Pattern Recognition Letters 13 (1992) 821-826
North-Holland

November 1992

A word shape analysis approach to lexicon based word recognition

Tin Kam Ho, Jonathan J. Hull and Sargur N. Srihari

Center for Document Analysis and Recognition, State University of New York at Buffalo, 226 Bell Hall, Buffalo, NY 14260, USA

Received 9 July 1991
Revised 30 March 1992

Abstract

Ho, T.K., J.J. Hull and S.N. Srihari, A word shape analysis approach to lexicon based word recognition, Pattern Recognition Letters 13 (1992) 821-826.

A method for word recognition is presented that is based on an analysis of the shape of a word as a whole object. It is demonstrated to be a useful alternative for recognizing degraded word images that are prone to errors in character segmentation.

Keywords. Word shape analysis, word recognition, text recognition.

1. Introduction

Visual word recognition has motivated many important studies in document image analysis and pattern recognition. In many applications, a robust methodology is desired for recognition of text images that contain a wide range of font types and qualities.

Traditionally, word recognition is done by a three-step process that includes character segmentation, character recognition, and contextual postprocessing, as in Baird and Thompson (1990). This approach is appropriate for isolated characters, abbreviations, or well-printed text. However, character recognition is not very successful in domains with many degraded images and large variations in font style. It is observed that character segmentation is difficult for degraded images such as those shown in Figure 1(a). Premature recognition decisions on character identities may also create irrecoverable errors for faint images such as those in Figure 1(b).

In a recently proposed approach, a word is treated as a whole unit without being segmented into individual characters (Hull (1987)). This is referred to as the *word shape analysis* approach. This method avoids committing errors in character segmentation and premature character recognition. Previous publications have analyzed theoretical aspects of the technique and projected performance in limited domains. In this paper, an algorithm for word shape analysis is presented that is suitable for degraded word images of multiple font types, such as those encountered in a postal optical character reader (OCR). An implementation is also demonstrated with exhaustive experiments on images scanned from a postal OCR.

Correspondence to: T.K. Ho, Center for Document Analysis and Recognition, State University of New York at Buffalo, 226 Bell Hall, Buffalo, NY 14260, USA.

This work was supported by the Office of Advanced Technology of the United States Postal Service.

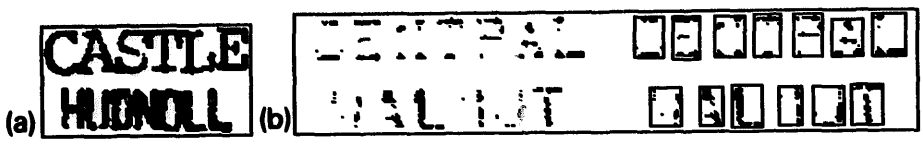

Figure 1. (a) Word images difficult to segment, and (b) images with segmented characters that are difficult to recognize (words in the images: CENTRAL, WALNUT).

The algorithm is intended to be an alternative word recognition technique which will supplement known approaches based solely on character recognition. As suggested in Ho et al. (1990), a robust recognition system can be constructed by combining decisions from a set of independent classifiers, which can perform better than any of the individual classifiers. The advantage of this algorithm is therefore twofold: while it is an effective technique on its own, as an independent method, it can be combined with a character recognition technique to achieve a robust system with further improved performance. An example of such an integrated system is suggested in Ho et al. (1991).

The algorithm assumes that an input word image is binarized. The rest of this paper presents the algorithm in detail. Specific steps of the method are outlined and an implementation is discussed. Experimental results are presented that validate the methodology and demonstrate its usefulness.

2. Algorithm statement

The basic word shape analysis algorithm is outlined in Figure 2. The word shape analysis approach attempts to describe and compare the shape of the word as a whole object. Every word is first partitioned into a fixed 4×10 grid to provide a frame of reference. Features that describe the details of the word shape are then extracted and their relative locations in the grid are recorded in a feature vector. The feature vector is matched to a lexicon of words and a ranking is produced. Words ranked near the top by the system are referred to as the *neighborhood* of the word in the image. These are the words that share similar shape characteristics as determined by the system. According to the performance of the system, one can determine the size of the neighborhood such that the true word will not be missed. Such a neighborhood can be treated as a set of hypotheses that are subject to further hypothesis testing or selection in accordance with other contextual constraints (Hull (1989)).

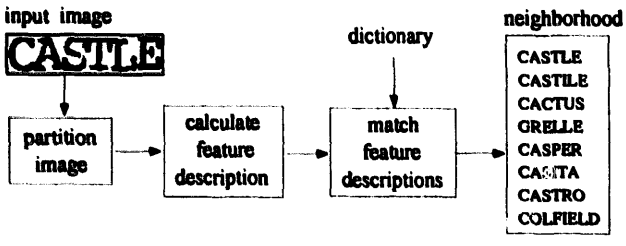

Figure 2. Outline of word shape analysis algorithm.

