Field Extraction for Form Documents

James You (游瑛準), Chen-Chiung Hsieh(謝禎冏),
H-H Chao (趙象華), and Wen-Tsung Chang(張文村)
Special System Division, Institute for Information Industry,
Taipei, Taiwan, Republic of China

Abstract

Since form documents play a very important role in various applications, form interpretation is always a good topic in image processing. In this paper, a method for extracting field features of a form document is presented. This method is immune to slant forms with skew angle being less than 45 degrees. It can be divided into the following steps: image scanning, connected-component generation, contour tracing, corner extraction, line generation, corner grouping, and line alignment. Experimental results show that this method produces good results in fast running speed.

1. Introduction

Paper documents have been serving as the main media for storing and transmitting information in various applications for a long time, such as business letters, journals, technical manuals, table forms, engineering drawings, etc. Traditional methods for paper documents processing are carried out mainly by human beings. Such schemes cost too much time and money. Therefore, an automatic form-reading system is a very good research topic in image processing. Many techniques[1-10] are proposed on this topic.

Since form documents are used widely in the modern world, recognition of form documents is a very important subject from the viewpoint of practical applications. Several form-processing systems[11-20] have been proposed so far. Traditionally, the processing flow of a form reader is as follows: image scanning, feature extraction, form classification, data extraction, recognition, human validation, and database management.

In fact, a good form reader always needs a powerful mechanism to extract field features from form documents. The proposed method extracts field-features accurately with remarkable running speed even when documents slant very seriously that is a fatal problem for conventional form-reading systems.

This paper is organized as follows. Section two describes the scheme we adopted for extracting field-features in a form document. Section three shows the experimental results. Section four gives a conclusion and the future work.

2. Field extraction

Our method contains the following steps: (1) image scanning, (2) connected-component generation, (3) tables/graphics/text classification, (4) contour tracing, (5) corner detection, (6) deskew, (7) line generation, (8) corner grouping, and (9) line alignment.

2.1 Image scanning

Form documents are scanned and converted to raster-based images. Some pre-processing are provided to obtain better quality of the input images.

2.2 Connected-component generation

In order to extract information from a form document, connected components[21] are generated at first. There are many types of connected components, such as 4-connected, 8-connected, and m-connected[23]. In this paper, 8-connected is adopted.

Let p denotes the current-examining pixel, p_1, p_2, … and p_8 represent the eight neighboring pixels of p as shown in Fig. 1. The scanning sequence, from top to bottom and left to right, ensures that p_4, p_6, p_7, and p_8 must have already been processed before p is reached. The algorithm adopted in this paper is described as follows:

```
if ( $p$ is a background pixel )
    Move to the next scanning position.
else if ( $p$ is a foreground pixel )
{
    if ( none of $p_4$, $p_6$, $p_7$, and $p_8$ is foreground )
        Assign a new label for $p$.
    if ( only one of $p_4$, $p_6$, $p_7$ and $p_8$ is foreground )
        Assign the label of the foreground neighbor to $p$.
    if ( two or more of $p_4$, $p_6$, $p_7$, and $p_8$ are foreground )
    {
        Assign one of the foreground neighbor's label to $p$.
        Make a note that the foreground neighbors have the
        same label.
    }
}
```

This algorithm is implemented with linked list to register the connected components that are determined.

2.3 Tables/graphics/text classification

There are many objects, including text, table, isolated lines, noise, etc., in a form document. The generated connected-components should be classified for further processing. Two predefined thresholds, width-threshold(*WT*) and height-threshold(*HT*), are used for connected-component classification. The algorithm is described in table 1.

To discriminate tables from images, another predefined threshold, black pixel density (*BPD*), is adopted. A connected component is an image if its black pixel density is larger than *BPD*, or it is a table.

2.4 Contour tracing

A table contains an outer frame and several inner frames. In order to obtain the information of a table, contour of each field is traced. Each tracing result corresponds to a field as shown in Fig.2.

