

Recognition of Bengali Handwritten Characters Using Skeletal Convexity and Dynamic Programming

Soumen Bag
Computer Sc. & Engg. Department
IIT Kharagpur
India
bagsoumen@gmail.com

Partha Bhowmick
Computer Sc. & Engg. Department
IIT Kharagpur
India
bhowmick@gmail.com

Gaurav Harit
Computer Sc. & Engg. Department
IIT Rajasthan
India
gharit.gh@gmail.com

Abstract—The main challenge in recognizing handwritten characters is to handle large-scale shape variations in the handwriting of different individuals. In this paper, we present a novel handwritten character recognition method based on the structural shape of a character irrespective of the viewing direction on the 2D plane. Structural shape of a character is described by different skeletal convexities of character strokes. Such skeletal convexity acts as an invariant feature for character recognition. Longest common subsequence matching is used for recognition. We have tested our method on a benchmark dataset of handwritten Bengali character images. Preliminary results demonstrate the efficacy of our approach.

Keywords—Concavity; Convexity; Handwritten character recognition; Bengali handwriting; Longest common subsequence.

I. INTRODUCTION

Handwritten character recognition is a very challenging area of research, particularly for Indian languages. There has been particular interest over the last few decades in recognition of handwritten characters. To improve the recognition performance, many feature selection and extraction methods are reported for Indian languages [1]. A brief overview about few important feature sets used in optical character recognition for Bengali (also called ‘Bangla’) documents is given below.

Dutta and Chaudhuri [2] have proposed topological features, such as junction points, holes, stroke segments, curvature maxima, curvature minima, and inflexion points of character images for performing printed and handwritten Bengali alphanumeric character recognition. Chaudhuri and Pal [3] have proposed the first complete Bengali OCR for printed documents in 1998. In this method, nine different strokes are used as the primary feature set for recognizing basic characters and template-based features are used for recognizing compound characters. Bhowmick *et al.* [4] have proposed ten stroke-based features to capture the shape, size and position information of a digital curve with respect to the character image. Bhattacharya *et al.* [5] have proposed a feature set, obtained by computing local chain-code histograms of the input character shape. Majumdar [6] has introduced a feature extraction method based on the curvelet transform of morphologically altered versions

TABLE I
DIFFERENT FEATURE SETS USED IN BENGALI OCR SYSTEMS

Method	Feature set
Dutta and Chaudhuri [2]	Structural and topological
Chaudhuri and Pal [3]	Structural and template
Bhowmik <i>et al.</i> [4]	Stroke based
Bhattacharya <i>et al.</i> [5]	Local chain code histogram
Majumdar [6]	Curvlet coefficient features
Pal and Chaudhuri [7]	Watershed, topological, and statistical
Das <i>et al.</i> [8]	Shadow, longest run and quad-tree based

of an original character image. Pal and Chaudhuri [7] have used water-flow model for detecting convexity of numerals. Topological and statistical features are also used for preparing the feature set. To recognize handwritten basic and compound characters, Das *et al.* [8] have used shadow, longest run, and quad-tree based features. Table I gives a summary of different feature sets used in Bengali OCR systems.

We see from the past several decades that different types of features are proposed for printed and handwritten OCR systems for Indian languages. But the performance of handwritten OCR still needs a lot of improvement. The main challenge in designing a handwritten OCR system is to handle large-scale shape variations in the same character written by different persons. From this point of view, we propose a novel method based on skeletal convexity and dynamic programming (DP) for Bengali handwritten character recognition. The approach of DP has been found to be useful in some contemporary on-line handwriting recognition systems [9], [10], [11], which has motivated us to apply it in our off-line system.

II. SHAPE ANALYSIS OF CHARACTER SKELETONS

Given a scanned document page, we first binarize it using Otsu’s algorithm [12]. Currently we are working with isolated character images. Before extracting the structural shape, character images are converted to thinned (i.e., single pixel thick) curve segments [13]. But to retain the proper shape of thinned character images is a big challenge. Here we consider a medial-axis based thinning strategy [14] for performing character skeletonization as a preprocessing (see Fig. 1).

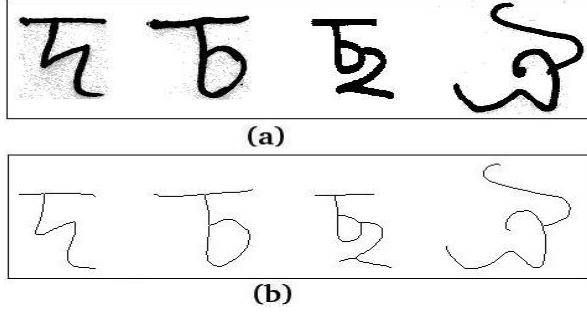


Fig. 1. (a) Input image; (b) Skeleton image.

