Structural Features for Recognizing Degraded Printed Gurmukhi Script

M. K. Jindal
Department of Computer Applications, P. U. Regional Centre, Muktsar(Punjab), India
manishphd@rediffmail.com

R. K. Sharma
School of Mathematics & Computer Applications, Thapar University, Patiala(Punjab), India
rksharma@tiet.ac.in

G. S. Lehal
Department of Computer Science Punjabi University, Patiala(Punjab), India
gslehal@gmail.com

Abstract

The performance of an OCR system depends upon printing quality of the input document. Many OCRs have been designed which correctly identify fine printed documents in Indian and other scripts. But, little reported work has been found on the recognition of the degraded documents. The performance of any standard OCR system working for fine printed documents decreases, if it is tested on degraded documents. Feature extraction is an important task for designing an OCR for recognizing degraded documents. In this paper, we have discussed efficient structural features selected for recognizing degraded printed Gurmukhi script characters.

1. Introduction

An OCR system for recognizing high quality machine-printed text can recognize words at a high level of accuracy [1]. However, given a degraded text page, performance usually drops significantly. There are different kinds of degradations [2] available in almost every script of the world. Touching characters and heavy printed characters are most commonly found degradations in printed Gurmukhi script.

In machine printed documents, shape discrepancy among characters belonging to same class is sometimes quite large because of the degradations of the document images. Particularly, when touching characters are segmented, the noise blobs near the cutting points overlap both sides of the characters, possibly resulting in a large dissimilarity between the input pattern and the corresponding sample class. Therefore, it is required to select features, which can adapt the shape variations due to touching noise blobs. In fact the main problem in OCR system is the large variation in shapes within a class of character. This variation depends from font styles, document noise, photometric effect, document skew and poor image quality. The large variation in shapes makes it difficult to determine the number of features that are convenient prior to model building. The performance of a character recognition system depends heavily on what features are being used. Though many kinds of features have been developed and their test performances on standard database have been reported, there is still room to improve the recognition rate by developing an improved feature.

Trier *et al*. [3] Summarized and compared some of the well-known feature extraction methods for off-line character recognition. Selection of a feature extraction method is probably the single most important factor in achieving high recognition performance in character recognition systems. They discussed feature extraction methods in terms of invariance properties, reconstructability and expected distortions and variability of the characters. Suen [4] and Impedovo *et al*. [5] agree that feature extraction plays an important role in the successful recognition of machine-printed and handwritten characters.

Structural features describe a pattern in terms of its topology and geometry by giving its global and local properties. Some of the main structural features include features like number and intersections between the character and straight lines, holes and concave arcs, number and position of end points and junctions [3]. These features are generally hand crafted by the researchers for the kind of pattern to be classified. Chinese characters contain rich structural information, which remains unchanged over font and size variation. Since the basic elements of a Chinese character are strokes, the types and numbers of strokes and relationships among the strokes are essential structural features of a Chinese character [6].

Amin [7] has used seven types of structural features for recognition of printed Arabic text. Lee and Gomes [8] have used structural features for handwritten numeral recognition. Rocha and Pavlidis [9] have proposed a method for the recognition of multifont printed characters using structural features like convex arcs and strokes, singular points and their

978-0-7695-3099-4/08 $25.00 © 2008 IEEE
DOI 10.1109/ITNG.2008.223

relationships. In a classic paper, Kahan *et al.* [10] have developed a structural feature set for recognition of printed text of any font and size. The feature set includes the following information for a character: number of holes, location of holes, concavities in the skeletal structure, crossings of strokes, endpoints in the vertical direction and bounding box of the character. Leedham and Vladimir [11] have used global features and local features for recognizing handwritten text.

2. Problem of degraded text recognition

Touching characters and heavy printed characters are most commonly found degradations in printed Gurmukhi script [2]. Characteristics of Gurmukhi script can be found in [2, 12]. In case of touching characters, two neighboring characters touch each other. The biggest issue involved in recognition of touching characters is to segment them correctly, i.e. identifying the position at which the touching pair of characters must be segmented. Every OCR must perform well to the sensitive task of separating the touching characters. The accuracy of any OCR depends heavily upon the accuracy of segmentation process. We have designed efficient algorithms for segmenting touching charcters in Gurmukhi script [12]. Figure 1 contains example words of Gurmukhi script containing touching characters.