Subsequent sections of this paper discuss each step of the algorithm in detail.

3. Word image area partitioning

Word shape analysis is different from traditional segmentation-based methods in that a global reference frame defined on a word image is used to represent the locations of shape features.

The global reference frame consists of four reference lines including the image upper boundary, the top line, the base line, and the image lower boundary. The image upper and lower boundaries are the upper and lower edges of the most succinct rectangle that contains all the black pixels in the image. The top line is a line formed by the top of the lower case characters that do not have an ascender nor a dot ({a, c, e, g, m, n, o, p, q, r, s, u, v, w, x, y, z}). The base line is a line formed by the bottom of all the upper case characters as well as the lower case characters that do not have a descender ({a, b, c, d, e, h, i, k, l, m, n, o, r, s, t, u, v, w, x}).

These four reference lines are detectable in most

words consisting of mixed case characters. The four lines divide those images into three vertical regions: the ascender region, the middle region, and the descender region. Because many characters locate entirely in the middle region, that region is further divided into upper and lower parts to facilitate more accurate position description. The image is divided into ten equal-sized regions along the horizontal axis. As a result, the image area is partitioned into 4 vertical regions, and 10 horizontal regions, i.e., 40 cells.

The top line cannot be detected in words consisting of purely upper case characters, as well as in words that do not contain short characters, such as the word 'Hill'. In such cases the top line will be made identical to the upper boundary. The base line may also be identified with the lower boundary for images without descenders. Therefore some images may have no ascender or descender region. The middle region divisions and the ten horizontal divisions are unaffected. For convenience in implementation, the image is still considered to be in 40 cells, though 10 or 20 of them are empty.

To avoid variations in gap widths between neighboring characters, the word image is first preprocessed by reducing all detected white gaps to one pixel wide. This normalization procedure is useful when the technique is applied in a multi-font environment. Though, it has to be applied with caution to faint images. For instance, the compression does not affect much of the shape of the word 'CENTRAL' in Figure 1(b), but causes problem to the word 'WALNUT' in the same figure. Therefore this procedure should be omitted if most of the images encountered in an application are vulnerable to the compression.

The top line and the base line are then estimated by the following method. A word image is first smeared horizontally. The upper and lower contours of the smeared image are then computed. A histogram is computed for the heights from the image upper boundary to the upper contour. The mode of these heights is taken as the location of the top line. The base line is determined by finding the mode in the histogram of heights from the lower contour to the image lower boundary.

Figure 3 shows the results of base line location and the 40 area partitions.

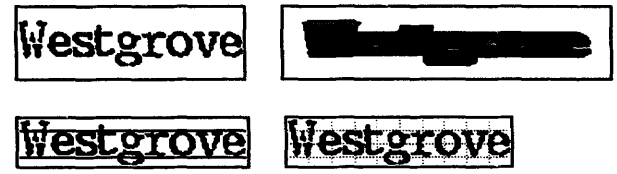

Figure 3. Baseline calculation and area partitions of an image.

4. Word shape feature extraction

A set of features that is referred to as the *stroke direction distribution* is used to describe the shape of a word. It captures the spatial distribution of black pixels belonging to strokes of various directions.

The features are extracted using the *local direction contribution* method suggested for use with isolated Chinese characters in Mori et al. (1984). At each black pixel in the image, the longest continuous run of black pixels in each of the four directions east-west, northeast-southwest, north-south, and northwest-southeast is computed. The pixel is labeled with the direction in which the *run length* is a maximum. That is, each black pixel is labeled as part of a stroke of one of the four directions. Figure 4(a)-(e) shows an example of such pixel labeling.

For each of the 40 cells in the image area, the labeled black pixels of each type in that area are counted. The counts are then normalized by the total number of black pixels in the image. The stroke direction distribution is represented by a 160-dimensional feature vector, which stores the normalized counts of black pixels of each of the four types in the 40 cells.

The calculation of the feature vector is summarized using pseudo C code in Figure 5. Each pixel of an input image is inspected. If it is black, the function *Max_run_direction* is called. This function returns a value between 0 and 3 (for E-W, NE-SW, N-S, or NW-SE, respectively) that indicates the direction of the run with the maximum length at the current location in the image. The loop indices i and j that correspond to row i and column j in the image are then normalized to the 4-by-10 grid, using the function *Vertical_region* that looks up the vertical region i belongs to according to the pre-calculated locations of the reference lines. The

