Detection of Unknown Forms from Document Images

Andrew Busch a.busch@qut.edu.au
Wageeh W. Boles w.boles@qut.edu.au
Sridha Sridharan s.sridharan@qut.edu.au
Vinod Chandran v.chandran@qut.edu.au
Research Concentration in Speech, Audio and Video Technology,
Queensland University of Technology, Brisbane, Qld, Australia

Abstract

This paper presents a novel technique for distinguishing images of forms from other document images. The proposed algorithm detects regions which are likely to be used for text entry, such as lines, boxes, and character entry fields, and calculates a probability of the document being a form based on the presence of such structures. Experimental results from testing on both filled and unfilled forms, as well as a selection of non-form documents are presented. All document images are assumed to have been scanned at a known resolution.

INTRODUCTION

The extraction and processing of information contained in printed forms is a task of great importance in many areas of business and government alike. To date, the vast majority of form processing has been done manually, with human operators performing all of the associated tasks up to and including data entry. In recent times, a large amount of research has been undertaken in the fields of form identification, field location and data extraction. All of the research to date, however, makes the assumption that the image to be analysed is indeed a form, which may not always be the case in many applications. For this reason, this paper presents a technique for classifying a document image as either a form or non-form, and identifying likely field areas within any forms detected.

Previous work in the field of form field detection has provided an excellent starting point for this research. The technique proposed by Wang and Srihari [1] removes isolated characters, then searches for intersections of line segments. Yuan et al [2] present a method of detecting fields in forms that relies on segmentation algorithms to find text and straight lines, and uses adjacency graphs to detect possible entry fields in form images with no text entered. Xingyuan et al [3] propose a more robust technique which detects rectangular fields and lines regardless of text or other markings, but does not explicitly detect other form structures.

A number of techniques have also been proposed to remove the effects of noise and poor image acquisition, which can often cause unwanted line breaks, false intersections and broken junctions [4-6].

The work presented in this paper is in two parts. The first section describes a technique for detecting the primitive data entry structures that distinguish forms from other documents, namely lines, bounded rectangular areas, checkboxes, and character cell fields, or 'tooth' structures. In the second section we attempt to determine if an unknown document is likely to be a form. Using the presence of the previously detected structures, combined with the amount of text found in the document, a form probability score is proposed as an indication of the likelihood of the candidate document being a form.

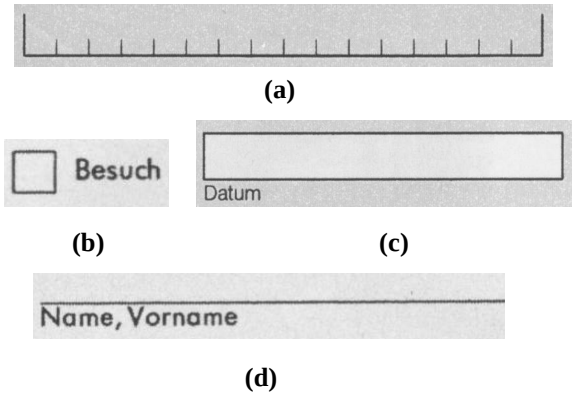
Results from experiments over 100 form and 200 non-form document images from a variety of sources are presented.

DETECTION OF FORM STRUCTURES

An initial investigation of documents contained in [7] has identified four major structural elements which can be used to identify forms. These are: horizontal lines (either solid or dotted), bounded rectangles, small checkboxes, and character cells or 'tooth' structures (Fig 1). Examination of all training data has shown that every form document contains one or more such structures. Detecting such structures in complex document images, however, is not a trivial problem. Attempting to segment a document image and classify regions is problematic due to frequent overlapping of neighboring regions, especially when dealing with completed forms. More traditional shape recognition techniques such as the generalized Hough Transform [8] are also inaccurate in the presence of noise, and also quite slow computationally. As all of the desired regions consist entirely of vertical and horizontal lines, our approach to the detection problem begins with finding all such lines in the candidate image. Once these lines are found, each is further processed to determine if it is a likely form structure.