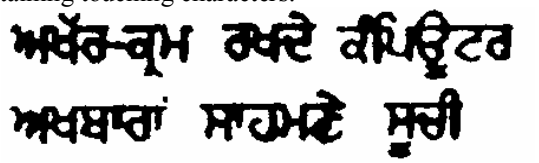

Figure 1. Words containing touching characters in printed Gurmukhi script.

Sometimes even if the characters that are easily isolated, heavy print can distort their shapes, making them unidentifiable. It is very difficult to recognize a heavily printed character. Figure 2 consists of some of the heavy printed characters in Gurmukhi script.

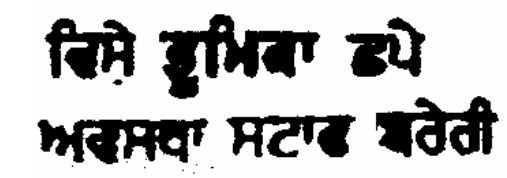

Figure 2. Heavy printed characters in Gurmukhi script.

Magazines with heavy printing, newspapers printed on low quality paper, very old books whose pages turn to be yellow due to aging, Photostatted documents copied on low quality machine etc. are few sources of producing documents containing touching characters and heavy printed charcters. We have studied in detail the characteristic of various degradations in printed Gurmukhi script [2]. Touching charcters and heavy printed charcters are normally found in other kinds of degradations such as faxed documents, typewritten documents, backside printing visible documents etc.

3. Structural features for recognizing degraded Gurmukhi text

It was not an easy task to decide which structural features should be chosen to extract the structural features from degraded characters of Gurmukhi script due to large shape variations in characters of same class. Feature codes of the structural features set have following common characteristics.

- These structural features are less sensitive to character size and font.
- The feature codes present a very high separability for different characters. In other words, the feature codes representing different characters have a very low probability to coincide.
- These features are very much tolerant to noise.

We have used following structural features:

1. Presence of Sidebar (St1): This feature is present if a vertical sidebar, of approximate the same height as of the character (sub-symbol), is present at the rightmost side of the character. If full sidebar exists in Gurmukhi characters, it is always at right end of the Gurmukhi character. There are 11 characters in middle zone having full sidebar at their right end. These characters are: a, s, k, G, j, W, Y, p, b, m, y. Additionally, second component '.' of the multicomponent character g has full sidebar at right end . There are two vowels I , i , whose one stroke falls in middle zone having full sidebar at their right end. Also, four characters have quarter sidebars at their right end. These characters are: r, h, C and first component 'r' of the multicomponent character g. These characters have also been considered for this feature. As such, there are total 18 sub-symbols, containing this feature. This feature divides the whole set of Gurmukhi characters in middle zone in almost two equal sized subsets. This feature is true if a vertical sidebar is present at rightmost side of the sub-symbol else it is false.

2. Presence of half Sidebar (St2): This feature is present if a sidebar, of approximately half the height of middle zone is present at the rightmost side of the character. There are 8 characters in Gurmukhi script having this feature: e, x, M, t, Q, d, f, v.

In addition to these, one vowel ‘ A ’ also contains this feature.

3. Presence of headline (St3): The presence of headline in the sub-symbol is another important feature for classification. For example, p has no headline while t has headline.Even when the characters are highly degraded, this feature is retained. There are 30 characters in middle zone having this feature present: u, e, s, h, c, g, L, C, x, j, J, M, t, T, D, Q, N, V, W, d, Y, n, f, b, B, y, r, l, v, R. Furthermore, three vowels I, i and A have this feature true. This feature is very much robust to the noise. This feature is extremely useful for differentiating similar characters such as s and m, k and W, Y and p.

4. Number of junctions with headline (St4): It can be noted that each character in middle zone of Gurmukhi character set has either one (true) or more (false) than one junctions with the headline. For example r has one junction with headline while y has two junctions with headline. There are 19 characters having this feature true: h, c, L, C, x, j, t, T, D, Q, N, V, d, n, f, B, r, v, R. Additionally, this feature is true for both the components of g. Moreover, this feature is also true for vowels I, i. On the contrary, this feature is false for u, a, e, s, k, G, J, M, W, Y, p, b, m, y and l. As shown in Figure 3, sometimes due to heavy printing of the characters c, C, D, Q, d, v, the feature value becomes false instead of true.

