

A System for Bangla Handwritten Numeral Recognition

Umapada Pal, Abdel Belaïd, B.B. Chaudhuri

► **To cite this version:**

Umapada Pal, Abdel Belaïd, B.B. Chaudhuri. A System for Bangla Handwritten Numeral Recognition. International Conference on Knowledge Based Computer Systems - KBCS, Mar 2003, Navi Mumbai, India, 11 p. inria-00107679

HAL Id: inria-00107679

<https://hal.inria.fr/inria-00107679>

Submitted on 19 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A System for Bangla Handwritten Numeral Recognition

U. Pal

CVPR Unit
Indian Statistical Institute
203 B. T. Road
Kolkata -108, India
umapada@isical.ac.in

A. Belaid

Group READ, LORIA
Campus Scientifique
B. P.239 , 54506
Vandoeuvre,Nancy,France
abelaid@loria.fr

B. B. Chaudhuri

CVPR Unit
Indian Statistical Institute
203 B. T. Road
Kolkata -108, India
bbc@isical.ac.in

Abstract

This paper deals with a recognition system for unconstrained off-line Bangla handwritten numerals. To take care of variability involved in the writing style of different individuals, a robust scheme is presented here. The scheme is mainly based on new features obtained from the concept of water overflow from the reservoir as well as topological and structural features of the numerals. The proposed scheme is tested on data collected from different individuals of various background and we obtained an overall recognition accuracy of about 92.8% from 12000 data.

1 Introduction

Recognition of handwritten numerals has been a popular research area for many years because of its various application potentials. Some of its application area are automatic postal sorting, automatic bank cheque processing, share certificate sorting etc. Although research on recognition of unconstrained handwritten numerals has made impressive progress in Roman, Chinese and Arabic script [Plamondon and Srihari, 2000], recognition of handwritten Indian numeral is largely neglected. Only a few research papers have been published on Indian handwritten numerals [Sethi and Chatterjee 1977; Dutta and Chaudhuri, 1993; Pal and Chaudhuri 2000; Bhattacharya et al., 2001] although there are 12 different scripts in India.

Various approaches have been proposed by the researchers for numeral recognition [Plamondon and Srihari, 2000] but most of them are used for non-Indian numeral recognition. One of the most widely used approaches is based on neural networks [Kim et al 2000]. Here the network architecture is trained by a set of training data and then the input is classified by the trained networks. Some researchers used structural approach, where each pattern class is defined by structural description and the recognition is performed according to structural similarities [Cai and Liu, 1999]. Statistical approach is also applied to numeral recognition. It is relatively insensitive to pattern noise and

distortion but modelling of statistical information is a tedious task [Plamondon and Srihari, 2000]. Among others, Hidden Markov Models, Fourier and Wavelet Descriptors [Wunsch and Laine, 1995], Fuzzy rules [Chi and Yan, 1995], tolerant rough set [Kim and Bang, 2000], are reported in the literature.

In this paper, we propose a normalization and thinning free automatic scheme for off-line Bangla handwritten numeral recognition. Bangla is the second-most popular language in the Indian sub-continent and fifth-most popular language in the world. To get an idea of Bangla numerals and their variability one set of printed and seven sets of handwritten numerals are shown in Fig.1. The proposed scheme has two parts: (a) segmentation (b) recognition. The segmentation part of the scheme first detect whether a component of numeral(s) is isolated or touching. If it is touching, segmentation scheme is applied on it to get individual numerals. Recognition technique is then applied on these individual numerals. Previously we proposed a scheme for touching English numeral segmentation [Pal et al., 2001]. Here we propose a modified segmentation scheme for touching Bangla numerals.

When two numeral touch each other, they create large space (reservoir) between the numerals (see Figure 2) which is very important for segmentation because touching points generally lie around this space. Using a simple concept based on water reservoir this space is encountered for segmentation. At

first, the positions and sizes of the reservoirs are analyzed and a reservoir is detected where touching is made. Considering the type (top or bottom reservoir) and analyzing *base* of this reservoir the touching position (top, middle or bottom touching) is decided. Next, noting touching position and analyzing the profile of the reservoir, the initial feature points for segmentation are determined. Considering close loops, reservoir heights and distance from center of the component the initial feature points are ranked and the best feature point (highest rank point) is noted. Finally, based on touching position, close loop positions and morphological structure of touching region the cutting path is generated. Since different individuals have different writing styles, the topological and other properties of numerals may vary. To take care of these variability water reservoir based features as well as topological and structural features of the numerals are used in the recognition scheme.