Next we analyze structural shapes of isolated character images using their concavity and convexity features irrespective of viewing directions, as explained below.

A. Straight Line Approximation

For noisy images, the proposed medial-axis based thinning results in undesired small concave and convex regions. To solve this problem, we apply a straight line approximation method [15] on thinned images (Fig. 2).

The approximation results often contain deviation of thinned images at the junction points. So, to preserve the true shape at the junction points during approximation, we perform *junction point refinement* as explained next.

Let p be a *preliminary junction point* detected in the thinned image. When $\mathcal{N}_8(p) > 2$, the point p is a junction point. Here $\mathcal{N}_8(p)$ denotes the 8-connected (object) neighbors of p . Before applying straight line approximation, we perform *junction point refinement* by placing the *preliminary junction point* p to a new place in such a manner that after straight line approximation there will be no distortion at the junction point. The steps are as follows:

- 1) Find $\mathcal{N}_8(p) = \{n_1, n_2, n_3\}$.
- 2) Compute the median μ of n_1, n_2, n_3 .
- 3) If μ corresponds to a background point, then mark μ as the final junction point, \hat{p} . But if it coincides with any of the neighbor points (n_1, n_2, n_3), then the collision is solved by using the following steps:
 - Perform set intersection operation between two sets $\mathcal{N}_4(p)$ and $\mathcal{N}_4(\mu)$, where $\mathcal{N}_4(\cdot)$ denotes the 4-connected neighbors. Let $\mathcal{N}_4(p) \cap \mathcal{N}_4(\mu) = \{p_1, p_2\}$. Observe that the number of points in $\mathcal{N}_4(p) \cap \mathcal{N}_4(\mu)$ is always 2.
 - Find $\mathcal{N}_8(p_1)$ and $\mathcal{N}_8(p_2)$. If $\mathcal{N}_8(p_1) < \mathcal{N}_8(p_2)$, then $\hat{p} = p_1$; otherwise, $\hat{p} = p_2$.

The improved results are shown in Fig. 2(d).

B. Concavity and Convexity Detection

After applying straight line approximation method on the thinned image, we get a set of approximation points $V = \{p_1, p_2, \dots, p_n\}$ and a set of edges $E = \{e_1, e_2, \dots, e_m\}$ connecting approximation points according to the structural

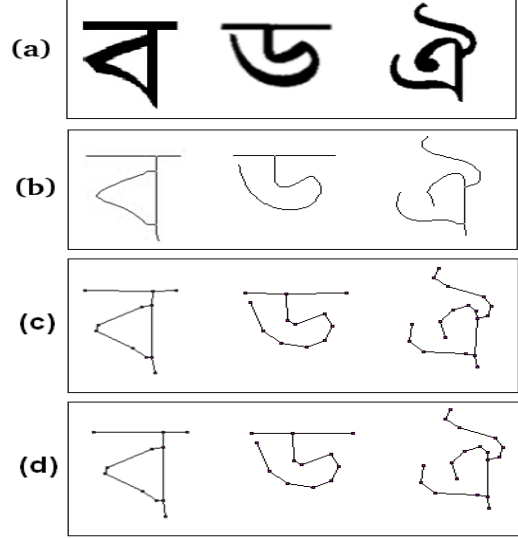


Fig. 2. Straight line approximation of skeleton images: (a) Input image; (b) Skeleton image; (c) Straight line approximation; (d) Improved result after junction point refinement. Notice that the horizontal pieces (mātrās) and the vertical pieces have been straightened after junction point refinement.

shape of character images (Fig. 3(a)-(b)). Now, the convexity and concavity are detected using the following steps:

- 1) Construct an adjacency list \mathcal{L} to represent the graph $G = (V, E)$ where V is the set of approximation points and E is the set of edges.
- 2) Visit all points of the graph starting from an end point p_1 in such a manner that end points (except the start point of the traversal) are visited for exactly one time, junction points are visited depends on the number of branches, and remaining points are visited for exactly two times. Finally, the traversal ends at the start point p_1 .
- 3) After performing the traversal we get a sequence of visited points $\mathcal{T} = \langle p_1, p_2, \dots, p_{i-1}, p_i, p_{i+1}, \dots, p_N, p_1 \rangle$ ($N > n$) (Fig. 3(c)). Now, we detect the concavity and convexity of all these points (except the start and end points of traversal) using the following steps:
 - To detect the concavity/convexity of a point p_i , we need to consider its two adjacent points, p_{i-1} and p_{i+1} . Consider $p_{i-1}(x_{i-1}, y_{i-1})$, $p_i(x_i, y_i)$, and $p_{i+1}(x_{i+1}, y_{i+1})$ as the three vertices of a triangle. Then twice the signed area of this triangle is

$$\Delta(p_{i-1}, p_i, p_{i+1}) = \begin{vmatrix} 1 & 1 & 1 \\ x_{i-1} & x_i & x_{i+1} \\ y_{i-1} & y_i & y_{i+1} \end{vmatrix}.$$

