Applying Feature Based Word Recognition Approach to Screen Text Recognition

G. Raza, A. Hennig, N. Sherkat, R. J. Whitrow
Department of Computing, The Nottingham Trent University,
Burton Street, Nottingham NG1 4BU, UK
{ghr,amr,ns,rjw}@doc.ntu.ac.uk

Abstract

In earlier work we described a feature based word recognition method for the recognition of poor quality words taken from fax messages. Various independent and robust features are used to identify alternatives of every object of the word without attempting to segment touching objects. Later, a lexical lookup method is used to verify the alternatives In this paper, the developed method is applied to screen text of different fonts and sizes in order to observe its performance on screen image. It has been observed that the developed method is capable of recognising screen text of varying fonts and sizes whilst avoiding segmentation of touching characters.

1 Introduction

Character segmentation is a key step in most conventional Optical Character Recognition (OCR) systems. This segmentation based approach is possible for good quality prints, where characters are clearly separated from their neighbours. However, such an approach is unsuitable for the documents containing characters touching their neighbouring characters. The performance of existing OCR systems is not acceptable for such documents. Documents of this type come from many different sources such as facsimile messages, low quality prints, photocopies etc. Furthermore, existing OCR systems are unable to deal with screen images[1][2].

A feature based word recognition method [3] for the recognition of poor quality word images has been developed. For this paper, this method has been modified and applied to screen text of different fonts and sizes. The results obtained show the ability of the method to cope with screen images.

In this method, different independent features of each object (which could be a single character or several touching ones) of a word are extracted from the input sample word. Different alternatives for each object are found by comparing the features with a database of ideal features for each object. These alternatives are ranked according to the number of features matched. The alternative with the most features matched is considered to be the most likely candidate.

These alternatives are combined to form different words using a dictionary lookup step. These words are also ranked according to the total number of features matched. The word with most features matched is deemed to be the recognised sample word. If there is more than one word, then higher level linguistic information, such as language syntax and semantics, can be used to identify the correct word [4][5][6].

2 Feature Extraction

Feature extraction is an important stage in the recognition of characters particularly in the case of poor quality documents. In order to recognise a character, different features are extracted, which will (hopefully) exhibit the distinctive characteristics of the character[7]. Ideally, the features should enable the recognizer to discriminate correctly betwwen distinct classes of characters.

In the literature different commonly used feature extraction methods have been described , e.g. global features, distribution of points, geometric and topological features, linguistic descriptions, use of context and fuzzy sets. There is no general technique for the design of feature extraction which utilises the designer's a priori knowledge of the recognition problem. Geometrical and topological features are commonly used by human beings in the recognition of patterns because such features can easily be detected by the human eye.

In the current research, various independent features of every object of the sample words are extracted and compared against the ideal form of the object (see Figure 1). Features are: the zones in which the objects are found (middle zone only, upper or lower zones and full, i.e. both upper and lower zone), vertical bars, holes (in the upper,