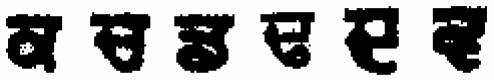

Figure 3. Degraded Gurmukhi characters having feature value of St4 false instead of true.

5. Number of junctions with the baseline (St5): This feature is true for a sub-symbol if number of junctions with the base line is one else it is false. This feature is true for sub-symbol r and false for sub-symbol j since it has two junctions with the baseline. Further, single vowel in middle zone A has the value of this feature false. There are 26 sub-symbols in Gurmukhi characters set having this feature true: u, e, h, c, L, C, x, J, M, t, T, D, Q, N, V, W, d, Y, p, f, b, B, r, v, R. The feature is false for a, s, G, j, n, m, y, l characters.

6. Aspect ratio (St6): Aspect ratio is obtained by dividing the height of the middle zone by the width of the character. We have divided the whole sub-symbols in middle zone into three categories depending upon the aspect ratio of the sub-symbols. For wider characters a and G aspect ratio is less then 0.90, giving St6=0. Also, the aspect ratio for I, i, A is greater than 3.0. So, St6=2 for these three vowels. For all other characters in middle zone the value of St6 is considered as 1.

7. **Left, Right, Top and Bottom Profile Direction Codes (St7, St8, St9, St10)**: A variation of chain encoding is used on left, right, top and bottom profiles. For finding the left profile direction codes, the left profile of a sub-symbol is scanned from top to bottom and local directions of the profile at each pixel are noted. Starting from current pixel, the pixel distance of the next pixel in east, south or west directions is noted. The cumulative count of movement in three directions is represented by the percentage occurrences with respect to the total number of pixel movement and stored as a 3 component vector with the three components representing the distance covered in east, south and west directions, respectively. The direction code of the profile of Figure 4 is {30, 20, 50}, since the movements in east, south and west directions are 3, 2 and 5 pixels, respectively. While calculating the direction code of the profile for Figure 4, moving from row number 1 to row number 10, the distance covered in pixels is row 1 $\rightarrow$ row 2(south 1) $\rightarrow$ row 3(west 1) $\rightarrow$ row 4(west 2) $\rightarrow$ row 5(west 1) $\rightarrow$ row 6(west 1) $\rightarrow$ row 7(south 1) $\rightarrow$ row 8(east 1) $\rightarrow$ row 9(east 1) $\rightarrow$ row 10(east 1). Similarly, right profile direction codes are found by scanning right profile from top to bottom and movement is noted in east, south and west directions. Furthermore, for finding the direction code of the top and bottom profiles, east, south and north directions are considered while moving from left to right. As such, a total of twelve (4×3) structural features are obtained using this feature. This feature gives the movements of the strokes along the external boundaries of the sub-symbols.

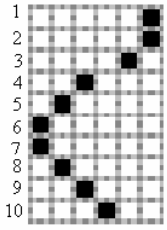

Figure 4. Projection profile of a Gurmukhi sub-symbol.

8. Directional Distance Distribution (St11): Directional Distance Distribution (DDD) is a distance based feature. For every pixel in the input binary array, two sets of 8 bytes which are called W (White) set and B (Black) set are allocated as shown in Figure 5. For a white pixel, the set W is used to encode the distances