Line Detection

We define a 'line' in a document image to be a contiguous or near-contiguous sequence of n 'on' pixels in the horizontal (vertical) direction, where n is directly proportional to the resolution of the image. As the smallest lines of interest are approximately the same width as a character, n is chosen as to correspond with a distance of 2mm in the original document. To detect such a sequence, we employ a one-dimensional summing filter in the horizontal (vertical) direction defined by the equation



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

Figure 1. Four common form structures, (a) tooth structure, (b) checkbox, (c) rectangle, (d) line

$$S(x,y) = \frac{1}{n}\sum_{i=x}^{x+n} I(i,y) \quad (1)$$

where *I(x,y)* is the original binary image. By applying a threshold to the resulting image, the starting points of all possible lines can be found.

$$L_H = S(x,y) \geq \tau \quad (2)$$

Binary morphological operations can then be used to extend these starting points across all *n* pixels in the line segment. Figure 2 shows the result of line detection on a typical form image.

The detection process thus outlined is successful at detecting regions likely to contain lines, however also gives rise to a number of false positives. In particular, large regions black regions in the document such as images, thick vertical lines and large sections of text are often falsely detected as horizontal lines. To remove such regions we first segment the line image L_H into connected components, and calculate the height and width of each component. Components which do not satisfy a minimum width and width:height ratio are removed. This process also has the effect of removing valid horizontal lines which are connected to thick vertical lines, however as such lines are almost always borders or part of images, this is not undesirable.

Vertical lines by themselves do not constitute a possible text entry field. For this reason, all vertical lines which do not at some point cross a valid horizontal line are also removed. To allow for noise, small breaks in lines, and scanning errors, we relax this constraint somewhat, allowing vertical lines which are close (within *n* pixels) to either a horizontal line or another valid vertical line to be kept as well.

Line Grouping

Once all possible lines have been detected, we then attempt to combine these lines to form one of the four form structures.

In order to detect character cells or 'tooth' structures, each horizontal line is analysed for vertical line crossings, or near crossings. Such crossings must extend significantly in the vertical direction, since we assume that the horizontal line represents the bottom of the tooth structure. We then look for a periodic structure within these crossings, constrained by likely cell size. Due to noise, handwriting or other markings within the structure, it is possible that extra vertical crossings unrelated to the structure are present. In order to allow for this, an algorithm has been developed as follows:

For every vertical line crossing not *already part of structure:*
  search for more crossings within search dist. x
  for each such crossing found:
    search line at same dist. ±5%
    if another crossing found,
      recalculate mean distance, search again
      if #crossings > 4, structure found.

The search distance *x* is proportional to the resolution of the document, and we have used a range of *n*-5*n* in our experiments with good results. In order to reduce the false detection rate, we have also enforced a criterion whereby a structure is *not* considered valid if more than half of its crossings are not fully joined. Finally, we search for a top bounding line, which is defined as a horizontal line within *n*-5*n* of the original lines, which crosses (or nearly crosses) each vertical segment of the structure. If two or more such lines are found, only the closest to the baseline is taken. Any such line found will still be considered for the baseline of further tooth structures.

For the detection of rectangles and boxes, we use a similar algorithm to that proposed in [3], whereby each set of candidate lines are checked to determine whether they form an enclosed area. In order to prevent rectangles being found in locations already covered by previously detected tooth structures, baselines of such structures are only considered as the *top* of a rectangle. Small breaks in the perimeter of rectangles are permitted, so long as they do not exceed 5% of the total distance. Rectangles that are completely covered by other rectangles are then removed. Regions whose area exceeds a certain size threshold are also removed, as these are unlikely to be text entry fields, and are more likely borders or frames.

A checkbox is defined as a special case of rectangle, where the following three criteria are met:

- The sides are of equal length (square)

