Automatic Detection of Change in Address Blocks for Reply Forms Processing

K R Karthick, S Marshall and A J Gray

***Abstract*— In this paper, an automatic method to detect the presence of on-line erasures/scribbles/corrections/over-writing in the address block of various types of subscription and utility payment forms is presented. The proposed approach employs bottom-up segmentation of the address block. Heuristic rules based on structural features are used to automate the detection process. The algorithm is applied on a large dataset of 5,780 real world document forms of 200 dots per inch resolution. The proposed algorithm performs well, with a detection accuracy of 98.96% and an average processing time of 108 milliseconds per document.**

***Index Terms*— Address correction detection, Document form processing, Document image analysis, Off-line scribble detection**

I. Introduction

Automatic document processing, which includes processing of cheques, tax forms, ballot papers, examination answer sheets, newspaper subscription payment forms, postal mails etc, has gained momentum in the last two decades among various sectors of industry. As these documents are processed in huge quantities daily, automation will bring enormous advantage to the industry. Sectors such as banking, utility companies, postal services and newspaper companies use automatic form processing to minimize processing cost, increase efficiency, reduce processing time and minimize manual intervention. In this paper, a system is presented that will automate the detection of the presence of handwritten changes in address blocks of reply forms. The reply form refers to newspaper subscription or utility billing forms.

II. Background

Service providers such as media and utility companies process millions of subscription and billing forms every day. The workflow process between the companies and their customers is shown in Fig 1. The companies send bills to their customers, enclosing the payment form. The completed payment forms are sent back to the company by the customer and are processed. For example, in the case of a newspaper subscription payment form, the information of interest will be customer address, subscription type and payment details. A sample subscription payment form is given in Fig 2. This information extraction is a vital step for the companies to offer a better service to their customers.

A combination of form pre-processing and OCR applications automatically extracts the payment details from the forms. So far this has required manual sifting of all the forms to detect any corrections in the customer name and address details. The algorithm presented will do away with this laborious and time-consuming process. It automates the address correction detection process, thereby eliminating the need for manual sifting. Hence only the address blocks with scribbles will be sent to the operator for updating the customer records.

The paper is organized as follows: the nature of the changes to addresses is explained in section III. Related work from the literature is described in section IV. Section V describes the concept behind the proposed algorithm and a detailed explanation of its workings. Experimental results and their analysis are presented in section VI. Section VII discusses some issues of the approach, followed by the conclusion in section VIII.

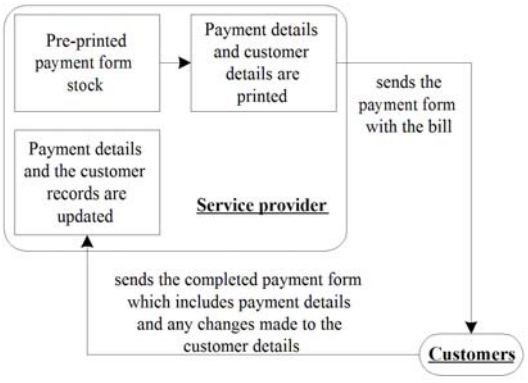

Fig 1. Workflow between the company and their customers

III. Nature of the Off-line Scribbles

Customers tend to correct their address details in an irregular manner. The algorithm categorizes these corrections and any other changes in the address block as scribbles. The following types of correction are observed from the analysis of scribbles:

- Horizontal striking of one or many word/s or entire line/s

The authors would like to thank the industrial partners, Aperta Ltd, for supplying the sample images. This work is sponsored under the United Kingdom's Knowledge Transfer Partnership (KTP) program.

K.R. Karthick is with the Department of Electronic and Electrical Engineering, University of Strathclyde, 204 George Street, Glasgow, United Kingdom, G1 1XW (phone: +44 (0) 141 548 2686; fax: +44 (0) 141 552 2487; e-mail: krk@eee.strath.ac.uk).

S. Marshall is with the Department of Electronic and Electrical Engineering, University of Strathclyde, 204 George Street, Glasgow, United Kingdom, G1 1XW (e-mail: s.marshall@eee.strath.ac.uk).