The proposed contour-tracing algorithm, which can be divided into 3 procedures: get a starting point for contour tracing (*GetTraceSP*), trace the outer frame (*TraceOuterFrame*), and trace inner frames (*TraceInnerFrame*), is described as follows:

```
SP = A starting point for contour tracing.
While( ( SP = GetTraceSP() != NULL )
{
    If ( SP is the first starting point of current table )
        Call TraceOuterFrame( SP ). //Trace outer frame.
    Else
        Call TraceInnerFrame( SP ).//Trace inner frame.
}
```

Procedure GetTraceSP scan the region of current examining connected-component, represented by *rectangle(Left, Top, Right, Bottom)*, from *bottom* to *top* and *left* to *right*. It returns a pixel as a starting point for tracing if it meets the following conditions.

(1) It is a foreground pixel.

(2) It has not been traversed.

(3) It has at least one 4-connected neighbor which is background.

TraceOuterFrame() and *TraceInnerFrame()* are conventional contour-tracing algorithms which can be found in [22]. The obtained contour points are recorded by linked lists to generate corners of a field. If the number of contour points of a field is less than a predefined threshold (for example, 40 pixels), that tracing result will be considered as a noise and then removed.

2.5 Corner generation

Four corners will be generated for each field by using the list of its contour points. The proposed corner-detection algorithm is described as follows:

(1) For each pair of pixels (p_i, p_j) in the list, calculate the distance of (p_i, p_j).

(2) Find a pair (p_k, p_m) with the largest distance. Set p_k and p_m as the two diagonal corners of current examining field.

(3) For any pixel p in the list, $p \neq p_k$ and $p \neq p_m$, calculate the projective distance from p to the diagonal (p_k,p_m). For each side of the diagonal (p_k,p_m), find a pixel with the largest projective distance. The found two pixels, p_s and p_t , are the other two corners in this field. An example is shown in Fig. 3.

The algorithm described above is imperfect for the computational complexity being too large. In order to find (p_k, p_m), calculating the distance of each possible pairs (p_i, p_j) in a field is ineluctable which makes the computational complexity be $O(n^2)$. For a rectangle with 5,000 contour pixels, which can be easily achieved by a high resolution image-scanner, needs 25,000,000 computations to find p_k and p_m.

To solve this problem, we propose the idea of finding corner-candidates in a field before applying the corner-detection algorithm. Since only the contour points with their curvature close to 90 degrees are possible to be the corners of a field. We select the contour points with their curvature close to 90 degrees as corner candidates. Now, the corner-detection algorithm take the corner candidates as input instead of all contour pixels. Because the number of input data is much smaller, the computational complexity is greatly reduced.

2.6 Deskew

After corners are generated for each field, line information is available as shown in Fig. 4. In order to get the skew angle and deskew, the following procedures are applied.

(1) Find the longest field line, L_m. Assume the starting and ending points of L_m are (x_1, y_1) and (x_2, y_2).
(2) Skew angle of this document is then easily obtianed by $tan^{-1}(y_2-y_1/x_2-x_1)$.
(3) Deskew all corners found in the previous section.

If the skew angle of a form document is larger than 45 degrees, for example, 55 degrees as shown in Fig. 5. It is impossible to know the accurate skew angle of a form document. Such ambiguity can not be solved without the help from a OCR system. Therefore, in this paper, the skew angle of a form document must be restricted to no more than 45 degrees.

2.7 Line generation

For each field, the following three steps are applied:
(1) Generate four field-lines by the deskewed corners
(2) Calculate the width of each field-line. Width of a field line can be easily obtained by calculating the distance between two lines as shown in Fig. 6.
(3) Correct the starting and ending points of each line by line-width information as shown in Fig. 7.

2.8 Corner grouping

Neighboring corners should be grouped into one because they are actually the same point. An example is shown in Fig. 8. The grouping algorithm contains two steps:
(1) Define a threshold T.
(2) Two corners, p_i and p_j in a table, will be grouped into a single pixel if $distance(p_i, p_j) < T$.

2.9 Line alignment

Near-colinear lines of different fields will be aligned into colinear as shown in Fig 9. This algorithm contains two steps:

(1) Define a threshold F.
(2) Two field lines, L_1 and L_2, will be aligned into colinear if $distance(L_1, L_2) < F$.

3. Experimental results

Some experimental results are given in this section. The platform of our system is IBM compatible PC, Pentium-120 CPU, running on Microsoft Windows-95. For two testing samples(Fig. 10 and Fig. 12), with resolutions being 2,352 × 1,480 and 2,925 × 2,490, it takes 6.3 and 10.5 seconds to generate all the connected components and field-features as shown in Fig. 11 and Fig. 13.