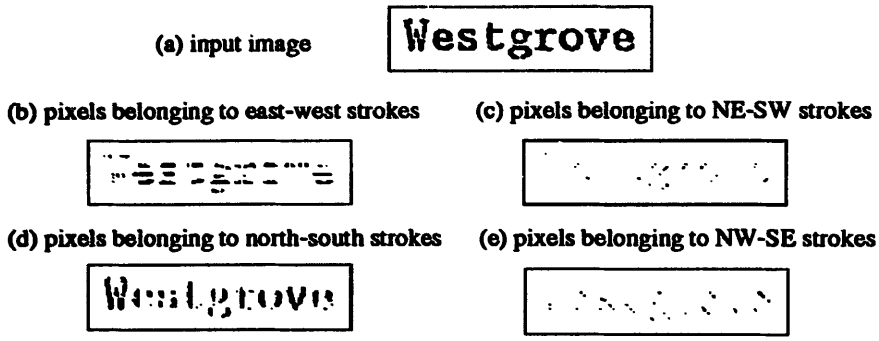

Figure 4. An example of stroke direction distribution.

location in *feature_vector* that corresponds to the run of maximum length is then incremented. Each location in *feature_vector* is then normalized by dividing it by the number of pixels in the image.

5. Prototype matching and classification

Models or feature prototypes for the words in the lexicon are needed to match the feature vector computed from the input image. It is impractical to collect a large set of image samples for each word in the lexicon, therefore we adopt an alternative strategy, that is, to synthesize the words from a set of *character* samples of various fonts. The features of the synthesized words are then extracted and stored as prototypes.

The city-block distance (Duda and Hart (1973)) is used to compare a feature vector of an input word and that of a word in a given lexicon. The distance is defined to be the sum of the absolute differences of corresponding feature components. That is, if $x = \langle x_1, x_2, \ldots, x_n \rangle$ and $y = \langle y_1, y_2, \ldots, y_n \rangle$ are two feature vectors, then the distance is

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

where $n = 160$ for the stroke direction distribution vector.

For an input image, the computed feature vector is matched to the vectors for each word in the lexicon that are synthesized with the font samples. If there are M words in the lexicon, and N font samples are used to synthesize the words, then NM distance computations are needed for each input image. For each word in the lexicon, the minimum distance among the N distances for the N fonts is determined. The M words in the lexicon are then

```
pixel_count = 0;

for (i=0; i < rows; ++i)
  for (j=0; j < cols; ++j)
    if (image[i][j] == BLACK) {
       run_label = Max_run_direction (image,i,j);
       row_index = Vertical_region (i);
       col_index = j/cols*10;
       ++feature_vector[row_index*40 + col_index*4 + run_label];
       ++pixel_count;
    }

for (i=0; i < 160; ++i)
   feature_vector[i] = feature_vector[i] / pixel_count;
```

Figure 5. Algorithm to calculate the stroke direction feature vector.

arranged in order of increasing distances. This is the order of similarity computed by the algorithm. The top choice is the word with a feature vector of minimum distance to the feature vector computed from the input image.

6. Alternative feature sets

A similar word shape analysis technique can be developed using other feature sets. One example of alternative feature sets is a set of template defined features proposed in Baird et al. (1989), which is defined by 32 feature templates, each of size 7×7. The features defined by these templates are detected by convolution and thresholding. Each nonzero response after thresholding represents that a feature of a particular type is detected at that pixel position. The outputs are described by a 1280-dimensional feature vector, which stores counts of the 32 features detected in the 40 cells. The same distance metric and classification procedure can be applied to these feature vectors.

7. Experimental results

Experiments were performed using a collection of images of *machine-printed* postal words obtained from live mail. They were scanned at 212 pixels per inch and binarized. The font and quality of the images vary. The image database was divided into separate training sets and testing sets. The stroke direction computation routines were developed and modified by observing performance on a small initial training set of about 200 images.

The classifier was applied to a test set of 1671 word images, using a lexicon of 500 words which contains all the true words in the images. Prototypes for words in input lexicons were synthesized using 10 and 77 font samples in two separate tests. Tests were also conducted on several random subsets of the lexicon. Similar tests were performed using the Baird template features. Table 1 summarizes the recognition performance measured as the percentage correct in a number of top choices in the output ranking. Table 2 summarizes the performance using the Baird templates as the feature set. It can be observed that the performance with stroke direction features is more sensitive to the font sample size than those with Baird features.

While satisfactory results are obtained using each of the two feature sets, it should be noted that either of the two sets is not necessary a replacement of the other. Instead, in a multi-classifier system as suggested in Ho et al. (1990, 1991), they can be used in two separate classifiers in parallel, and their decisions can be combined and further improved.