A.J. Gray is with the Department of Statistics and Modelling Science, University of Strathclyde, 26 Richmond Street, Glasgow, United Kingdom, G1 1XH (e-mail: alison@stams.strath.ac.uk).

- Multiple horizontal striking of word/s or entire line

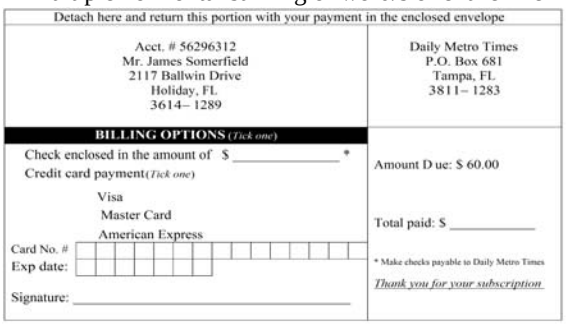

Fig 2. Sample billing form

- Irregular striking of word/s
- Crosses for scoring out blocks of texts
- Drawing irregular lines and arrows pointing to customer entered address
- Circling the address block
- Writing adjacent to the horizontal lines of words which have been struck out
- Blocks of text written on either side of the address block
- Blocks of text written on either the top or bottom of the address block
- Drawing arbitrary shapes over the area of correction
- Customer prints address seals to update addresses and sticks these on.

Fig 3 illustrates the variability of the nature of the scribbles by showing examples of the various ways in which different customers correct the same address block.

IV. Related work

Erasure/scribble detection can be classified into on-line and off-line approaches. In the on-line approach, devices such as LCD digitizing tablets and stylus capture the pen movements, and the dynamic information in the strokes is saved for processing. In the off-line approach, the paper documents are scanned and only the pixel information in the image is available for processing. Wiart [9] proposed an on-line approach for detecting and recognizing erasures in on-line captured forms. A preprocessing step eliminates the isolated strokes. A multi-layer perceptron (MLP) along with a user-defined threshold is used to classify the preprocessed strokes into erasures and non-erasures.

A structural layout analysis is a vital step in the process of feature extraction. The literature indicates that structural layout analysis can be performed by means of a top-down approach, bottom-up approach, or a hybrid approach.

The top-down approach [1, 3] starts from a whole document, and results in smaller sub-components such as one or more columns of blocks of text, paragraph blocks, text lines, and individual characters. The bottom-up approach [2, 4] begins with individual pixels, grows into run-lengths, and then merges to give connected components. Connected components are further merged into words, then into lines, paragraphs etc. The hybrid approach [8] combines the functionalities of both the approaches to segmentation.

A real time off-line algorithm using a bottom-up segmentation approach, combined with statistics, to automatically detect user-corrected address blocks is proposed. In this method, connected components based on pixel run-lengths are implemented utilizing graph techniques and the standard template class library of C++, to achieve real time processing.

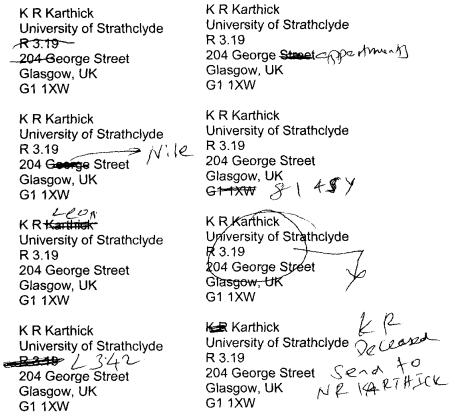

Fig 3. Different types of scribbles on a sample address block

V. Proposed approach

A. Concept

The algorithm presented in this paper is based on prior knowledge of the problem domain using heuristic rules. The address block is localized from prior knowledge of the document layout. A machine printed address block will include distinct characters. As the address block is expected to contain only printed text, the inter-line spacing will be uniform. The inter-character spacing will be equal and the characters will show regularity in alignment, and some similarity in physical features such as width, height and aspect ratio. Any unconstrained handwriting/corrections/scribbles by the customer on to the address block will affect the features of the characters and the statistical quantities associated with alignment of the lines within the address block. This concept is exploited to determine the presence of scribbles within the address block. Though the above-mentioned properties are well known, we interpret them in a unique way so that the heuristic rules are applicable to different types of fonts. The flow diagram in Fig 4 shows an overview of the algorithm.