4. Conclusion and future work

In this paper, a simple but efficient algorithm for extracting field-features of form documents is presented. It contains nine steps: (1) image scanning, (2) connected-component generation, (3) tables/graphics/text classification, (4) contour tracing, (5) corner generation, (6) deskew, (7) line generation, (8) corner grouping, and (9) line alignment. Steps (4) and (5) are rotation invariant. Therefore, the proposed method is immune to a slant document with the skew angle being less than 45 degrees. Also the running speed is remarkable.

The other form-processing procedures, such as text block generation, character grouping, writing direction detection, and character recognition, segmentation of touching characters ...etc, are still under developing in our team.

References

[1] T. Watanabe, Q. Luo, and N. Sugie, "A cooperative document understanding method among multiple recognition procedures," Proc. 11th ICPR, pp. 689-692, 1992.

[2] D. Niyogi and S. Srihari, "A rule-based system for document understanding," Proc. AAAI 86, pp.789-793.

[3] F. Esposito, D. Malerba, G. Semeraro, E. Annese, and G. Scafaro, "An experimental page layout recognition system for office document automatic classification: An integrated approach for inductive generalization," Proc. 10 th ICPR, pp. 557-562, 1990.

[4] J.L. Fisher, S.C. Hinds and D.P. D'amato, "A rule-based system for document image segmentation," Proc. 10th ICPR, pp.567-572, 1990.

[5] J. Higashino, H. Fujisawa, Y. Nakano, and M. Ejiri, "A knowledge based segmentation method for document understanding," Proc. 8th ICPR, pp. 745-748, 1986.

[6] K. Kise, K.Momota, M. Yanaka, J. Sugiyama, N. Babaguchi, and Y. Tezuka, "Model based understanding of document images," Proc. MVA 90, pp.471-474.

[7] A. Dengel and G. Barth, "High level document analysis guided by geometric aspects," Int'l J. Pattern Recognition and Artificial Intelligence, vol. 2, no.4, pp. 641-655, 1988.

[8] Q. Luo, T. Watanabe, Y. Yoshida, and Y. Inagaki, "Recognition of document structure on the basis of

spatial and geometric relationships between document items," Proc. MVA 90, pp. 461-464.

[9] T. Watanabe, Q. Luo, and T.Fukumura, "A framework of layout recognition of document understand," Proc. 1st Symp. Document Analysis and Information Retrieval, pp. 777-95, 1992.

[10] T. Watanabe, Q. Luo, and N. Sugie, "Structure recognition methods for various types of documents," Int' J. Machine Vision and Application, 6, pp. 163-176, 1993.

[11] Lai, C. P. and R. Kasturi, "Detection of dashed lines in engineering drawings and maps," in Proc. 1th Int. Conf. Document Anal. Recognition, Saint Malo, France, Sep. 30-Oct.2, 1991, pp.507-515.

[12] Wang, K. Y., R. G. Casey, and F. M. Wahl, "Document analysis system," IBM J. Res. Develop., vol. 26, no.6, pp.647-656, 1982.

[13] Watanabe, T., Q. Liu, and N. Sugie, "Layout recognition of multi-kinds of table-form documents," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 17, no.4, pp.432-445, 1995.

[14] Watanabe, T., H. Naruse, Q. Luo, and N. Sugie, "Structure analysis of table-form document on the basis of the recognition of vertical and horizontal lines segments," in Proc. 1th Int. Conf. Document Anal. Recognition, Saint Malo, France, Sep.30-Oct.2, 1991, pp.638-646.

[15] Wang, D. and S. N. Srihari, "Analysis of form images," in Proc 1st Int. Conf. Document Anal. Recognition, Saint Malo, France, Sep.30-Oct.2, 1991, pp.181-191.

[16] Tayor, S. L., R. Fritzson, and J. A. Pastor, "Extraction of data from preprinted forms," Machine Vision and Applications, vol.5, no.3, pp.211-222, 1992.

[17] Pizano, A., "Extracting line feature from images of business forms and tables." In Proc. 11th Int. Conf. On Pattern Recognition, The Hague, The Netherlands, Aug.30-Sep.3, 1992, pp.399-403.

