Journal of Computer Science 7 (12): 1778-1783, 2011
ISSN 1549-3636


# Handwritten Characters Extraction from Form Based on Line Shape Characteristics

[1]Ali Qusay Al-Faris, [2]Dzulkifli Mohamad,
[1]Umi Kalthum Ngah and [1]Nor Ashidi Mat Isa
[1,] School of Electrical and Electronic Engineering, Universiti Sains Malaysia,
Engineering Campus, Seri Ampangan 14300 Nibong Tebal, Penang, Malaysia
[2]Department of Computer Graphics and Multimedia,
Faculty of Computer Science and Information System,
Universiti Teknologi Malaysia, 81300 Skudai, Johor, Malaysia

**Abstract: Problem statement:** Data entry form is a convenient and successful tool for information collection by filling in the sheets using pen and handwriting. One of the most important fields in these forms is the data filled boxes. Extracting the handwriting from the data entry forms is important for many purposes such as in documenting and archiving. The extraction process is also important in situations such as when it is necessary to the handwritten recognition process. **Approach:** A simple and effective approach is presented to extract handwritten characters, including digits and letters of any language from data filled boxes of data entry form and to deal with cases of overlaps between the handwritten characters and boxes' lines. The proposed approach is based on line shape characteristic by detecting and removing the vertical and horizontal straight boxes' lines, while preserving the curved lines which represent the handwritten characters. The problem of the handwritten characters overlapping with the data filled boxes' line is solved using morphology dilation to reconstruct the broken characters after the removal of the boxes' lines. **Results:** Experimental results have demonstrated that the proposed approach can extract handwriting from data filled boxes with overall 94.052% for data collection of 150 forms. **Conclusion:** The proposed algorithm has been successfully implemented and tested to achieve the objectives of handwritten extraction of any language from data filled boxes. However, this work could not deal with situations whereby the characters touch other immediate characters.

**Key words:** Document image processing, overlapping characters, proposed algorithm, successfully implemented, immediate characters, handwritten characters, boxes' lines, shape characteristic, extract handwritten

## INTRODUCTION

Data entry form is one of the oldest and most successful ways to collect information in a number of different fields. These forms contain empty fields can be dotted lines, check boxes and data filled boxes. The data filled box is a table with small squares designed to be filled with one character in each of the square spaces. A data filled box can be generally defined as a structured document composed by cells delimited by vertical line segments (Neves *et al*., 2006). Cells can be blank or filled with data, either printed or handwritten, as illustrated in Fig. 1.

To date, most of the data entry processes are performed online with the help of the computer. This advancement has speeded up the process of data entry.

However, manual operation is still used widely by filling in the sheets using pen and handwriting. To extract the information from these forms in the computer systems, several processes would be needed.

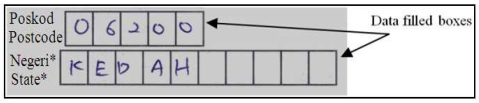



Fig. 1: Example of data filled boxes

**Corresponding Author:** Ali Qusay Al-Faris, School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Engineering Campus, Seri Ampangan 14300 Nibong Tebal, Penang, Malaysia

Extracting the handwriting from the data entry forms is important for many purposes such as in documenting and archiving. The extraction process is also important in situations such as when it is necessary to change the handwritten text into computer printed characters in order to easily organize the information in the entry forms into database systems. This may also speed up the data entry process.

Character extraction problems have been solved by some researchers using variant methods. Neves *et al.* (2006) proposed the cell extraction method for Table Form Segmentation which consists of steps such as initially locating and extracting the intersections of table lines. The weakness of this method is that the process involved complicated table extractions. Chen and Lee (1996) presented a novel approach using a gravitation-based algorithm. However, in their work, some field data could not be extracted correctly, which led to mis-extraction. Tseng and Chuang proposed the stroke extraction method (Tseng and Chuang, 1992) and then used it for the Chinese characters (Tseng and Chen, 1996). However, the method did not solve the overlapping problems. Liolios *et al.* (2002) described a system for form identification based on power spectral density of the horizontal projection of the blank form to obtain the feature vectors. Here too, the overlapping problem has not been addressed.