2 Water Reservoir Analogy Principle

The water reservoir principle is as follows. If water is poured from top (or bottom) of the numeral, the cavity regions of the numerals where water will be stored are considered as reservoirs. For an illustration see Fig.2. Here by top (bottom) reservoirs we mean the reservoirs obtained when water is poured from top (bottom). (Water from bottom may be visualized by the water being poured from top after rotating the component by 180°). One of these reservoirs will be obtained in the large space created by touching.

All reservoirs from top and bottom are not considered for future processing. The reservoirs having heights greater than a threshold T_1 are only considered. The value of T_1 is chosen as 1/6 times the corresponding numeral height, thus making it numeral size invariant (The constant 1/6 is obtained

from experiment). Reservoir concept is used both for touching numeral segmentation and isolated numeral recognition.

১	২	৩	৪	৫	৬	৭	৮	৯
০	১	২	৩	৪	৫	৬	৭	৮
৯	০	১	২	৩	৪	৫	৬	৭
৮	৯	০	১	২	৩	৪	৫	৬
৭	৮	৯	০	১	২	৩	৪	৫
৬	৭	৮	৯	০	১	২	৩	৪
৫	৬	৭	৮	৯	০	১	২	৩
৪	৫	৬	৭	৮	৯	০	১	২
৩	৪	৫	৬	৭	৮	৯	০	১
২	৩	৪	৫	৬	৭	৮	৯	০
১	২	৩	৪	৫	৬	৭	৮	৯
০	১	২	৩	৪	৫	৬	৭	৮

Fig.1: Example of Bangla numerals.

3 Feature Selection and Detection

The features are chosen with the following considerations: (a) Independence of various writing styles of different individuals, and (b) simplicity of detection.

The direction of water overflow, height of water reservoir when water overflows from the reservoir, position of the reservoir with respect to the character bounding box, shape of the reservoir etc. are the main reservoir features used in the proposed scheme. The computation of water reservoir is simple. We find the regions of white space in the bounding box of the component, where water could be stored. These regions represent the water reservoirs.

The close loop features are the main topological features in our scheme. The number of close loop, their positions with respect to the bounding box of the component, center of gravity and the ratio of close loop height to component height are considered here.

In the structural feature we consider the morphological pattern of the touching region. This feature is very helpful for the cutting path detection of connected numerals segmentation. The number of crossings in a particular region of the numeral, contour tracing feature, and convexity of hole are also used in the proposed scheme.

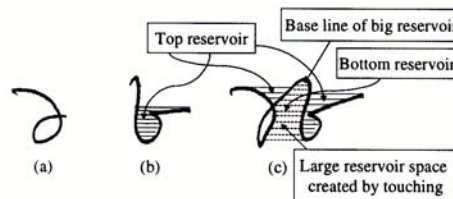


Fig.2:Example of water reservoir and large space created by touching is shown. Here touching numeral shown in Fig.2(c) is generated by two isolated numerals shown in Fig.2(a) and 2(b).

4 Numeral separation and segmentation

4.1 Numeral separation

This stage of the scheme classifies an input numeral string into isolated or touching digit group. Generally in the earlier studies, aspect ratio (height width ratio) of the component is used for the separation [Chen and Wang, 2000; Kim et al., 2000]. If two or more numerals are connected, then the width of the connected component should be larger than its height. Although by and large, it may be true for printed text, it is not so for handwritten cases because of different writing styles of the individuals. Complexity (in terms of curvature) is also used for isolated/touching numeral separation. This is also not true in general. For example, if two zeros touch each other to make a touching pattern, the complexity of this pattern is more or less equal to the numeral “8”.

In principle, when two numerals get connected, one of the following happens in most cases: (i) two numerals create a large space (reservoir) between them (as shown in Fig.2(c)). (ii) the number of reservoirs (obtained from both top and bottom) in a connected numeral is greater than that of a isolated numeral. For example, see Fig.2. In the isolated numeral shown in Fig.2(a) there is no reservoir while the numeral shown in Fig.2(b) has only one reservoir. The connected numeral shown in Fig.2(c) has three reservoirs. Based on the number of reservoirs, their size and positions, the number of close loop and their location, isolated and connected numerals are identified as follows.

