. The effectiveness of this measure $Confid_{numeric}$ in differentiating between alphabetic and numeric images can be seen from results obtained on 4205 samples of month words and 4000 numeric samples from Database 2. These are shown in Fig. 2, and it can be seen that there is a strong correlation between very high (low) values of $Confid_{numeric}$ and numeric (alphabetic or word) samples. Of course, the accurate determination of $Confid_{numeric}$ is also important for the detection of the style of *Month*, and an appropriate recognizer can be selected for subsequent processing.
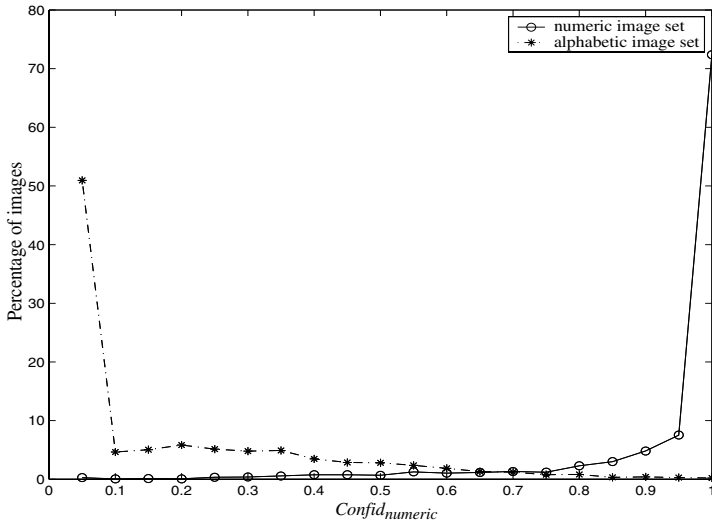

Fig. 2. Relationship of $Confid_{numeric}$ to numeric and word images

Using this measure, the two-level strategy is implemented and tested on 809 cheques of Database 2, and the results are given in Table 6, for both the previous and two-level methods.

Table 6. Performance of two methods for *date_zone* segmentation (809 images)

	Correct (%)	Reject (%)	Error (%)
Two-level strategy	83.19	4.82	11.99
Previous Method [2]	70.83	5.07	24.10

From Table 6, it is observed that the new approach has achieved a higher correct segmentation rate and lower error rate than the previous method under the same testing conditions.

8   Concluding Remarks

This paper proposes a method of automatically segmenting the date information handwritten on bank cheques, together with an improvement of this method. The improvement depends on using contextual information provided by a connected digit recognizer, and it has been found to be effective. We will continue to further improve our work in this area through the incorporation of contextual information and results from the recognizer(s).

Acknowledgements

References

1. E. Cohen, J.J. Hull, and S. N. Srihari. Control structure for interpreting handwritten addresses. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(10):1049–1055, Oct 1994.
2. R. Fan, L. Lam, and C. Y. Suen. Processing of date information on cheques. *Progress in Handwriting Recognition*, pages 473–479, eds. A. C. Downton and C. Impedovo, World Scientific, 1997.
3. D. Guillevic and C.Y. Suen. Recognition of legal amounts on bank cheques. *Pattern Analysis and Applications*, 1(1):28–41, 1998.
4. G.F. Houle, D.B. Aragon, R.W. Smith, M. Shridhar, and D. Kimura. A multi-layered corroboration-based check reader. In *Proc. IAPR Workshop on Document Analysis Systems*, pages 495–546, Malvern, Penn. USA, Oct. 1996.
5. U. Mahadevan and R. C. Nagabushnam. Gap metrics for word separation in handwritten lines. In *Proc. of Third Int. Conf. on Document Analysis and Recognition*, pages 124–127, Montréal, Canada, Aug. 1995.
6. G. Seni and E. Cohen. External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 27(1):41–52, 1994.
7. N. W. Strathy. Handwriting recognition for cheque processing. In *Proc. 2nd Int. Conf. on Multimodal Interface*, pages III–47–50, Hong Kong, Jan. 1999.