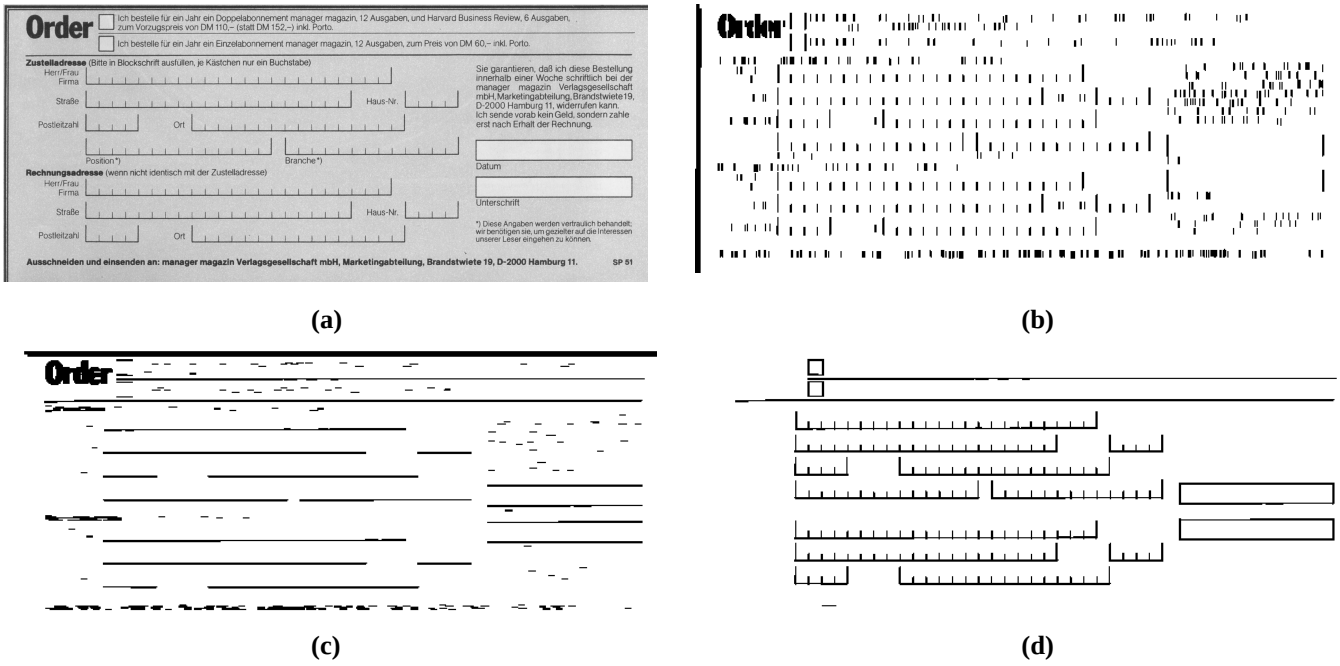


Figure 2. Results of form structure detection. (a) Original document image, (b) detected vertical line segments, (c) detected horizontal line segemnts, (d) final form structures

- Side length is within a given range (we use *n*- *5n*)
- Sides do not significantly extend beyond the corners of the rectangle

All horizontal lines that do not form part of any of the above structures are considered lines.

FORM CLASSIFICATION

The classification of documents into form and non-form classes is achieved using a score based on the presence of previously detected form structures combined with the amount of text contained in the document. Examination of a large number of forms has revealed that most do not contain as much text as other documents of a similar size. Thus, the presence of text in a document image has a negative impact on the probability of that document being a form. Numerous algorithms exist for the segmentation and extraction of printed text from documents , but for accuracy we have manually measured the amount of text present in each test document. As we are only interested in the body text of the document, any large segments such as headlines or titles are not included. We thus define the form probability score as:

$$p = \mathbf{w}_1 d_{tooth} + \mathbf{w}_2 d_{box} + \mathbf{w}_3 d_{rect} + \mathbf{w}_4 d_{line} - \mathbf{w}_5 d_{text} \quad (3)$$

where d_{type} represents the total horizontal lineal distance covered by the given structure type, and **w** is a weighting vector. A positive *fps* value indicates that the document is likely to be a form. In order to obtain a true likelihood estimate, this value can be normalised, such that:

$$\hat{p} = \frac{p}{\left(d_{tooth} + d_{box} + d_{rect} + d_{line} + d_{text}\right)} \quad (4)$$

In order to calculate the weighting vectors we have processed a large number of both form and non-form documents, and examined the relationship between the amounts of each structure present. By constructing plots of dtext vs dtype for each structure type, it can be seen that there exists an almost linear separation between form and