Let for a component C:

N = Number of close loops

$G_i(x,y)$ = CG (Centre of Gravity) position of i^{th} close loop, $i = 1, \dots, N$

W = Total number of water reservoirs (both top and bottom)

$2_j = j^{\text{th}}$ reservoir, $j = 1, 2, \dots, W$

$\hat{G}_i(x,y)$ = CG position 2_j

2_{Hj} = Height of 2_j

$T = N+W$

Also let:

$S_A = \text{func}_1(N, G_i(x,y)) = 1$ if $N \geq P_1$ and $-45^\circ \leq \theta \leq 45^\circ$

(θ is the angle between the CGs of any two close loops)

= 0 otherwise

$S_B = \text{func}_2(2_{Hi}, \hat{G}_i(x,y)) = 1$ if there exists an i such that $2_{Hi} \geq 75\%$ of the component height and

$\hat{G}_i(x,y) \in h_m$ ($i=1,2,..W$) (h_m is defined later).

= 0 otherwise

$S_C = \text{func}_3(W) = 1$ if $W \geq P_2$

= 0 otherwise

Where func_1 , func_2 and func_3 are Boolean functions described as above. The values of the parameters P_1 and P_2 are chosen as 2 and 3 respectively, from the experiment. Based on these above values the separation scheme is as follows.

If ($S_A = 1$) then C is connected
 else if ($S_c = 1$) then C is connected
 else if ($S_B=1$ and $T \geq 3$) then C is connected
 else if ($S_B=1$ and $T < 3$) then C is rejected
 else C is isolated.

The advantage of this method is that it is size independent and there is no need of any normalization of the component. To evaluate the performance of the separation method, a data set of 8000 components was collected of which 3000 data were connected (touching). It is observed that the proposed method has 98.85% accuracy for separating isolated digits and connected strings. Rejection rate of the system is very small (only 1.6%). The rejected components are no longer automatically processed.

4.2 Segmentation of touching numerals

For the segmentation of touching pattern, at first the touching position (top, middle and bottom) is found. Next, the feature points for segmentation are extracted using touching position, reservoir position and topological features of the component. Finally, considering the loops, structural features and reservoir features the segmentation path of the touching pattern is constructed. The bounding box (BB) area of a touching component is partitioned horizontally in three regions. The top region (h_t) is 25% of BB. Middle region (h_m) and bottom region (h_b) are 50% and 25% of BB, respectively. Similarly, by vertical division of BB the left (v_l), middle (v_m) and right (v_r) regions (25%, 50% and 25% of BB, respectively) are formed. At first, the largest reservoir of the component whose center of gravity lies in v_m region is found. This reservoir if present, is called the *best reservoir* for touching. The *base-line* (lowermost row of the reservoir) of the best reservoir is then detected. Base-line of the best reservoir is shown in Fig.2. Touching position detection is done as follows.

Let I_p = base line row of the best reservoir, then

Touching position = top, if $I_p \in h_t$
 = middle, if $I_p \in h_m$
 = bottom, if $I_p \in h_b$.

The feature point extraction for connected numeral cutting is done based on the touching position. If the touching position is top then all reservoirs whose base-line lies in the h_t region are considered for feature extraction. Similarly, if the touching position is bottom (middle) then all reservoirs whose base-line lies in the h_b (h_m) region are considered. The leftmost and rightmost points of the base-line of considered reservoirs are the initial feature points. Now, from initial feature points the best feature point for segmentation is chosen by the confidence values of these points. The following features are considered to compute confidence value (CV) of an initial point: (i) Euclidean distance of initial feature points from the center of gravity of the touching component. Let there be F initial feature points and the Euclidean distance of these points from center of gravity (CG) of the component are d_1 ,

$d_2 \dots d_F$. Then the confidence value of a feature point with distance d_i is $1/K$, where $K = d_i / (d_1 + d_2 + \dots + d_F)$. (ii) Distance of initial feature points from the CG of each close loop. Confidence value computation for this feature is similar to above feature. (iii) Height of the reservoir. The main idea of this feature is that the points coming from bigger reservoir should get more confidence value. Let F initial points be obtained from p reservoirs of heights $2_{H1}, 2_{H2}, \dots, 2_{HP}$. Then the confidence value of a point obtained from reservoir 2_i is $2_{Hi}/S$, where $S = 2_{H1} + 2_{H2} + \dots + 2_{HP}$.