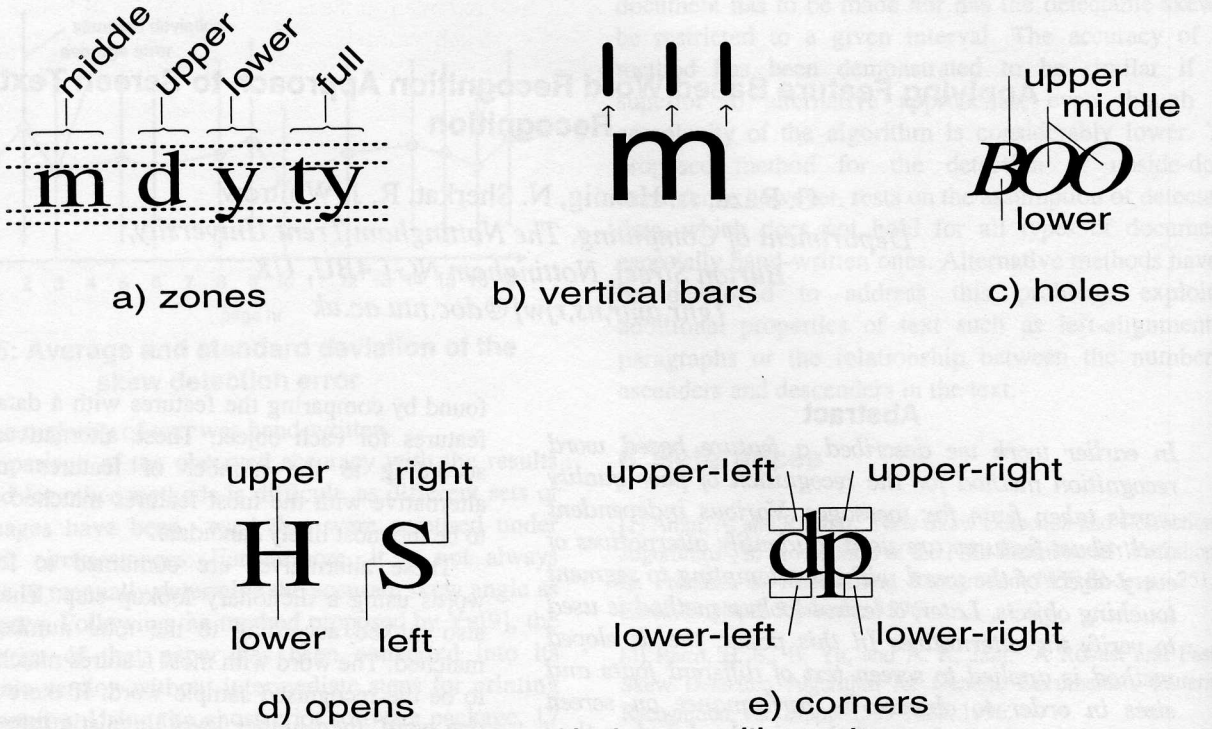

Fig. 1: Features used in the recognition system

middle or lower part of the object), and openings in any of the eight principal directions (i.e. upper, left, lower right side opens as well as upper left, lower left, lower right, upper right corners)

These features have been selected in order to achieve the following goals:

(a) We are primarily dealing with poor quality documents. Therefore, consideration has been given to define features, which are generally not affected by the image quality. As this is not always possible, at least the majority of features are designed to be unaffected by additional or missing parts of the object.

Fig. 2: Objects width missing or additional parts

In Figure 2, the first letter has an unwanted extra part in its top half. This extra part leads to converting top side open feature of this object to the upper hole. But this change has not affected the other features of this object, e.g. bottom side open, left side open, right side open and vertical bars etc. Similarly, in Figure 2, some part in the upper half of the second object is missing. Therefore, instead of upper hole of that object, a top side open feature may now be extracted. Nevertheless other features of this object can be extracted correctly and are sufficient for its classification.

(b) We are dealing with multi-size documents i.e. sample text may be of any size from 6 point to 48 point size. Therefore attempts have been made to find features, which are generally the same for different sizes of a particular object. As an example feature, we consider upper holes as present in the letter 'A'. This upper hole (as well as the lower open) should be found in letters 'A' of any size. Similar applies to the upper and lower open of the letter 'H', as shown in Table 1.

	10pts	12pts	18pts	24pts
'H'	H	H	H	H
'A'	A	A	A	A

Table 1: Letters with different sizes having same features

(c) We are also dealing with multi-font documents. Therefore, attempts have been made to find features, which remain unchanged for several fonts. For example, the upper open feature in the letter 'U' should be present different fonts, e.g. Helvetica, Courier, Arial and Times Roman etc. Similar applies to the lower hole of the letter 'a', as can be seen in Table 2.

	Courier	Helvetica	Times Roman
'a'	a	a	a
'U'	U	U	U

Table 2: Letters with different fonts having same features

Our experiments have shown that these features are reasonable tolerant to noise such as broken, touching and degraded characters.

3 Database Development

The presented approach aims to recognise each object within word as a whole, without trying to segment touching characters. Hence the proposed method requires a database of single letter features (a-z, A-Z), as well as a database of touching letters

Development of a single letter database is trivial. In order to develop two letter and three letter databases, all possible combinations of two letters and three letters in the words of a 4k dictionary were found. Later, all these combinations are stored in a file along with their features. Hence as a whole, we have used three different kinds of databases namely database 1 containing single letters (a, b, c, etc.), database 2 containing two letter combinations (ab, ac, al, ol, etc.) and database 3 containing three letter combinations (aba, ack, arn, etc.) along with their features. The ideal features of each object are currently defined manually. Work into automatic creation of the database is currently being carried out.

4 Algorithm Description

The algorithm for word recognition is illustrated in Figure 3.