Lines are the essential elements of a form and their extraction is an important process before data extraction. This is evident in the works of researchers such as Pizano (1992) Paquet and Lecourtier (1991) Moreau *et al.* (1991) Guillevic and Suen (1993) Mandal *et al.* (2006; 2005) However, in these researches, data extraction is degraded when characters are overlapped by boxes' lines. Boatto *et al.* (1992a; 1992b) proposed an interpretation system for land register maps. Their system could only work on the assumption that the symbols consist of certain topological structure. On the other hand, Wang and Srihari (1994) proposed a system to analyze form images. Their shortcoming was that the system could not patch the broken characters well in all conditions; it would cause the data to further deteriorate in some cases. (Casey and Ferguson, 1990 and Casey *et al.*, 1992) proposed Intelligent Form Processing systems (IFP). Similarly, the setback here was that the processes of overlapped characters in related works usually lose partial information of the character and the broken strokes are not well reconstructed before recognition, such that the performance of character recognition is degraded. In this study, we propose simple algorithms based on the differentiating characteristics of the shape of the boxes' lines and the handwritten characters lines.

## MATERIALS AND METHODS

The proposed approach starts with preprocessing phases, i.e., converting to gray level, image binarization and noise reduction and then distinguishing the handwritten characters from the box's lines based on the characteristic differences of the line shapes between the two. Finally, the approach overcomes the problem of overlapping using the Morphological operations.

**Document Preprocessing:** First, the entered scanned form image is converted to grey level using the standard conversion in Eq. 1:

$$\text{Grey} = 0.3R + 0.59G + 0.11\ B \quad (1)$$

where, R, G and B represent the colors red, green and blue respectively, with values from 0-255.

Then Binarization operation is necessary to clarify the main foreground objects in the form such as characters and boxes from the background. A fixed intensity threshold value is chosen for the global thresholding algorithm whereby the pixel value of an input image is set to white if its intensity is more than the threshold and vice versa (He *et al*., 2005). A non-linear digital filtering technique is used to reduce noise in the form image. The median filter is chosen to remove the salt and pepper noise in the document before proceeding with the higher level processing steps (MuhdZain *et al*., 2009; Ouchtati *et al.*, 2007).

**Boxes' lines detection and deletion:** Identification of the straight lines is based on the differentiating properties of line handwritten characters and the boxes. The line's shape has measurable characteristics of the two lines, while the boxes' lines shape are straight vertical and horizontal, the shape of characters' lines consist of curved lines or zigzags.

We propose a simple algorithm to distinguish between the curved lines shapes which represent handwritten characters and straight lines which represent the box. This approach is divided into three sub-steps i.e., vertical straight boxes' lines detection and deletion, horizontal straight boxes' lines detection and deletion and removal of the line segments of the remaining boxes.

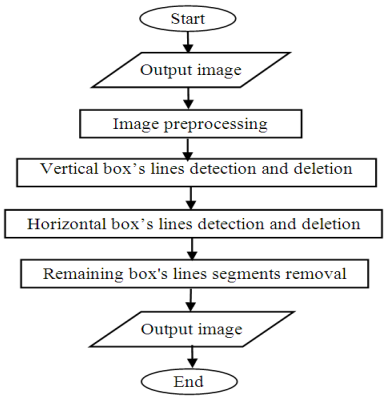



Fig. 2: Flow chart for the algorithm of filled box's lines detection and deletion

The same technique is used for the two stages of identifying and eliminating the vertical and horizontal lines. The only difference is that for the vertical lines, scanning is done by column by column while for the horizontal lines; the scanning is done row by row. The algorithm is illustrated as in the flow chart of Fig. 2.

Identifying the vertical and horizontal lines starts by setting a minimum threshold value, the specific threshold value chosen after a number of experiments on the collection of data. The algorithm scans the image's array vertically/horizontally, then compares the summation of connected black pixels in the same column/ row to the threshold value. If the summation value is greater than threshold value, the segment of connected black pixels in that column/row is considered as a straight line (Str). Otherwise it is a curved line (Crv). After the straight line detection, the black pixels of the line are changed to white, while the curved line is preserved. This is repeated for all columns/rows in the image's array column by column from top to bottom. This operation can be formulated as follows Eq. 2 and 3:

$$Crv = \sum_{c=0}^{n} \mathcal{B} < \mathcal{T} \quad (2)$$

$$Str = \sum_{c=0}^{n} \mathcal{B} > \mathcal{T} \quad (3)$$

Where:

C = The first pixel in the column/row,
N = The number of pixels in the column/row,
(B) = The black pixel
J = The threshold value for straight lines.