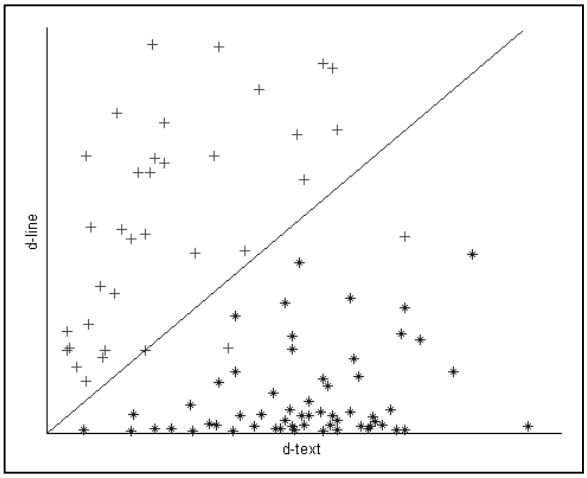


Figure 3. Plot of d-line vs d-text for a selection of form(+) and non-form(*) document images

non-form documents. We then find the gradient of this line and use it to calculate the corresponding weighting coefficient in **w**, assuming $\mathbf{w_5} = 1$. Figure 3 shows an example of such a plot. It should be noted that we could find no non-form documents containing the tooth structure, meaning that the value of $\mathbf{w_1}$ would approach infinity. For this reason we have made this coefficient very large, approximately ten times the value of the next highest coefficient.

RESULTS

Experiments were conducted in two stages, using a set of 100 form and 200 non-form images acquired from a variety of sources, including the University of Washington database [7]. Firstly, the form structure detection algorithm was applied to all form images, and results compared to those calculated manually. Overall, 2443 of 2567 (95%) form structures were successfully detected as the correct type. Of those structures that were not detected successfully, approximately two thirds were due to misclassification of one structure as another, with the remaining missed entirely. An additional 181 form structures were falsely detected, with almost all of these being small lines. A typical form image with all detected structures is shown in Figure 2. Table 1 shows the confusion matrix for this experiment.

Table 1. Confusion matrix for detection of form structures

Actual Type	Detected Type				
	tooth	**rect.**	**box**	**line**	**missed**
tooth	229	0	0	3	0
rect	0	767	9	38	0
box	0	10	318	4	14
line	1	15	1	1253	19
none	0	9	4	168	x

The second stage of experiments involved calculating the normalised form probability score for each test document using the detected structures and known text amounts. Those documents obtaining a positive score were classified as forms, with the remaining classified as non-forms. From a total of 300 (100 form, 200 non-form) document images, 258 were correctly classified. Of those that were misclassified, 6 form images were missed, and 36 non-form images falsely detected. The overall error rate of the test was approximately 14%. Total processing time for both structure detection and form classification is approximately 5 seconds on a Pentium 3 600MHz computer.

CONCLUSIONS AND FUTURE RESEARCH

This paper has presented a technique to distinguish form documents from other types by identifying common structures usually present in such images. Experimental results have shown our algorithm for detecting such structures to be accurate and robust, with over 95% of structures detected correctly. Classification of form and non-form documents is accomplished by comparing the total number of such structures to the amount of text in the document, creating a form probability score. This statistic has shown to perform well, with almost all form images correctly identified and a false detection rate of under 15%.

Future research will aim to more accurately model the typical line and rectangle structure in forms by examining surrounding text. This should greatly reduce the number of false positive results.

REFERENCES

[1] D. Wang and S. N. Srihari, "Analysis of form images," Proc. of First International Conference on Document Analysis and Recognition, 1991.

[2] J. Yuan, Y. Tang, and C. Y. Suen, "Four directional adjacency graphs (FDAG) and their application in locating fields in forms," Proc. of Third International Conference on Document Analysis and Recognition, 1995.

[3] L. Xingyuan, D. Doermann, W.-G. Oh, and W. Gao, "A robust method for unknown forms analysis," Proc. of Fifth International Conference on Document Analysis and Recognition, 1999.

[4] H. Shinjo, K. Nakashima, M. Koga, K. Marukawa, Y. Shima, and E. Hadano, "A method for connecting desappeared junction patterns on frame lines in form documents," Proc. of 4th Int. Conf. on Document Analysis and Recognition, 1997.

[5] O. Hori and D. Doermann, "Robust table-form structure analysis based on box-driven reasoning," Proc. of Third International Conference on Document Analysis and Recognition, 1995.

[6] H. Fujisawa and Y. Nakano, "Segmentation methods for character recognition: from segmentation to document structure analysis," *Proceedings of the IEEE*, vol. 80, pp. 1079-1092, 1992.

[7] I. Phillips, S. Chen, and R. Haralick, "CD-ROM Document Database Standard," Proc. of Second International Conference on Document Analysis and Recognition, 1993.

[8] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, pp. 111-122, 1981.