B. Algorithm description

Step 1: The input image is a binary image. The address block location is obtained from prior knowledge of the layout of the document. The address localization can also be automated, based on the application domain. Palumbo [6] proposed a real time algorithm for postal address block location in postal letters. The address block is not always printed at the same spatial position in the document, but suffers translation along the vertical and horizontal directions. This is because the customer details are printed on a pre-printed payment form stock.

The extent of translation for a random subset of twenty-six documents was analyzed. The average and standard deviation of the left spatial positions in this subset are 186.26 and 14.66

pixels respectively. The average and standard deviation of the top spatial positions in this subset are 97.07 and 7.83 pixels respectively. Care needs to be taken to include enough spatial area to include the customer scribbles on all sides of the address block.

Step 2: The foreground pixels inside the address block are mapped into run-lengths. A graph data structure is constructed with the run-lengths as the vertices.

Step 3: An adjacency list [7] of all the runs is constructed. A graph data structure based search technique [7] is applied to derive the spatial relationship between connected run-lengths. This implementation is carried out using the standard template class library of Borland C++. The connected components obtained are eight-connected.

Step 4: The bounding boxes of the connected components are calculated. The bounding box is the smallest box enclosing the connected component. From the bounding box, the following four features are calculated and represented in a data structure: height, width, area and aspect ratio.

Step 5: The connected components are screened initially to eliminate false candidates such as noise, pre-printed horizontal lines, and pre-printed vertical lines. Aspect ratio, height, width and area of connected components are the features utilized for this screening.

Heuristic rule for finding an over-height component: Components with height more than twice the average height are flagged as scribbles.

Heuristic rule for finding an over-width component: Components with width more than three times the average width are chosen as the prospective candidates for scribbles. Due to the poor quality of printing, adjacent characters might overlap and form false candidates for over-width component scribbles. In order to differentiate between touching characters and components connected by scribbles, a horizontal profile of the component is calculated. A pattern was observed in which most touching characters connect either on the top or bottom of the components, whereas the user scribble occurs irregularly in the middle of the component. An experimental threshold on the top and bottom of pre-selected pixel rows of the candidate component to eliminate the touching characters and to spot the scribbled components was identified.

The average values of height and width can be calculated from a histogram of the size of the address block or can be set as a heuristic threshold.

Step 6: Rows of characters are segmented into individual lines based on the top and bottom spatial coordinate values of the bounding boxes. The rule of thumb for this operation is that if the top spatial position of the previous component is less than or equal to the bottom spatial position of the current component then a new line is found.

Step 7: The means and standard deviations of the bottom spatial row coordinates of the components within each segmented line are calculated. This statistical measure is used to extract the irregularity, caused by the outliers, of the user-corrected address. If the calculated standard deviation is above an experimental standard deviation value, then this confirms the presence of a user-corrected address.

Step 8: A log file gives the file names that contain scribbles.