The algorithm's steps are illustrated as follows:

- Set a minimum threshold value (J) for comparing
- Set counter of black pixels in the column/row (B) = 0
- Scan first column/row in the image's array vertically/horizontally and add each black pixel to counter/ row (B)

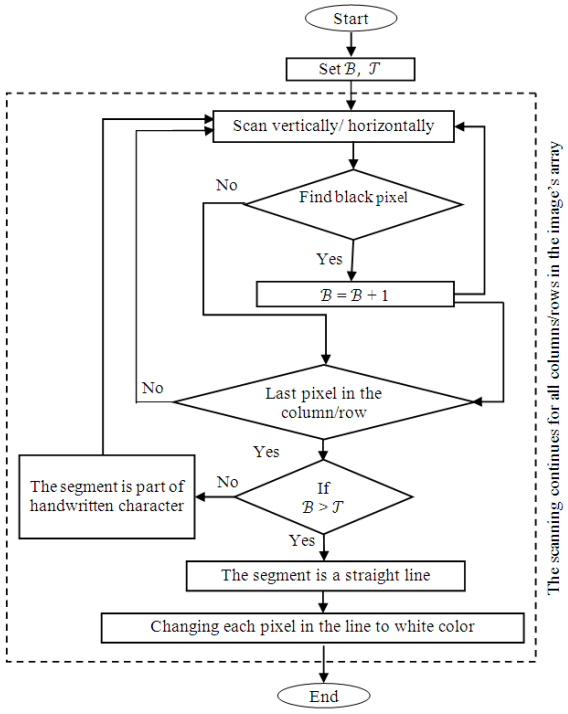



Fig. 3: The proposed algorithm to detect and delete the straight box lines

- If (B>J) then the segment is straight line
- Change each black pixel in the detected straight line to white pixel
- Repeat steps (2-5) until the last column/ row in the array

The flow chart of the algorithm is illustrated in Fig. 3. After deleting the vertical and horizontal lines some broken small segments of the boxes' lines may still remain which is caused by the scanned form poor quality image. Thus the median filter is used. After removing the box's lines, the character that is involved in the process will appear as broken characters. To overcome the missing pixels in the broken character, the morphological dilation operator is used. This is to gradually enlarge the boundaries of the regions of foreground pixels. Thus areas of the foreground pixels grow in size while holes within those regions become smaller. Soori *et al*. (2007) Two pieces of data are used as inputs. The first is the image which is to be dilated (here, the image is a broken character). The second is a set of (usually small) coordinate points i.e., the structuring element (also known as a kernel). This kernel determines the precise effect of the dilation on the input image. Here, the chosen structuring element is a 3×3 matrix as follows:

0 1 0
1 1 1
0 1 0

Equation 4 defines the basic morphological operator dilation on sets A and B Eq. 4:

$$A \oplus B = \bigcup_{b \hat{I} B} A_b \quad (4)$$

Where:
A = The image array
B = The structuring element(Gonzalez and Woods, 2008; Soori *et al*., 2007)

The end result is the gaps between the handwritten character images with the structuring elements are bridged. However, the side effect is, the character becomes thicker. Therefore, a thinning procedure will be applied in order to obtain the original thickness of the character (Gonzalez and Woods, 2008; Aljuaid *et al.*, 2010).

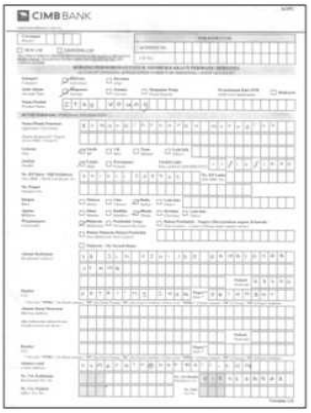
## RESULTS

This study has been tested on three types of data entry forms of i.e.; university application form (Fig. 4a) and two types of bank account opening application (Fig. 4b and c). The forms have several field types and one of these fields is the data filled box table (the scope of this study focuses on this region in the document). 50 forms of each form type (university, bank1, bank2) samples had been distributed and filled by handwritten characters with blue and black pens.The total number of collection forms is 150 forms and there are 1900 filled boxes. These samples were then scanned and digitized with a resolution of 300 dpi. After the application of the preprocessing (binarization and noise reduction) followed by the testing phase of the boxes' lines detection and deletion, the subsequent proposed algorithm is then used. These include the three substeps; vertical straight boxes' lines detection and deletion, horizontal straight boxes' lines detection and deletion, removal of the remaining segments of lines using the median filter and morphological dilation and thinning.