[18] Yan, C. D., Y. Y. Tang and C. Y. Suen, "Form understanding system based on form description language," in Proc. 1th Int. Conf. Document Anal. Recognition, Saint Malo, France, Sep.30-Oct 2, 1991, pp.283-293.

[19] Casey, R. G. and D. R. Ferguson, "Intelligent forms processing, "IBM Systems Journal, vol.29, no.3, pp.435-450, 1990.

[20] Casey, R. G., D. R. Ferguson, K. Mohiuddin, and E. Walach, "Intelligent forms processing system," Machine Vision and Applications, vol.5, pp.143-155, 1992.

[21] Fletcher, L. A. and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," IEEEE Trans. Pattern Analysis and Machine Intelligence, vol.10, no.6, pp.910-918, 1988.

[22] Rosenfeld, A., and Kak, A.C. [1982]. Digital Picture Processing, 2nd ed., vol. 1.

[23] Academic Press, New York.Gonzalez and Woods, Digital Image Processing, 2nd ed, Addison-Wesley Publishing Company, Inc. 1992.

Table 1. *CCW*: Width of a connected component.
CCH: Height of a connected component.

	CCW>WT	*CCW* ≦ *WT*
CCH>HT	tables or images	isolated horizontal lines
CCW ≦ *HT*	isolated vertical lines	text

p_6	p_7	p_8
p_4	p	p_5
p_1	p_2	p_3

Fig. 1. *p* and its eight neighbors.

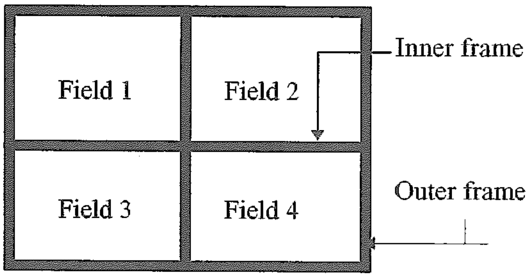


Fig. 2. The outer frame and inner frames in a table.

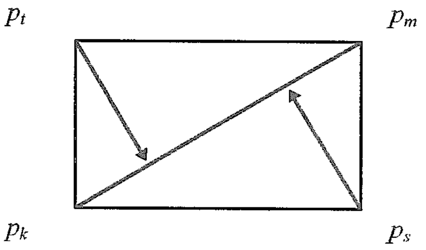


Fig. 3. A field with 4 corners: p_k, p_m, p_s, and p_t.

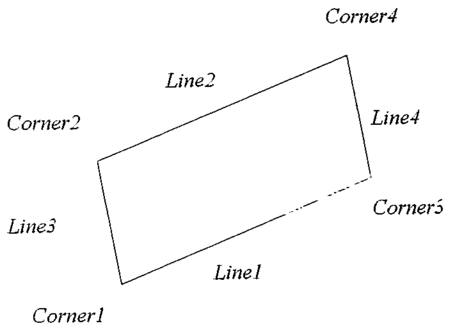


Fig. 4. Lines and corners in a field.

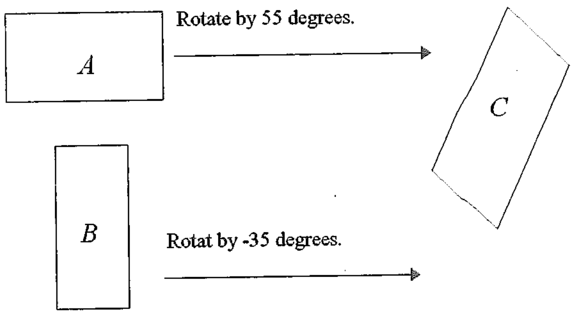


Fig. 5. It is impossible to identify *C* is rotated from *A* or *B*.

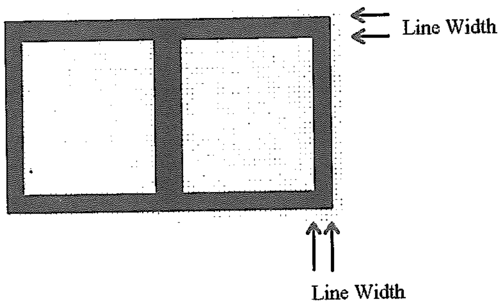


Fig. 6. Line width calculation.