to the nearest black pixels in 8 directions (0^0, 45^0, 90^0, 135^0, 180^0, 225^0, 270^0, 315^0). The set B is simply filled with value zero. Similarly, for a black pixel, the set B is used to encode the distances to the nearest white pixels in 8 directions. The set W is filled with zeros. In Figure 5, the color of pixel at coordinates (6, 6) is white. For the direction 0^0, the traveled sequence is: (6, 6)W $\rightarrow$ (6, 7)W $\rightarrow$ (6, 8)W $\rightarrow$ (6, 9)B. The traveled distance 3 is recorded for 0^0. Sometimes, we are encountered with boundary of the array without finding the black pixel. At this stage, array is supposed to be circular. Therefore, while finding the nearest black pixel in 225^0 direction, the following travel sequence will be followed: (6, 6)W $\rightarrow$ (7, 5)W $\rightarrow$ (8, 4)W $\rightarrow$ (9, 3)W $\rightarrow$ (10, 2)W $\rightarrow$ (11, 1)W $\rightarrow$ (1, 11)W $\rightarrow$ (2, 10)W $\rightarrow$ (3, 9)W $\rightarrow$ (4, 8)W $\rightarrow$ (5, 7)B, and the traveled distance 10 is recorded. The set B is simply filled with zeros as shown in Figure 6(a). Similarly, for the black pixel at coordinates (4, 6), the directional distance values have been shown in Figure 6(b). After computing WB encoding for each of the pixel, we have divided the input array into four equal zones both horizontally and vertically, hence producing 16 zones. We have taken the average of WB encoding in each of these 16 zones. Finally, we got a 16 × 16 feature vector.

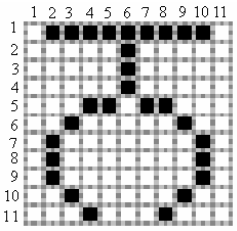

Figure 5. Projection profile of a Gurmukhi Character.

w 0	w 1	w 2	w 3	w 4	w 5	w 6	w 7	b 0	b 1	b 2	b 3	b 4	b 5	b 6	b 7
3	1	2	1	3	10	6	11	0	0	0	0	0	0	0	0

(a)

w 0	w 1	w 2	w 3	w 4	w 5	w 6	w 7	b 0	b 1	b 2	b 3	b 4	b 5	b 6	b 7
0	0	0	0	0	0	0	0	1	1	4	1	1	2	1	2

(b)

Figure 6. Example of WB encoding: (a) WB encoding for the white pixel at (6, 6), (b) WB encoding for the black pixel at (4, 6).

9. Transition features (St12): In this structural feature, location and number of transitions from background to foreground pixels in the vertical and horizontal directions are noted. The transition feature used here is similar to that proposed by Gader *et al.*[13] To calculate transition information, image is scanned from left-to-right, right-to-left, top-to-bottom and bottom-to-top. To ensure a uniform feature vector size, the transitions in each direction is computed as a fraction of the distance traversed across the image. For example, if the transitions were being computed from top-to-bottom, a transition found close to the top would be assigned a high value compared to a transition computed further down. A maximum value (M) was defined to be the maximum number of transitions that may be recorded in each direction. Conversely, if there were less than M transitions recorded (n for example), then the remaining M - n transitions would be assigned values of 0 (to aid in the formation of uniform vectors). Therefore, for a character matrix of size 50 × 50, left to right transitions will produce 50 × 5=250 (for M=5) values of transitions. We have taken average (A) of these transactions and taken a size of 5, 7 or 10 rows for average. If size for A is taken 10, then number of feature values reduces to 25.

The above-mentioned features have been used with different options. Also following additional assumptions have been proposed to enhance the accuracy of detection of the structural features:

1. If St1 is true for any sub-symbol, then St2 is false, *i.e.*, if full sidebar (St1) is detected, then half sidebar (St2) is absent.
2. If St1 is false, then St6 is not equal to 1, *i.e.*, if full sidebar is not detected, then it can not be wide character having low aspect ratio (for which St6 = 1).
3. If St3 is false, then St4 is also false, *i.e.*, if character has no headline, then the number of junctions with headline is not one.
4. If St6 = 3 and St5 is false, then the character is `A` as only this single vowel has high aspect ratio (St6=3) and number of junctions with the baseline is not one.
5. If St6 = 3, then St7 = St8 = St9 = St10 = {0, 0, 0}.
6. If St1 is true, then St7 = {0, 0, 0}, *i.e.*, if full sidebar is present then left profile chain code is 0.
7. If headline is absent for a sub-symbol, only then aspect ratio = 1, *i.e.*, if St3 is false and aspect ratio is less than 0.90 only then St6 = 1. As aspect ratio is less than 0.90 (St6 = 1) for only two characters `a` and `G`. However, most of the times, sub-symbol `y` or a touching pair of sub-symbols has aspect ratio < 0.90 and the feature St6 is evaluated wrongly for such sub-symbols. Generally, `y` or a touching pair of sub-symbols having aspect ratio < 0.90 would have headline. However, as discussed if St6 = 1, headline must be absent. As a result of this, the wrongly calculated value of St6 can be corrected.