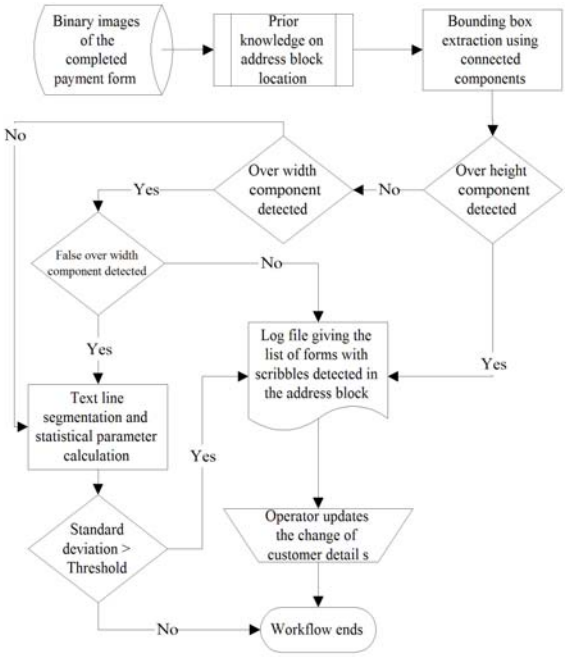

Fig 4. Flow diagram of the proposed algorithm

VI. Experiments and Results

The experimental sample consists of five different types of 5,780 real world documents, Type A through Type E. A section of text is shown in Fig 5 from each type of document. The sample set contains 149 address blocks with scribbles and 5631 without scribble. The address block contains different types of fonts, a varying number of address lines, one-dimensional barcodes and in some cases, scribbles. The documents were scanned at a resolution of 200 dpi, under different industrial scanner settings.

The test machine runs at 1.86 GHz speed in Windows XP with an Intel(R) Pentium(R) M processor. The processing time ranges from 0.062 seconds to 0.391 seconds for one document. The average time taken to process one document is 0.108 seconds. The processing time taken by the function (excludes reading in images into the buffer), which performs the detection process for one document, ranges from 0.015 seconds to 0.266 seconds. The average time taken by the function that performs the detection process for one document is 0.043 seconds.

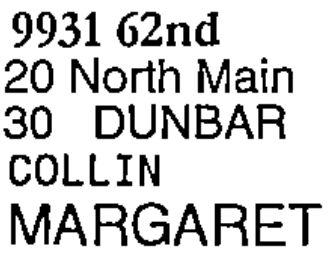

Fig 5. Document sample type

For any application, the success depends not only on achieving the desired outcomes (scribble detection in our case) but also it must be robust enough to avoid too many false positives. The algorithm's performance is summarized using four quantities. They are:

- True positives (TP) – Documents with scribbled address block, which are detected.
- True negatives (TN) – Documents with no scribbled address block and not detected.
- False positives (FP) – Documents with no scribbles in the address block but detected
- False negatives (FN) - Documents with scribbled address block but not detected.

A confusion matrix helps us to calculate the true positive rate (TPR) or sensitivity or recall, and the false positive rate (FPR). The confusion matrix is given in Table I.

Table I. Confusion matrix of the classification results

With scribble 149	Without scribble 5631
TP: 139	FP: 50
FN: 10	TN: 5581

TPR, FPR, precision and accuracy act as measures of classification performance. TPR is the ratio of TP and the sum of TP and FN. FPR is the ratio of FP and the sum of FP and TN. Precision or positive predictive value is the ratio of TP and the sum of TP and FP. Accuracy is the ratio of the sum of TP and TN and the size of the sample set. Table II gives the values of TPR, FPR, precision and accuracy.

Table II. System classification performance

True positive rate	93.29%
False positive rate	0.89%
Precision	73.54%
Accuracy	98.96%

The effect of the three thresholds on performance can be analyzed in a standard ROC curve by varying them one at a time, as shown in Fig 6. These curves result from varying one at a time the three thresholds on which the algorithm depends, while keeping the other two thresholds fixed at their empirically determined value. This allows an ideal point of operation to be selected in terms of each threshold separately. An alternative would be to vary all three thresholds simultaneously over a fine grid, to generate a multi-dimensional surface for each measure of classification accuracy, however it is then more difficult to establish an optimal trade-off between TPR and FPR. In the ROC curves, the ideal point of classification (0,1) is the perfect classification. Our classification point using the three empirically chosen thresholds is (0.008, 0.93) which lies very close to the ideal point. This performance is also reflected in the accuracy value of 98.96%.