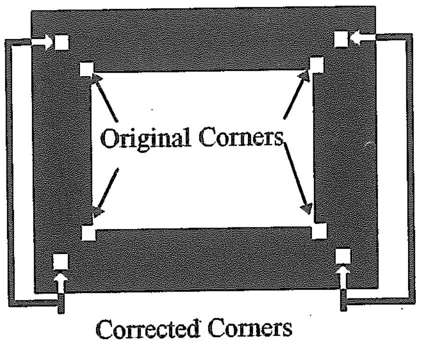


Fig. 7. Correct corners by line width.

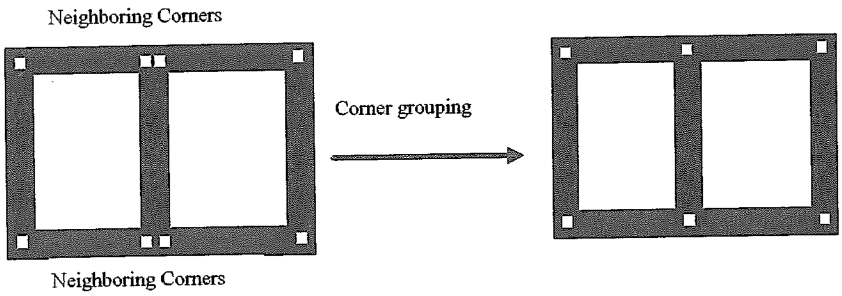


Fig. 8. An example of corner grouping.

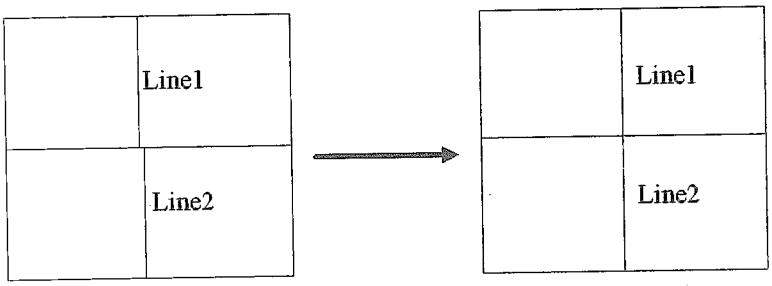
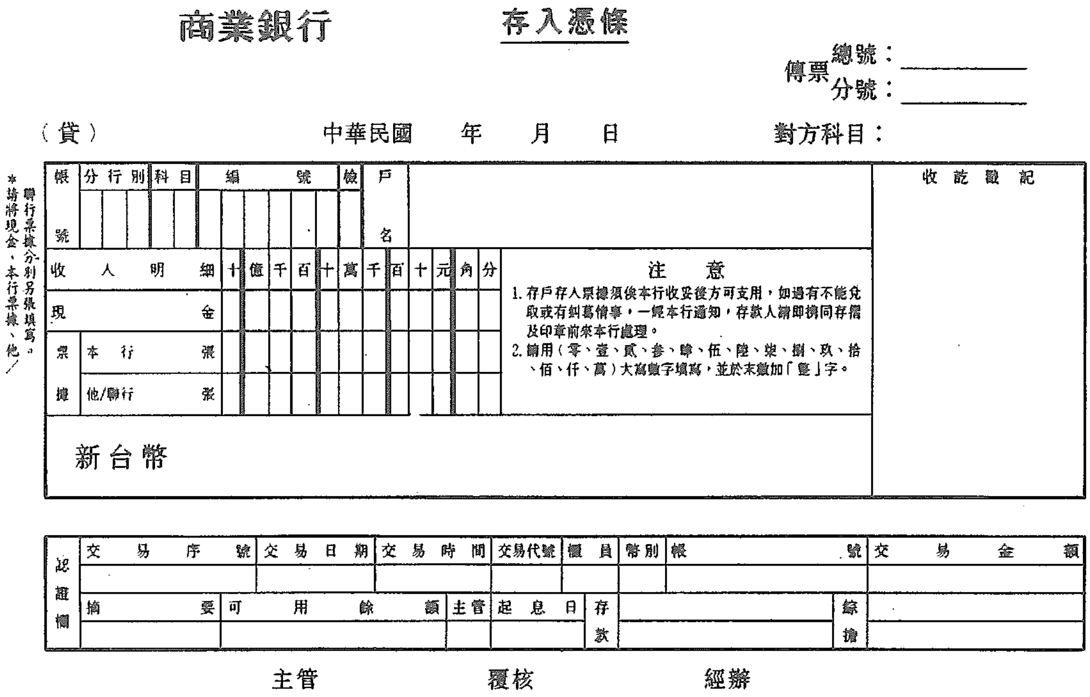


Fig. 9. Alignment for near-colinear field lines.