For each initial point, its total confidence value is checked. The point with highest confidence value is the best feature point. Now noting closed loop position and the structural shapes of the touching portion, the best cut point is determined from the best feature point. To get the best cut point we trace the boundary of the reservoir (the reservoir from which the best feature point is obtained) in clockwise direction to find a node (obstacle) point. During tracing we note the vertical run length of black pixels at each boundary point. The point, where difference of two consecutive run lengths is $3R/2$, is the node point. Here R is the statistical mode of the vertical black run lengths of the component. The boundary of the reservoir is also traced anti-clockwise direction and another node point is found in similar way. Noting the position of these node points the best cut point (best node point) is decided.

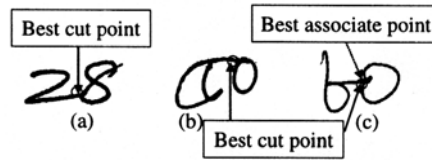


Fig.3: Best cut point and associate point are shown. (a) bottom (b) top and (c) middle touching component.

For the top and bottom touching components vertical segmentation is done at the best cut point. For the middle touching components the cutting method is different. From the best cut point we try to associate the other initial features which are obtained from the opposite type reservoir to best reservoir and choose a point (best associate point) from these initial points. Cutting path is obtained by joining best cut point and its best associate point (best associate point is calculated in similar way in which best feature point was calculated). The best cut points and best associated point are shown in Fig.3.

The segmentation result was verified manually and it was observed that 92.4% of the connected numerals were correctly segmented. Some segmented results are shown in Fig.4. The result shown in Fig.4(c) is mis-segmented because of the big overlap region. The proposed scheme does not depend on the size and style of the handwritten numerals. The rejection rate of our segmentation system was 2.9%.

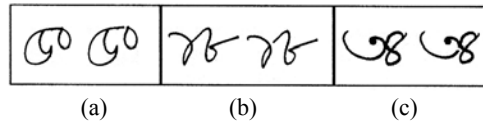


Fig.4: Some segmented results of the proposed scheme.

The main features used for rejection were: (a) the widths of one of the segmented part is very small compared to the width of other part; (b) the length of cutting path is very long compare to the height of touching pattern; (c) no best reservoir is obtained in a touching component.

At present, the segmentation scheme handles two digit single touching components. In the next studies we plan to develop a more general system to handle touching patterns of three or more numerals.

5 Numeral Recognition

The proposed recognition scheme is an extension of our earlier work [Pal and Chaudhuri, 2000]. Here, a binary tree classifier is employed for the numeral recognition. At first, using reservoir features like number of reservoirs, their heights and positions, topological feature like number of hole, their centre of gravity (CG) positions etc, we generate a binary tree where a leaf node of the tree may contain up to 3 numerals. Next, we use more specific feature to identify numerals of different leaf nodes.

In Bangla printed numeral set, there are five numerals which have hole. These numerals are ৪ (four), ৫ (five), ৭ (seven), ৮ (eight) and ০ (zero). (Printed numeral are shown in first column of Fig.1) Although there are five numeral with hole, some handwritings may not show up holes. Conversely, in some handwritings we may get holes in some numerals where no hole is there in their printed form. It may be noted that some persons write the numeral ১ (one) with a hole while others write it without hole. To make the system robust we use both of these characteristics in the system.

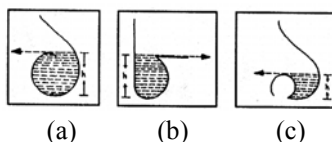


Fig.5: Three numerals of a leaf node and their reservoir height (h) and water overflow direction (by arrow) from the top reservoir are shown. These reservoirs are obtained in the hole-like regions.

From a handwritten Bangla data set of 12000 numerals we noted that most people try to make a loop in the numerals if the actual numeral has a loop. For example, in numeral ৮ (eight) there is a loop in the lower half position. From our experiment on the above data set we note that 65% individuals draw a loop in this numeral. In other handwritings, a hole (cavity region) occurs instead of loop. We use water reservoir principle to find loop like structure in the hole position of these handwritings.

To get an idea about the use of this hole-like information in the recognition process consider the numerals shown in Fig.5. Because of their common feature (all have hole-like structure) these numerals belong in the same leaf node of the classification tree. Note that the position of reservoir for all these numerals is in the lower part of the bounding box and as a result, they have been classified in the same group. For their classification the water overflow direction from reservoir is noted. For the numerals shown in Fig.5(a) and (c), the overflow direction is left whereas it is right for the numeral of Fig.5(b). For the classification of the numerals of Fig.5(a) and (c) we use number of crossing. We

consider left side of the hole (reservoir) and compute row-wise number of crossing in this part. For the numeral of Fig.5(a), the maximum number of crossings in row-wise scans will be only one, whereas for the numeral of Fig.5(c) the number of crossings will be more than one.