The image of a document is first separated into lines. The word images are then identified within a given line. Each word image is passed on to the object separator, where it is separated into different objects. Features of every object of the word are extracted next. Comparing the features with the database yields a list of alternative interpretations for each object. Words from a given dictionary are then sought that can be built from the object alternative lists

The two major parts of the recognition system are described in more detail below, namely finding object alternatives and searching for valid words

4.1 Finding Object Alternatives

Having found the features of an object, its height and width is calculated. Then, the minimum (L_{min}) and maximum (L_{max}) number of letters in the object are estimated, using the following heuristics.

$$L_{\min} = \frac{height}{width} + 1 \quad , \quad L_{\max} = \frac{height}{width} + 2 \quad \text{(EQ1)}$$

We then compare the features of this object with the corresponding features of database objects containing objects of length L_{min} and L_{max}. For every feature f the matching number (M_n) is found using the following equation.

$$M_f = 1 - \left| features_{db} - features_{sample} \right| \quad \text{(EQ2)}$$

The sum of all M_f yields the total matches T_m of the sample object with database object. This database object is recorded in a table of object alternatives together with its T_m. The table is kept sorted with the best alternative on top and restricted in size to a given maximum, currently 100 entries.

The above procedure is repeated for every object in the word. The alternatives of every object are stored in a separate table of object alternatives.

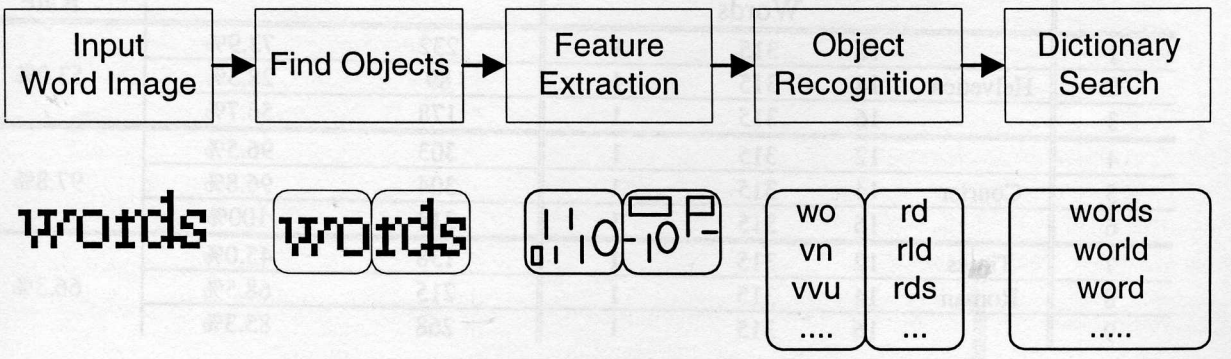

Fig. 3: Word recognition algorithm: Data flow and example.

4.2 Search Dictionary Words

Object alternative may represent single characters or a combinations of characters. For each table, we find the minimum and maximum number of characters per object. We use these figures to calculate the longest and shortest possible words, thus giving us the range $[W_{min}, W_{max}]$ of the length of words that may be formed using these tables of alternatives, i.e.

$$W_{min} = \sum_{i=1}^{nrObjects} shortest\ object\ length_i$$
$$W_{max} = \sum_{i=1}^{nrObjects} longest\ object\ length_i \quad \text{(EQ3)}$$

For each dictionary word, whose word length is within $[W_{min}, W_{max}]$, it is determined whether this word can be constructed from the tables of object alternatives using a recursive search. If the word can be constructed, the total number of matching features for the word (W_m) is calculated as the total of number of matching features (T_m) for each object alternative used to form the word.

$$W_m = \sum_{i=1}^{nrObjects} T_{m,i} \quad \text{(EQ4)}$$

The word formed from the tables of alternatives is stored in a further table along with its total number of matching features. This table is also kept sorted by the total number of matching features maintaining the best word alternative at its top.

The word(s) with the highest total number of matching features in the resulting list are considered to be the recognised word. If several words have identical high W_m higher level linguistic information such as language syntax and semantics may be used to identify the correct word. We can thus further improve the performance of the recognition system.

5 Experimental Results

Nine different sample documents consisting of screen text were created in order to evaluate the performance of the recognizer. Each document contained the same 315 words, one of which was not present in the dictionary. These documents were written in three different fonts (Helvetica, Courier and Times Roman) in three different font sizes (12, 14 and 16 points).