On-line approaches often exhibit improved performance compared to their off-line counterparts [5] due to the presence of dynamic information. We compare our off-line approach with the on-line method of Wiart [9], even in the absence of temporal information to demonstrate the superior performance of our system. Wiart's approach achieves a highest TPR of around 97% with a precision of around 60% and a FPR of 15% accompanied by a FNR of 2.5%. We achieve a TPR of 93.29% with a precision of 73.54% and a FPR as low as 0.89% with a FNR of 0.17%. Our approach shows superior performance in terms of precision, FPR and FNR.

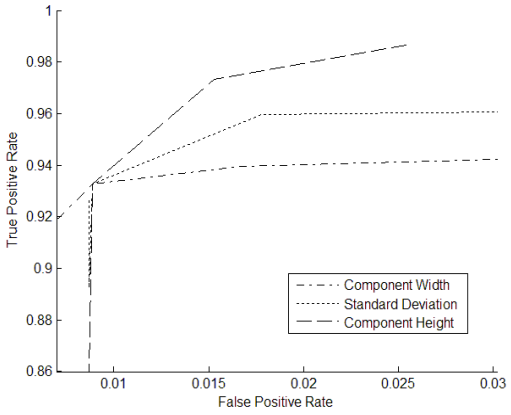

Fig 6. ROC curves showing the performance

VII. Discussion

Due to poor quality of the printing and the choice of fonts, characters can overlap along the horizontal direction, resulting in over-width components. Though the approach based on horizontal profile projection differentiates between the touching characters and the scribble-linked component, this can still be a cause of many false positives. Three such sample images are given in Fig 7.

Fig 7. Characters touching horizontally

Minimal spacing between adjacent lines causes character overlapping in the vertical direction, resulting in false positives. This shows that an optimum spacing between address lines is required for a good performance. Two such sample images are shown in Fig 8.

Fig 8. Characters touching vertically

A correction within a single component does not cause any significant changes in the bounding box features and on the standard deviation values. The algorithm fails to spot such single character corrections. Three such sample images are given in Fig 9.

Fig 9. Single corrected character

VIII. Conclusion

In this paper, a real time off-line algorithm to detect the presence of scribbles in address block of forms has been

presented. The algorithm was tested on five different types of document forms on an industrial scale and the results support the robustness of the heuristic rules. It currently relies on three empirically determined thresholds. We are carrying out further work to make the system independent of these thresholds by using classifiers, while maintaining the excellent system performance.

REFERENCES

[1] T. Akiyama and N. Hagita, "Automated Entry System for Printed Documents", Pattern Recognition, Vol. 23, 1990, pp. 1141-1154.

[2] D. Drivas and A. Amin, "Page Segmentation and Classification utilising a Bottom-Up Approach", Proceedings of the Third International Conference on Document Analysis and Recognition, Vol. 2, 1995, pp. 610-614.

[3] J. Ha, R. M. Harlick and I. T. Philips, "Recursive X-Y Cut using Bounding Boxes of Connected Components", Proceedings of the Third International Conference on Document Analysis and Recognition, Vol. 2, 1995, pp. 952-955.

[4] A.K. Jain and B. Yu, "Document Representation and its Application to Page Decomposition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, 1998, pp. 294-308.

[5] R. Plamondon, and S. N. Srihari, “On-line and Off-line Handwriting Recognition: A Comprehensive Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22 (1), 2000, pp. 63-84.

[6] P. W. Palumbo, S. N. Srihari, J. Soh, R. Sridhar and V. Demjanenko, "Postal Address Block Location in Real Time", Computer, Vol. 25, 1992, pp. 34-42.

[7] Sedgewick R., “Algorithms in C, Part 5: Graph Algorithms”, Addison-Wesley Professional, 2001.

[8] D. Wang and S. N. Srihari, "Classification of Newspaper Image Blocks using Texture Analysis", Computer Vision, Graphics, and Image Processing, Vol. 47, 1989, pp. 327-352.

[9] A. Wiart, T. Paquet, L. Heutte, “Detection and Recognition of Erasures in On-line Captured Paper Forms”, Pattern Recognition Letters, Vol. 28, 2007, pp.1263-1270.