The results obtained are illustrated in the two example images in Fig. 5. Figure 5a shows the original image which includes handwritten letters with two overlapped characters (4, 7). Figure 5b shows the original image which include handwritten digits with two overlapped characters (R, T). Figure 5c-f show the resultant images after removing the vertical and horizontal table lines respectively. The gaps in characters (4, 7) and (R, T) can also be noticed in these Figures. Figure 5g and h shows the images after the removal of the remaining segments.


(a)


(b)




(c)

Fig. 4: Form samples: (a) university application form; (b) bank account application form 1; (c) bank account application form 2

Figure 5i and j show the images after applying the dilation operation to reconstruct the gaps in-between the characters. Fig. 5k,l show the final images after applying the thinning operation.

As of the present moment, there is no standard mechanism to judge the correctness of the result. The results are measured by visual observations.

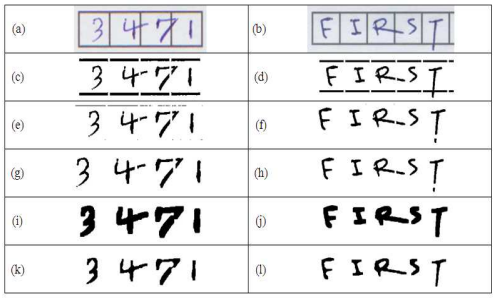


Fig. 5: The approach in implementation: (a, b) input images; (c, d) vertical boxes' lines detection and deletion; a (e, f) horizontal boxes' lines detection and deletion; (g, h) remaining segments removing; (i, j) after applying dilation operation; (k, l) after applying the thinning operation.

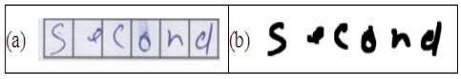


Fig. 6: One of the bad resultant images: (a) input image; (b) the image after extraction and dilation

Table 1: The results of the proposed approach

| Form Type | No. of samples | No. of filled boxes | No. of bad results | Bad rate | Correct rate | Error rate (%) |
|---|---|---|---|---|---|---|
| Bank1 | 50 | 800 | 53 | 6.375 | 93.625 | 0 |
| Bank2 | 50 | 650 | 37 | 5.692 | 94.308 | 0 |
| University | 50 | 450 | 26 | 5.778 | 94.222 | 0 |
| Total | 150 | 1900 | 116 | 5.948 | 94.052 | 0 |

The results are defined as "Correct" if all the vertical and horizontal lines of boxes are removed correctly and all gaps in handwritten characters are bridged if there is any overlapping problem. The results are considered "Bad" if some of the boxes' lines are not removed correctly or if there is disfigurement in the handwritten characters after solving the overlapping problem. If all of the lines still remain or the overlapping problem is not solved, the results are considered "Error". We use Correct Rate, Bad Rate and Error Rate, which are illustrated in (Table 1).

## DISCUSSION

From the results given in Table 1, it can be seen that some preprocessing operations have been done, i.e., binarization using thresholding method and noise reduction using filtering. To extract the handwritten characters from the data filled boxes, the proposed algorithm has the ability to find the straight vertical and horizontal lines in the image that represent the boxes' lines and also delete them. The algorithm keeps the curved lines which represents the handwritten characters. The output results have shown that the proposed algorithm is able to solve the problem. However, this study is focused only on the data filled box regions. The problem of overlapping between the handwritten characters and boxes' lines is solved using morphological dilation to reconstruct the broken characters after the removal of the boxes' lines followed by morphological thinning. The experiments also returned good results. However, 5.948% of the results show that the character is disfigured after reconstruction using dilation especially when the handwritten character is blurry, or when the holes in the handwritten character is very small as indicated in the character "e" as in Fig. 6b.

## CONCLUSION

In this study, the objectives of developing an algorithm to extract handwritten characters from data filled boxes of data entry form and solving the problem of the overlaps between the handwritten characters and the filled boxes lines have been met. The proposed algorithm have also been successfully implemented and tested to achieve the objectives. However, this study could not deal with situations whereby the characters touches other immediate characters. For future development, this study can be expanded in a number of ways. These may include the extraction of handwritten characters from the other regions of the data entry form, such as dotted lines, check boxes and data tables. At the same time, the extension of this study would also focus on methods to improve the problem handling of character touching other characters on the designated forms.