If $\Delta(\cdot)$ yields a negative value, then the point p_i has a concave property and is marked as L . If the value is positive, then p_i has a convex property and is marked as R (Fig. 4). If the value is equal to 0, then the point p_i has the same property of its

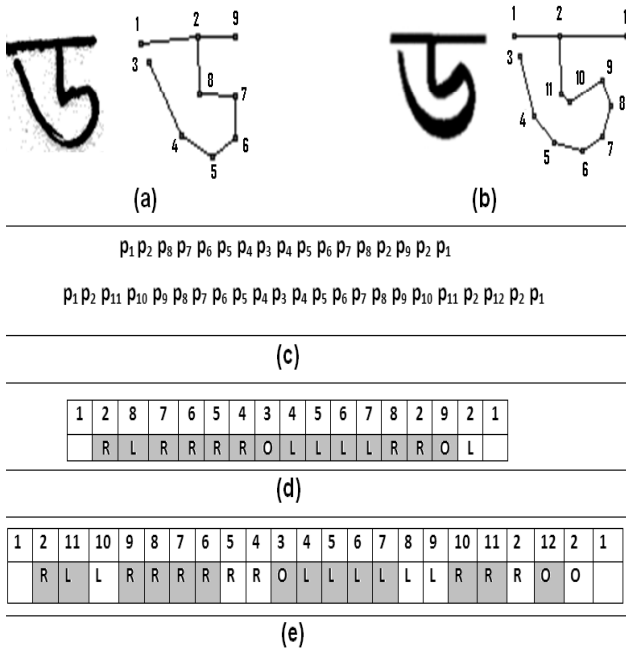


Fig. 3. (a)-(b) Handwritten and printed images and their skeletons with approximation points; (c) Sequence of traversed points (1st and 2nd rows represent the sequences for image (a) and (b) respectively); (d)-(e) *LR* sequence of image (a) and (b) respectively ('L' : Concave, 'R' : Convex, 'O' : End point).

previous point p_{i-1} . We mark end points as *O* to exclude them from concavity/convexity detection.

- After detecting the concavity/convexity of all the points in the set \mathcal{T} , we get a sequence $\{R_2, R_3, L_4, L_5, R_6, R_7, \dots, R_i, \dots, R_N\}$ where L_i/R_i indicates the concavity/convexity of point p_i (Fig. 3(d,e)).

III. CHARACTER RECOGNITION USING DYNAMIC PROGRAMMING

For character recognition, we use *longest common subsequence* (LCS) method [16] on the *LR* sequence detected in the previous section. This dynamic programming method gives the matched longest sub-sequence between two strings. For example, if $A = \langle L, L, R, R, L, L, R \rangle$ and $B = \langle L, R, L, L, L, L \rangle$, then the longest common subsequence of A and B is $\langle L, R, L, L \rangle$.

The following steps outline our approach.

- 1) Prepare a set of prototypes which contains the *LR* sequences of all printed Bengali character images. This prototype set acts as a ground truth for handwritten character recognition.
- 2) Take each handwritten character image as the test image and compute its *LR* sequence, as discussed in the previous section.
- 3) Compute the longest common subsequence between the *LR* sequence of test image and that of each of the

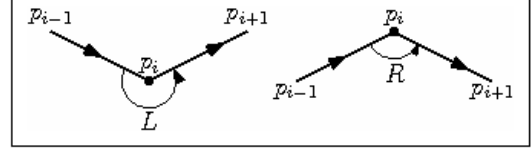


Fig. 4. Detection of concavity and convexity of a point with respect to its neighbor points. *left*: Concave shape; *right*: Convex shape.

prototype images. For each matching, we get a matching score.

One major drawback of the LCS method is that it only computes the matching score but does not consider the mismatch. But this mismatch decreases the recognition performance. To overcome this problem, we modify the matching score as:

$$MS = \frac{|match(s, t)|}{\max(|s|, |t|)}$$

where, $|s|$ and $|t|$ denotes the string length of *LR* sequence of test image and prototype image respectively.