4. Database

We have scanned documents at 300 dpi from newspapers, magazines, books *etc*. to create a large set of database consisting of printed degraded Gurmukhi documents. Each document in the database consists of touching characters and heavily printed characters. We have used our segmentation algorithms [12] to segment the touching characters. After applying segmentation methods, we obtain individual charcters. Thus we have constructed a large database consisting of segmented degraded printed Gurmukhi characters. Few Gurmukhi characters from the database have been shown in Figure 7 (training purpose) and Figure 8 (testing purpose). One can see from these figures the large variability in shapes belonging to same class.

Figure 7. A sample of degraded printed Gurmukhi characters taken for training purpose.

5. Experimental results

For classification purpose we have used *k*-NN, SVM classifiers. Both the classifiers have been used using MATLAB 7.2. The results of *k*-NN using various kinds of structural features and their various options have been shown in Table 1. Table 1 shows the error rate using *k*-NN with different values of *k* on different sets of structural features. From Table 1, it is observed that we get maximum recognition accuracy of 83.60% at $k = 1$ when all the structural features have been used. It is also observed from the Table 1 that individually each structural feature does not perform that well, but when used in combination, these structural features produces more accurate results.

Figure 8. A sample of degraded printed Gurmukhi characters taken for testing purpose.

Table 1. Recognition accuracy using *k*-NN with different values of *k* on different sets of structural features.

Feature used	Length of feature vector	*k*=1	*k*=3	*k*=5
St1-St10	18	18.90	19.90	20.90
St11	256	69.65	70.15	68.66
St11	128(odd)	58.71	60.20	58.71
St11	128(even)	60.20	62.19	62.69
St12	200 ($M = 5$, $A = 5$)	73.63	74.63	70.15
St12	140($M = 5$, $A = 7$)	66.67	69.15	65.67
St12	100($M = 5$, $A = 10$)	66.17	66.67	66.67
St12	160 ($M = 4$, $A = 5$)	74.13	74.13	70.15
St12	112($M = 4$, $A = 7$)	67.16	69.15	65.67
St12	80($M = 4$, $A = 10$)	66.67	66.67	66.67
St12	120 ($M = 3$, $A = 5$)	73.63	72.64	70.65
St12	84($M = 3$, $A = 7$)	68.66	69.15	64.68
St12	60($M = 3$, $A = 10$)	65.67	65.67	67.16
Combined structural	416 [256(St11) + 160 (St12)]	80.60	80.10	77.11
Combined structural	376 [256(St11) + 120 (St12)]	81.59	80.10	78.11
Combined structural	316 [256(St11) + 60 (St12)]	78.61	78.11	79.60
Combined structural	248 [128(St11) + 120 (St12)]	79.10	79.60	76.12
Combined structural	188 [128(S11) + 60 (St12)]	79.10	79.10	77.11
Combined structural	206 [18(St1-St10) + 128 (St11-even) + 60(St12)]	80.60	80.60	79.10
Combined structural	264 [18(St1-St10) + 128(St11-even) + 120(St12)]	82.59	81.60	81.09
Combined structural	394 [18(St1-St10) + 256(St11) + 120(St12)]	**83.60**	82.60	81.59

Similarly, the results of SVM classifier using various kinds of structural features and their various options have been shown in Table 2. It is observed that we get maximum accuracy of 91.54% using SVM Classifier.

Table 2. Recognition accuracy using SVM on different sets of structural features.