## REFERENCES

Aljuaid, H., Z. Muhammad and M. Sarfraz, 2010. A tool to develop arabic handwriting recognition system using genetic approach. J. Comput. Sci., 6: 619-624. DOI: 10.3844/jcssp.2010.619.624

Boatto, L., V. Consorti, M. De Buono, S. Di Zenzo and V. Eramo *et al*., 1992a. An interpretation system for land register maps. Computer, 25: 25-33. DOI: 10.1109/2.144437

Boatto, L., V. Consorti, M.D. Buono, V. Eramo and A. Esposito *et al*., 1992b. Detection and separation of symbols connected to graphics in line drawings. Proceedings of 11th International Conference on Pattern Recognition, Aug. 30-Sep. 3, IEEE Xplore press, The Hague , Netherlands, pp: 545-548. DOI: 10.1109/ICPR.1992.201837

Casey, R., D. Ferguson, K. Mohiuddin and E. Walach, 1992. Intelligent forms processing system. Machine Vision Appl., 5: 143-155. DOI: 10.1007/BF02626994

Casey, R.G. and D.R. Ferguson, 1990. Intelligent forms processing. IBM Syst. J., 29: 435-450. DOI: 10.1147/sj.293.0435

Chen, J.L. and H.J. Lee, 1996. Field data extraction for form document processing using a gravitation-based algorithm, Pattern Recognit., 34: 1741-1750. DOI: 10.1016/S0031-3203(00)00115-1

Gonzalez, R.C. and R.E. Woods, 2008. Digital Image Processing. 3rd Edn. Pearson/Prentice Hall, Upper Saddle River, NJ., ISBN: 013505267X, pp: 954.

Guillevic, D. and C.Y. Suen, 1993. Cursive script recognition: A fast reader scheme. Proceeding of 2nd International Conference on Document Analysis and Recognition, Oct. 20-22, IEEE Xplore Press, Tsukuba Science City, Japan, pp: 311-314. DOI: 10.1109/ICDAR.1993.395725

He, J., Q.D.M. Do, A.C. Downton and J.H. Kim, 2005. A comparison of binarization methods for historical archive documents. Proceedings of the 8th International Conference on Document Analysis and Recognition, Aug. 29-Sep. 1, IEEE Xplore Press, UK., pp: 538-542. DOI: 10.1109/ICDAR.2005.3

Liolios, N., N. Fakotakis and G. Kokkinakis, 2002. On the generalization of the form identification and skew detection problem. Pattern Recognit., 35: 253-264. DOI: 10.1016/S0031-3203(01)00030-9

Mandal, S., S.P. Chowdhury, A.K. Das and B. Chanda 2005. A hierarchical method for automated identification and segmentation of forms. Proceedings of 8th International Conference on Document Analysis and Recognition, Aug. 29-Sep. 1, IEEE Xplore Press, India, pp: 705-709. DOI: 10.1109/ICDAR.2005.17

Mandal, S., S.P. Chowdhury, A.K. Das and B. Chanda, 2006. Fully automated identification and segmentation of form document form processing, springer, Comput. Graphics, 32: 953-961. DOI: 10.1007/1-4020-4179-9_139

MuhdZain, M.L., I. Elamvazuthi and M. Begam, 2009. enhancement of bone fracture image using filtering techniques. Int. J. Video Image Proc. Network Security, 9: 49-54.

Neves, L.A.P., J.M. De Carvalho, J. Facon and F. Bortolozzi, 2006. A new table extraction and recovery methodology with little use of previous knowledge. Proceedings of the International Workshop on Frontiers in Handwriting Recognition, (FHR'06), La Baule. France.

Ouchtati, S., M. Bedda and A. Lachouri, 2007. Segmentation and Recognition of Handwritten Numeric Chains. J. Comput. Sci., 3: 242-248. DOI: 10.3844/jcssp.2007.242.248

Pizano, A., 1992. Extracting line features from images of business forms and tables. Proceedings of the 11th IPAR International Conference on Pattern Recognition, Aug. 30-Sep. 3, IEEE Xplore Press, The Hague , Netherlands, pp: 399-403. DOI: 10.1109/ICPR.1992.202008

Tseng, L.Y. and C.T. Chuang, 1992. An Efficient Knowledge-based stroke extraction method for multi-font Chinese character. Pattern Recognition, 25: 1455-1458. DOI: 10.1016/0031-3203(92)90119-4

Tseng, L.Y. and R.C. Chen, 1996. An efficient recognition and data extraction method for table-form documents. Proceedings of the IAPR Workshop on Machine Vision Applications, Nov. 12-14, Tokyo, Japan, pp: 131-134.

Wang, D. and S.N. Srihari, 1994. Analysis of form images. Int. J. Pattern Recognit., 8: 1031-1031.