IV. EXPERIMENTAL RESULTS

For experimental purpose, we have taken the handwritten Bengali character database of ISI, Kolkata [17]. The database contains 50 Bengali basic characters with wide variation. We have taken 10 different shapes for each character from the database. At first, we have prepared the prototypes of all printed Bengali characters. Each prototype contains the *LR* sequence of the corresponding printed character. Now, the handwritten characters are recognized using the longest common subsequence (LCS) matching with the prototypes. In Fig. 3(a)-(b), there are two images, one handwritten and another printed. To compute the matching score between these two characters, we consider their *LR* sequences (Fig. 3(d)-(e)) and apply LCS matching. The resultant longest common subsequence is highlighted by *gray* color. In this example, $|s| = 15$, $|t| = 21$, and $|match(s, t)| = 14$, which yields $MS = 14/21 = 0.67$. We have tested 500 handwritten characters and the success rate is 60.6%. Fig. 5 shows a subset of experimental results. For each test case, the first three best matches (based on their matching scores) are reported. Few failures are also reported in Fig. 6.

V. CONCLUSION

In this paper, we have proposed character recognition method based on skeletal convexity. We have analyzed concavity/convexity of character strokes along two possible directions and have used these features for recognition. The proposed method is tested on handwritten Bengali character images and the preliminary results are quite promising. But this method is not performing well for few particular characters. In future, we shall extend our work to improve the accuracy of recognition and to make it applicable to character classification for handwritten Bengali OCR system.



Fig. 5. 1st col: Input image; 2 – 4th col: Best three matches as per their matching scores (MS).

REFERENCES

- [1] U. Pal and B. B. Chaudhuri, "Indian script character recognition: A survey," *Patt. Rec.*, vol. 37, pp. 1887–1899, 2004.
- [2] A. Dutta and S. Chaudhuri, "Bengali alpha-numeric character recognition using curvature features," *Patt. Rec.*, vol. 26, no. 12, pp. 1757–1770, 1993.
- [3] B. B. Chaudhuri and U. Pal, "A complete printed Bangla OCR system," *Patt. Rec.*, vol. 31, no. 5, pp. 531–549, 1998.
- [4] T. K. Bhowmik, U. Bhattacharya, and S. K. Parui, "Recognition of



Fig. 6. 1st col: Input image; 2 – 4th col: First three mismatches as per their matching scores (MS).

Bangla handwritten characters using an MLP classifier based on stroke features," in *Proc. ICONIP*, 2004, pp. 814–819.

- [5] U. Bhattacharya, M. Shridhar, and S. K. Parui, "On recognition of handwritten Bangla characters," in *Proc. ICVGIP*, 2006, pp. 817–828.
- [6] A. Majumdar, "Bangla basic character recognition using digital curvelet transform," *J. Patt. Rec. Research*, vol. 2, no. 1, pp. 17–26, 2007.
- [7] U. Pal and B. B. Chaudhuri, "Automatic recognition of unconstrained off-line Bangla handwritten numerals," in *Proc. Intl. Conf. Advances in Multimodal Interfaces*, 2000, pp. 371–378.
- [8] N. Das et al., "Handwritten Bangla basic and compound character recognition using MLP and SVM classifier," *Journal of Computing*, vol. 2, no. 2, pp. 109–115, 2010.
- [9] F. Andrianasy and M. Milgram, "Dynamic character recognition using an elastic matching," in *Proc. CAIP*, 1995, pp. 888–893.
- [10] X. Li and D. Y. Yeung, "On-line handwritten alphanumeric character recognition using feature sequences," in *Proc. Intl. Comp. Sc. Conf. Image Anls. Appl. and Comp. Graphics*, 1995, pp. 197–204.
- [11] L. Saysourinhong, B. Zhu, and M. Nagakawa, "On-line handwritten Lao character recognition by using dynamic programming matching," in *Proc. ISCIT*, 2008, pp. 473–476.
- [12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall (USA), 2008.
- [13] L. Lam, S. W. Lee, and C. Y. Suen, "Thinning methodologies—A comprehensive survey," *IEEE Trans. PAMI*, vol. 14, no. 9, pp. 869–885, 1992.
- [14] S. Bag and G. Harit, "A medial axis based thinning strategy for character images," in *Proc. NCVPRIPG*, 2010, pp. 67–72.
- [15] P. Bhowmick and B. B. Bhattacharya, "Fast polygonal approximation of digital curves using relaxed straightness properties," *IEEE Trans. PAMI*, vol. 29, no. 9, pp. 1590–1602, 2007.
- [16] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithm*. Prentice Hall (India), 1998.
- [17] www.isical.ac.in/~ujjwal/download/database.html

Acknowledgement

A part of this work is sponsored by the project "Image analysis for preservation and archiving of Indian Cultural Heritage (APA)" No. NRDMs/11/1586/2009 Dt.03.06.2010.