As seen from Table 1, using a lexicon of 500 words, a correct rate of 86.3% at the top choice and 90% in the top two choices was achieved. The correct rate is 94.1% in the top 10 choices. That is, the technique can be applied to reduce a lexicon of 500 words to a neighborhood of 10 words with a 94.1% accuracy. This is significant in an application where other recognition techniques can be applied to such a neighborhood to uniquely determine the identity of the input word. Useful

Table 1
Performance on test set with stroke direction features

Lexicon size	Number of font samples	% Correct at top N decisions, $N=$ 1	2	3	4	5	10	20	30
500	10	77.3	86.1	88.3	89.2	89.8	92.2	94.1	95.2
500	77	86.3	90.0	91.9	92.8	93.4	94.1	95.2	95.8
200	10	84.5	89.2	91.1	92.0	92.8	94.6	96.2	96.6
200	77	90.3	92.6	93.7	94.3	94.3	95.4	96.5	96.9
100	10	87.9	91.6	93.1	93.8	94.6	96.2	97.0	98.3
100	77	92.2	93.7	94.5	94.8	95.1	96.5	97.3	98.0
50	10	90.4	93.1	94.7	95.5	96.1	97.3	98.6	99.1
50	77	93.4	94.7	95.4	95.8	96.3	97.1	98.7	99.2

Table 2
Performance on test set with Baird template features

Lexicon size	Number of font samples	% Correct at top N decisions, N= 1	2	3	4	5	10	20	30
500	10	78.4	83.2	84.6	85.6	86.4	88.5	90.5	91.7
500	57	80.8	84.7	85.9	86.7	87.4	88.9	90.8	91.8
200	10	83.0	86.0	87.6	88.5	89.3	91.5	93.7	94.6
200	57	84.4	87.1	88.3	89.3	89.9	91.6	93.4	94.5
100	10	84.0	87.4	88.5	89.4	90.4	92.7	94.8	95.9
100	57	85.4	87.8	89.0	89.9	90.5	92.5	94.6	96.1
50	10	85.6	88.8	89.9	91.0	92.2	94.4	97.0	98.0
50	57	87.3	89.2	90.9	91.8	92.4	94.8	96.5	97.4

knowledge sources include higher level contextual constraints such as the syntactic categories in a running text. Such constraints can be applied to reduce the size of the lexicon and improve the performance significantly. To illustrate this, random subsets of sizes 200, 100, and 50 are extracted from the 500 word lexicon. The results show that a correct rate of 93.4% at the top choice can be achieved with a lexicon of 50 words.

8. Conclusions and future work

A methodology for word recognition was presented that is based on word shape analysis without character segmentation and recognition. This method is used in a fixed vocabulary word recognition system and outputs a ranking of a given lexicon, which specifies the order that the words are determined to be similar in shape to the input image. The objective of the method is to insure that a small number of words at the top of the ranking contain the word in the image. This technique was tested using word images segmented from address block images captured on a postal OCR. Experimentation is discussed where lexicons of various sizes are used. Robust performance was achieved on images that would, in many cases, be difficult for an alternative approach to recognize correctly.

Word shape analysis avoids committing errors in character segmentation and premature character recognition. Though, there are cases that isolated character information is useful such as in abbreviations and well-printed text. Therefore it is suggested this method be used in combination with a character recognition based technique to achieve maximum performance. A multiple classifier methodology can be applied to construct a robust word recognition algorithm, with both word shape analysis and character recognition methods as its components.

References

Baird, H.S., H.P. Graf, L.D. Jackel and W.E. Hubbard (1989). A VLSI architecture for binary image classification. In: J.C. Simon, Ed., *From Pixels to Features*. North-Holland, Amsterdam, 275-286.

Baird, H.S. and K. Thompson (1990). Reading chess. *IEEE Trans. Pattern Anal. Machine Intell.* 12 (6), 552-559.

Duda, R.O. and P.E. Hart (1973). *Pattern Classification and Scene Analysis*. Addison-Wesley, New York.

Ho, T.K., J.J. Hull and S.N. Srihari (1990). Combination of structural classifiers. *Pre-Proc. IAPR Syntactic and Structural Pattern Recognition Workshop*, New Jersey, June 13-15, 1990, 123-136.

Ho, T.K., J.J. Hull and S.N. Srihari (1991). Word recognition with multi-level contextual knowledge. *Proc. First Int. Conf. on Document Analysis and Recognition*, Saint-Malo, France, 1991, 905-915.

Hull, J.J. (1987). A computational theory and algorithm for fluent reading. *Proc. Third IEEE Conf. on Artificial Intelligence Applications*, Kissimmee, FL, Feb. 23-27, 1987, 176-181.

Hull, J.J. (1989). Feature selection and language syntax in text recognition. In: J.C. Simon, Ed., *From Pixels to Features*. North-Holland, Amsterdam, 249-260.

Mori, S., K. Yamamoto and M. Yasuda (1984). Research on machine recognition of handprinted characters. *IEEE Trans. Pattern Anal. Machine Intell.* 6 (4), 386-405.