商業銀行　　存入憑條

傳票 總號：＿＿＿＿
傳票 分號：＿＿＿＿

（貸）　　中華民國　年　月　日　　對方科目：

*請將現金、本行票據、他/聯行票據分別另張填寫。

帳號	分行別	科目	編號	檢	戶名		收訖戳記

收入明細	十	億	千	百	十	萬	千	百	十	元	角	分	注意 1. 存戶存入票據須俟本行收妥後方可支用，如遇有不能兌取或有糾葛情事，一經本行通知，存款人請即攜同存摺及印章前來本行處理。 2. 請用（零、壹、貳、參、肆、伍、陸、柒、捌、玖、拾、佰、仟、萬）大寫數字填寫，並於末尾加「整」字。
現金													
票據 本行 張													
票據 他/聯行 張													

新台幣

認證欄	交易序號	交易日期	交易時間	交易代號	櫃員	幣別	帳號	交易金額
	摘要	可用餘額	主管	起息日	存款		綜擔	

主管　　覆核　　經辦

Fig. 10. A testing example.

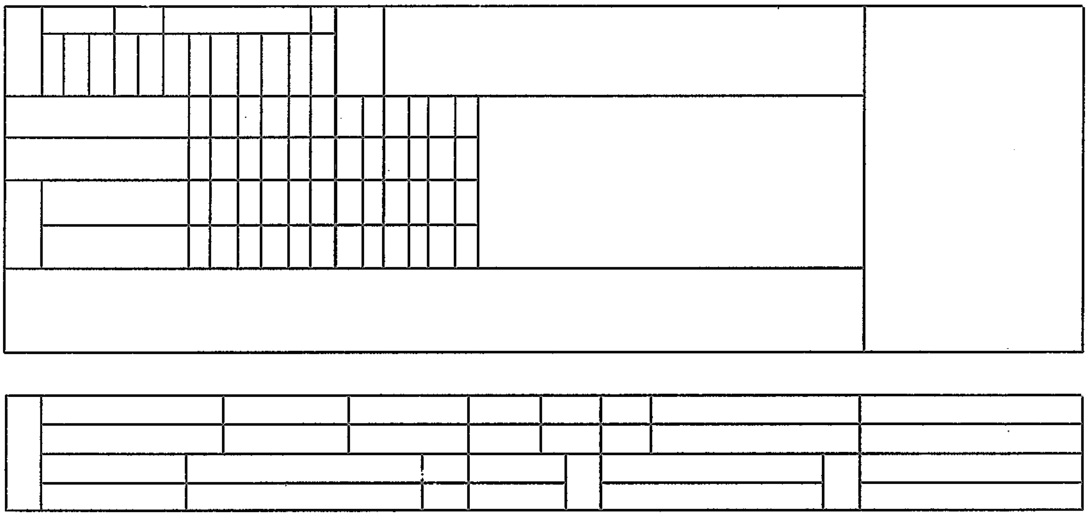
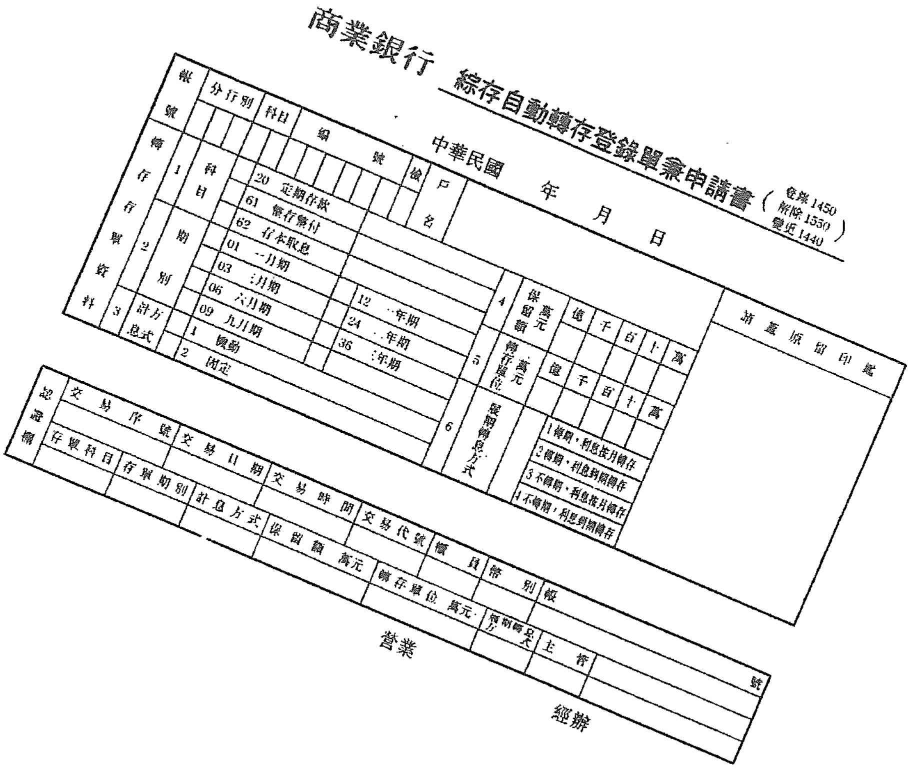