These sample documents were initially written and displayed in 'Word Perfect' and their image captured into 'Paint Brush' in order to obtain their bitmap. The 'Graphics Workshop' program was then used to convert the bitmaps into TIFF format as required by the recognizer. The results obtained from screen text are given in Table 3.

The recognition behaviour of the screen text in the Helvetica font in 12, 14 & 16 point size appears strange, as the highest recognition rate has been observed for the smallest i.e. 12 point size text and the least recognition rate is noticed for 14 point size text. However, in other two cases, i.e. the Courier and Times Roman screen text tests, the relationship between point size and recognition rate is consistent. This clearly means that recognition rate increases with an increase in the point size.

Sample No	Font	Point size	Total nr of Words	Invalid Words	Words recognised	Recognition Rate	Average Recognition Rate
1		12	315	1	232	73.9%	
2	Helvetica	14	315	1	89	28.3%	53.0%
3		16	315	1	178	56.7%	
4		12	315	1	303	96.5%	
5	Courier	14	315	1	304	96.8%	97.8%
6		16	315	1	314	100%	
7	Times	12	315	1	138	45.0%	
8	Roman	14	315	1	215	68.5%	66.3%
9		16	315	1	268	85.3%	

Table 3: Screen text recognition results

6 Discussion

The main difficulty appeared in the case of Helvetica font text documents where no linear relationship between recognition rate and the text point size was noticed, see Figure 3a. The reason behind this observation could be a severe limitation in extracting a variety of character features, including top left side open feature (in the letter 'd'), top right side open feature (in b and h), bottom left side open feature (q), bottom right side open feature (p). Those characters were almost without ascenders and descenders, all characters therefore seemed to be in middle zone. This rendered the zone-feature virtually useless.

However similar to printed text (Courier and Times Roman fonts) recognition, screen text in Courier and Times Roman fonts (see Figure 3b and 3c) were recognised according to the human capabilities. The highest recognition rate was observed for the largest point size screen text and the recognition rate kept on decreasing with reducing the text point size. This is not surprising, considering theat the size of a character determines the efficiency of the features extraction. More accurate feature extraction might therefore yield better result even for smaller characters.

7 Conclusion and Future Work

A word recognition algorithm for the recognition of screen word images is presented. The method has been applied to screen text of different fonts and sizes. The results obtained using this method are encouraging. This method avoids segmentation of touching characters and hence bypasses errors incurred during the segmentation stage of a typical OCR system. It tries to identify each object of the word based on its features. This method is capable of recognising poor quality words of different fonts and sizes. The method is robust and particularly suited for the recognition of poor quality documents which need a small dictionary.

The work described in this paper is currently in progress. Modifications involve the modification and improvement of the method, in order to enable it to cope with a larger variety of fonts and sizes and higher degrees of noise in the image. Investigations are carried out into additional features, improved matching of database and sample features, investigation into automatic creation of the databases used as well as the introduction of artificial noise during databases creation.

8 References

[1] N. Sherkat, R. J. Whitrow, K. Pugh and S. Harness, Fast icon and character recognition for providing universal access to a WIMP environment for the blind, Studies in Health Technology and Informatics", IOS Press, Amsterdam, (1993), pp. 19-23.

[2] S. Harness, K. Pugh, N. Sherkat and R. J. Whitrow, Enabling the use of windows environment by the blind partially sighted, IEE Colloquium, London, (1993)

[3] G. Raza, N.Sherkat and R. J. Whitrow, Word recognition using multiple independent features, International Conference on Natural language Processing and Industrial Applications, Vol. 2, (1996), pp. 233-236.

[4] F. G. Keenan and L. J. Evett, Applying syntactic information to text recognition, in L. J. Evett and T. G. Rose (Eds.) Computational Linguistics for speech and Handwriting Recognition, AISB Workshop, 1994

[5] A. C. Jobbins, G. Raza, L. J. Evett and N. Sherkat, Postprocessing for OCR: Correcting errors using semantic relations. in L. J. Evett and T. G. Rose (Eds.) Language Engineering for Document Analysis and Recognition (LEDAR), AISB96 Workshop, Sussex, England, 1996, pp. 154-161.

[6] T. G. Rose and L. J. Evett, The use of context in cursive script recognition, Machine Vision and Applications, Vol. 8, 1995, pp. 241-248.

[7] C. Y. Suen, Distinctive features in automatic recognition of handprinted characters, Signal Process, Vol. 4, (1982), pp. 193-207.