To recognize some numerals, the ratio of reservoir/hole height to the numeral height is used. For example, numerals 5 (five) and 0 (zero) are classified in the same leaf node because the ratio of reservoir/hole height to numeral height is more than 0.80 for these numerals. For the classification of these numerals we use convex set property of the points inside the reservoir/hole. The points inside the reservoir/hole of the numeral 0 (zero) is nearly convex whereas points inside the reservoir/hole of the numeral 5 (five) are highly non-convex. For example, see Fig.6. To test the near convexity we have chosen two points (x_1, y_1) and (x_2, y_2) on these numerals. We divide the reservoir of the numeral into three horizontal strips and find the rightmost point among the points of the first strip of the reservoir. This point is (x_1, y_1) . The rightmost point from the points of the third strip of the reservoir is (x_2, y_2) . These points are shown in Fig.6 on the two numerals 0 (zero) and 5 (five). The line segment obtained by joining these two points always lie on the reservoir for the numeral 0 (zero), whereas it goes outside the reservoir for the numeral 5 (five).

To identify similar shaped numerals of a leaf node a contour tracing algorithm is implemented. During contour tracing we calculate the distance of each traced point (pixel) to the top or right side of the character's bounding box. Top or right side is decided according the group of the leaf node. Noting the distance sequence we have identified the characters.

For contour tracing two points (S_1 and S_2) are at first identified. Starting from the point S_1 the outer contour is traced upto the point S_2 in clock-wise or anti-clockwise direction. The points S_1 and S_2 and the tracing direction are decided according the group of the leaf node. For an example see Fig.7. Here two confusing numerals 1 (one) and 2 (two) are shown. For this confusing pair we choose S_1 (S_2) as the topmost (bottommost) point of the external contour of the numeral and the direction is clockwise. For the numeral 1 (one) we get 1 transition point while for 2 (two) we get 2 transition points. This transition number is used as the feature for their identification. By transition we mean change of distance value from increasing mode to decreasing mode or decreasing mode to increasing mode. The transition points are marked by t_1 and t_2 in Fig.7.

Note that in our classification scheme we do not use any stroke-like features because in hand-written text stroke features may change widely from person to person.

6 Results and Discussion

We applied our scheme on 12000 numerals obtained from different individuals of different professions like school, college and university student and teacher, bank and post office employee, business men etc. We noted that the data sets contain varieties of writing styles. We noted that the accuracy rates of the recognition scheme is 92.8%.

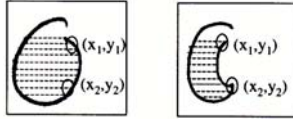


Fig.6: Numeral recognition using convex set property: Here reservoir is marked by dashed lines. (x_1, y_1) and (x_2, y_2) are the rightmost points of the first and third strip of the reservoir.

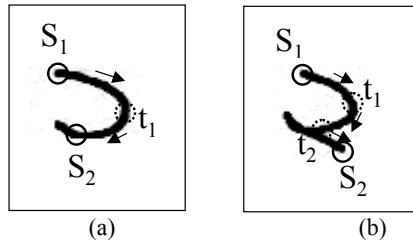


Fig.7. Identification of two confusing numeral by boundary tracing algorithm.

There are not enough work on Bangla numeral recognition. These exists two pieces of work which are based neural network. Recognition scheme proposed by Dutta and Chaudhuri (1993) has an accuracy of 90% tested on a data set of 100 numerals. The scheme proposed by Bhattacharya et al. (2001) has an accuracy of 89.63%. They tested their scheme on 3330 samples of Bangla numerals.

From the experiment we noted that numeral ৮ (eight) has the highest recognition rate of 97.8%. This is because of its different water overflow direction as compare to other two numerals (১ (one) or ৯ (nine)) in the same leaf node of the tree. Also, we noted that the highest confusing numeral pair is ৩ (three) and ৬ (six) (the first pair of Fig.8(a)). We noted that in about 8.19% cases they confuse each other. Their similar shapes rank their confusion rate at the top position. Next confusion pair is ১ (one) and ২ (two) and their confusing rate is 7.78%. Other examples of confusion pairs noted during experiment are given in Fig.8(a). Confusion rates are computed from the total number of confusions obtained from the experiment.