Fig. 11. The obtained result.



商業銀行 綜存自動轉存登錄單彙申請書（電傳1450 解除1550 變更1440）

中華民國 年 月 日

帳號 分行別 科目 編號 檢 戶名

轉存存單資料

1 科目 20 定期存款 61 整存整付 62 存本取息

2 期別 01 一月期 03 三月期 06 六月期 09 九月期 12 一年期 24 二年期 36 三年期

3 計息方式 1 機動 2 固定

4 保留額 萬元 億 千 百 十 萬

5 轉存單位 萬元 億 千 百 十 萬

6 到期轉息方式 1轉期，利息按月轉存 2轉期，利息到期轉存 3不轉期，利息按月轉存 4不轉期，利息到期轉存

請蓋原留印鑑

記錄欄 交易序號 交易日期 交易時間 交易代號 櫃員 帳別 帳號

存單科目 存單期別 計息方式 保留額萬元 轉存單位萬元 到期轉息方式 主管

營業 經辦

Fig. 12. Another testing example.

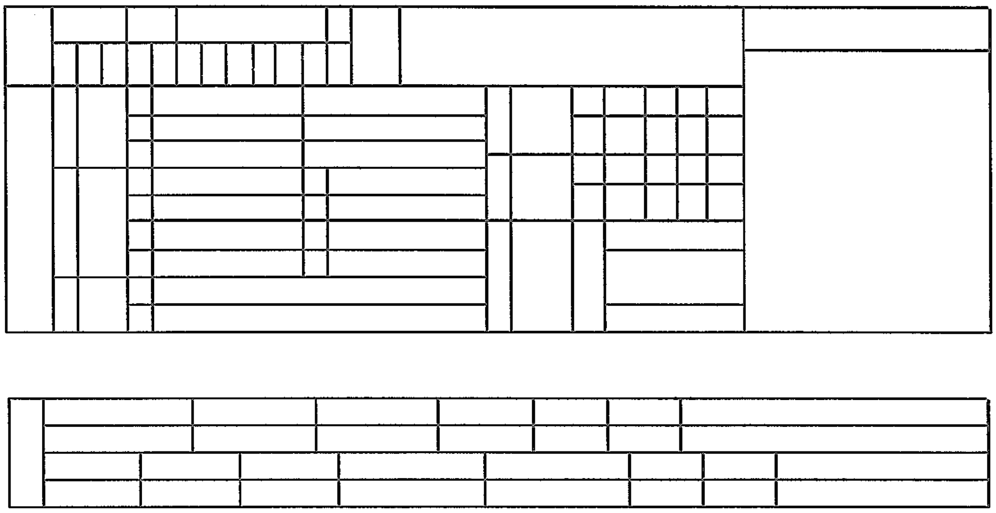


Fig. 13. The obtained result.