Feature used	Length of feature vector	Percentage Accuracy
St1-St10	18	24.05
St11	256	84.08
St11	128 (odd)	77.61
St11	128 (even)	79.10
St12	200 (M = 5, A = 5)	82.59
St12	140(M =5, A = 7)	73.13
St12	100(M = 5, A = 10)	64.68
St12	160 (M = 4, A = 5)	83.08
St12	112(M = 4, A = 7)	76.12
St12	80(M = 4, A = 10)	67.16
St12	120 (M =3, A = 5)	81.59
St12	84(M = 3, A = 7)	76.12
St12	60(M = 3, A = 10)	61.19
Combined structural	416[256(St11) + 160 (St12)]	88.54
Combined structural	376[256(St11) + 120 (St12)]	89.55
Combined structural	316 [256(St11) + 60 (St12)]	91.54
Combined structural	248[128(St11) + 120 (St12)]	90.54
Combined structural	188[128(St11) + 60 (St12)]	91.04
Combined structural	334[18(St1-St10) + 256 (St11) + 120(St12)]	90.05
Combined structural	206[18(St1-St10) + 128 (St11) + 60(St12)]	91.04

6. Conclusions

We have discussed various structural features used for recognizing degraded printed Gurmukhi script documents containing touching characters and heavy printed characters. St1-St6 features are based on the structural properties of the script. Structural features St7-St10 are direction codes based on left, right, top and bottom profiles. These features are invariant to noise. Structural feature St11 has shown very effective results for recognizing degraded documents. Three different options of this feature for all eight directions, 4 even and 4 odd directions have been discussed. Last structural feature St12 have been used for three different transitions (M = 3, 4, 5) and three different averages (A = 5, 7, 10). Along with showing the results of individual features with individual options, various combinations of the features have been implemented and results have been shown. It is shown that the best result have been found using 316 features containing 256 features of St11 and 60 features of St12 using SVM classifier.

7. References

[1] R. G. Casey and E. Lecolinet. "A Survey of Methods and Strategies in Character Segmentation", *IEEE Transactions on PAMI*, Vol. 18(7), pp. 690-706, July 1996.

[2] M. K. Jindal, R. K. Sharma and G. S. Lehal, "A Study of Different Kinds of Degradation in Printed Gurmukhi Script", in Proceedings of the IEEE International Conference on Computing: Theory and Applications (ICCTA'07), pp. 538-544, Published by IEEE Computer Society USA, March 2007.

[3] O. D. Trier, A. K. Jain and T. Taxt, "Feature extraction methods for character recognition: – A survey", *Pattern Recognition,* Vol. 29(4), pp. 641-662, 1996.

[4] C. Y. Suen, "Character Recognition by Computer and Applications", in Handbook of Pattern Recognition and Image Processing, New York: Academic pp. 569-586, 1986.

[5] S. Impedovo, L. Ottaviano and S. Occhinegro, "Optical Character Recognition - A Survey", *International Journal of Pattern Recognition & Artificial Intelligence*, Vol. 5, pp. 1-24, 1991.

[6] K. W. Gan and K.T. Lua, "A new approach to stroke and feature point extraction in chinese character recognition", *Pattern Recognition Letters,* Vol. 12(6), pp. 381-387, 1991.

[7] A. Amin, "Recognition of printed Arabic text based on global features and decision tree learning techniques", *Pattern Recognition,* Vol. 33, pp. 1309-1323, 2000.

[8] L. L. Lee and N. R. Gomes, "Disconnected handwritten numeral image recognition", in the Proceedings of Fourth International Conference on Document Analysis and Recognition (ICDAR'97), pp. 467-470, 1997.

[9] J. Rocha and T. Pavlidis, "A shape analysis model with applications to a character recognition system", *IEEE Transactions on PAMI,* Vol. 16, pp. 393-404, 1994.

[10] S. Kahan, T. Pavlidis and H. S. Baird, "On the Recognition of Printed Characters of Any Font and Size", *IEEE Transactions on PAMI*, Vol. 9(2), pp. 274-288, 1987.

[11] Graham Leedham and Vladimir Pervouchine, "Validating the use of Handwriting as a Biometric and its Forensic Analysis", in the Proceedings of International Workshop on Document Analysis (IWDA'05), pp. 175-192, 2005

[12] M. K. Jindal, G. S. Lehal and R. K. Sharma, "Segmentation Problems and Solutions in Printed Degraded Gurmukhi Script", *International Journal of Signal Processing*, Vol. 2(4), pp. 258-267, 2005.

[13] P. D. Gadar, M. Mohamed, and J. H. Chiang, "Handwritten Word Recognition with Character and Inter-Character Neural Networks", *IEEE Transactions on Systems, Man, and Cybernetics-Part B:Cybernetics*, Vol. 27(1), pp. 158-164, 1997.