The methods based on thinning may face problems due the protrusions obtained after thinning [Chen and Wang, 2000]. Also, contour following approach will not work properly because of various writing styles. For example, if a touching numeral is segmented in a form shown in Fig.8(b), then the method based on thinning and contour tracing may fail. Our recognition method will properly recognize these segmented numerals without any modification in the system.

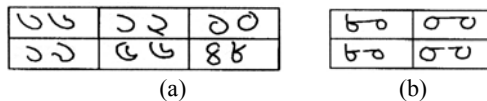


Fig.8:(a) Example of some confusing Bangla handwritten numeral pairs (b) Example of touching Bangla numerals and their segmented form.

The drawback of the proposed method is that it will fail if there is a break on the contour portion used as the boundary of the reservoir. In that case water cannot be filled up properly to get reservoir (hole) and hence miss-recognition will occur. But many other methods (for example those based on contour following) will also fail for this type of situation. We may note that such cases are very rare. We use smearing technique [Chaudhuri and Pal, 1998] to remove some of these situations where size of the break point area is small.

Conclusion:

This paper deals with a new scheme for the recognition of unconstrained off-line Bangla handwritten numerals. The proposed scheme has two parts: segmentation and recognition. The segmentation part first detects whether a component is touching or not. If a component is touching, then segmentation scheme segments the touching component into isolated numerals. To take care of variability involved in the writing style of different individuals, a recognition scheme based on water reservoir concept is proposed here. At present the segmentation scheme handles two digits single touching components only. In future we plan to propose a more general scheme to segment three or more touching components as well as multi-touching components.

References:

- [Bhattacharya et al., 2001] Bhattacharya et al. Self-organizing neural network-based system for recognition of handprinted Bangla numerals. *Proc. Computer Society of India*, pages C92-C96, 2001.
- [Cai and Liu, 1999] J. Cai and Z. Q. Liu. Integration of structural and statistical information for unconstrained handwritten numeral recognition. *IEEE Trans. on PAMI*, 21(3): 263-270, 1999.
- [Chaudhuri and Pal, 1998] B. B. Chaudhuri and U. Pal. A complete printed Bangla OCR system. *Pattern Recognition*, 31(5):531-549, 1998.
- [Chen and Wang, 2000] Y. K. Chen and J. F. Wang. Segmentation of handwritten connected numeral string using background and foreground analysis. *Proc. 15th ICPR*, pages 598-601, 2000.
- [Chi and Yan, 1995] Z. Chi and H. Yan. Handwritten numeral recognition using self-organizing maps and fuzzy rules. *Pattern Recognition*, 28:56-66, 1995.
- [Dutta and Chaudhuri, 1993] A. K. Dutta and S. Chaudhuri. Bengali alpha-numeric character recognition using curvature features. *Pattern Recognition*, 26:1757-1770, 1993.
- [Kim and Bang, 2000] K. Kim and S. Y. Bang. A handwritten numeral character classification using tolerant Rough set. *IEEE Trans. on PAMI*, 22(9): 923-937, 2000.
- [Kim et al., 2000] K. Kim, J. H. Kim and C. Y. Suen. Recognition of Unconstrained Handwritten Numeral Strings by Composite Segmentation method. *Proc. 15th ICPR*, pages 594-597, 2000.
- [Pal et al., 2001] U. Pal, A. Belaid and Ch. Choisy. Water Reservoir Based Approach for Touching Numeral Segmentation. *In Proc. Sixth Int. Conf. on Document Analysis and Recognition*, pages 892-896, 2001.
- [Pal and Chaudhuri, 2000] U. Pal and B. B. Chaudhuri. Automatic Recognition of Unconstrained Off-line Bangla Hand-written Numerals. *Advances in Multimodal Interfaces, Springer Verlag Lecture Notes on Computer Science (LNCS-1948)*, Eds. T. Tan, Y. Shi and W. Gao, pp. 371-378, 2000.
- [Plamondon and Srihari, 2000] R. Plamondon and S. N. Srihari. On-line and off-line handwritten recognition: A comprehensive survey. *IEEE Trans. on PAMI*, 22: 62-84, 2000.

[Sethi and Chatterjee, 1997] I. K. Sethi and B. Chatterjee. Machine recognition of constrained hand-printed Devnagari. *Pattern Recognition*, 9: 69-76, 1977.

Wunsch and Laine, 1995] P. Wunsch and A.F. Laine. Wavelet Descriptors for Multi-resolution Recognition of Hand-printed Digits. *Pattern Recognition*, 28:1237-1249, 1995.