Synthesis of individual handwriting in Bangla script

Bidyut B Chaudhuri
CVPR Unit,
Indian Statistical Institute
203 B.T. Road, Kolkata-700108
E-Mail: bbcisical@gmail.com

Abhisek Kundu
CSE Department,
Jadavpur University
Kolkata-700032
E-Mail: abhisekkundu@hotmail.com

Abstract

Document processing concerns synthesis as well as analysis and recognition of documents. This paper concerns automatic synthesis of handwritten Bangla text in an individual style of writing. Individual handwriting has macro structures like line orientation, neighboring word unevenness, interline spacing, margin orientation etc as well as micro properties like individual character shapes and inter-character transitions. Because of high complexity of Bangla writing, a combination of stored character database and stroke synthesis by spline with paint brushing is employed in our present synthesis model. The macro structure properties are also computed and used in the synthesis approach. A typical example of synthesis result is shown and other results are discussed.

Keywords: handwriting-synthesis, macro structures, micro properties, stroke synthesis, spline.

1. Introduction

There are two major aspects of digital document processing namely, document synthesis and document analysis. Document synthesis is mostly covered by printing and reprography technology, while document analysis and recognition falls under the general domain of pattern recognition and image processing. Outstanding advancement has been achieved in printing methodology and even a complex document can now be generated in the DTP environment. However, the commercial printing softwares work with well-structured fonts of various styles. Although semi-cursive handwriting-like fonts are available, they are truly mechanical and any character printed at various places in a document will look exactly the same everywhere. In other words, such fonts do not capture the individuality of human handwriting.

This paper deals with computer synthesis of individual handwritten text. A handwriting synthesis approach may have several practical usefulness: (a) One may be interested in printing or generating a soft copy of his/her own handwriting in computer and send to near and dear ones, (b) For publishing a book in two forms, e.g. writer's own handwriting, as well as in mechanized font, machine synthesis offers some advantages during editing and page-making. (c) Synthesis approach may help automatic search and retrieval from handwritten document image database, (d) For handwritten OCR research, synthesized database is useful in creating more versatile training sets, (e) Synthesis software development may require some degree of handwriting individuality analysis, and the methodology may be useful in authentication of writing in questioned documents. Also, individuality analysis may help a person to rectify shortcomings of his/her handwriting styles. Moreover, the handwriting teachers of primary schools may also benefit from such software.

Computer synthesis of handwriting has been initially attempted in early nineties. Two types of models have been used. One is based on motor action of fingers, e.g. delta lognormal model [11], its modified version beta velocity model [2] and oscillatory motion model [12]. The other approach is based on mathematical expression of curve drawing without considering the motor action, such as [11, 12] or allograph code based model [6]. A matching based approach is described in Zheng and Doermann [13]. Some studies are reported on synthesis of newer character shapes by distorting the written characters. The motivation is to increase the data for robust training of recognition engine [3]. Properties of a handwritten text are also computed to know the individuality of forensic application, but not to simulate the handwriting [8]. Although most synthesis studies are on English, those for Korean [2] and Kanji [7] are also found in the literature. However, work on Indian script is rare, an example being [1], which is for online synthesis of individual words only. On the other hand, we are concerned here with offline synthesis.

The present work considers Bangla handwriting synthesis, the script used to write

Bangla, Assamese and Manipuri languages of Eastern India, having a following of more than 300 million people. It is a relatively complicated script, containing basic vowel and consonant characters, vowel modifiers, some of which are glued at the bottom or side of the basic characters, compound of two, three or four consonants as well as their vowel-modifier-joined versions, thus making more than one thousand shapes. It is basically a non-cursive script, but people often try to write it cursively and make various deformations in shape. See Fig 1 for example. The deformation and cursiveness make synthesis more complicated task, since transitions from previous as well as to the next character are involved.

In the rest of the paper, Section 2 describes Bangla handwriting properties. The synthesis model is proposed in Section 3, while the training approach is described in Section 4. Finally, results are discussed in Section 5.

2. Continuous handwriting properties

As seen in Fig 1 the 'headline' (a horizontal bar at the top of most characters) is often omitted in continuous handwriting. The shapes of some character parts are greatly changed even by people with 'good handwriting' (The handwriting of Fig 1 is by Tagore, a Nobel prize winning poet). Also, cursiveness is often introduced for ease and speed-up in writing.

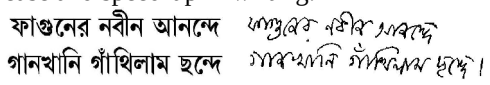


Fig 1. left: Printed text right: Its handwritten version

A page of continuous handwriting has two categories of properties. One is writer's trend of geometrical layout, which may be called global or macro-structure. The other is the individual character shape and the transition stroke to the next character, which may be called local shape, or micro property. The macro structure may include left and right margin, top and bottom space maintained by the writer. Some writers can not maintain uniform margin on the left side, it may increase/decrease as (s)he moves from top to bottom lines. Few people maintain good right margin while others break a word, using hyphen and write the rest in the next line. The writer varies the horizontality of individual text line in [+0.5, +7] degree range where the writing goes slowly upwards from left to right. Some people become conscious about this upward trend after a few lines and try to rectify in the next line, making a big gap at the right side. Downward (i.e. negative angle with horizontal) trend on line is very rare. Some writers cannot maintain good linearity, which shows an undulating word positioning in a text line. Text inter-line gaps vary from author to author. The ligatures of one line may intersect the text of the following line, creating problems in automatic segmentation and analysis. Inter paragraph separation is also writer dependent. Moreover, habit of striking out, indented writing between lines also vary with individuals.

Local variations are common on individual character and transition from one to the next character. So, the synthesis model should be able to reproduce both character shape of an individual and the transition between characters. Some characters are inherently multi-stroke, e.g. র, য় while for others, it is the writer's habit that makes it multi-stroke. Very few people have non-cursive writing habit and they do write in almost isolated characters. Synthesizing their writing is somewhat easier.

The cursive nature demands a big handwriting database for the machine training in automatic synthesis. For example, in English with 52 character shapes 52x51/2 bigrams may be created and if we want 10 handwritten copies of each for a writer (s)he should write 26x51x10 bigrams. Now, for Bangla, which has about 1000 basic and mixed shapes, the valid bigrams will run into many thousands of combined shapes. This is unattractive for any person who wants to make his handwriting synthesis model. It is good to device a compromising approach where a smaller training set is needed.

3. Synthesis Model and Approach

The system id described by block diagram of Fig. 2 where there are two phases, namely training phase and synthesis phase. In the training phase, the handwriting from individual is collected; various features are computed and stored in database. In the synthesis phase the user enters the text through keyboard and it is output in the particular individual's handwriting styles.

As stated above, there are two aspects of synthesis; one is the synthesis of individual words and the other is the simulation of page layout habit by the writer. For both, information should be collected from the handwritten pages of the particular writer. At first, we discuss the page layout macro structure nanlysis.

3.1 Macro Structure Model

At present, our page layout model macro structure has five parameters to be estimated. They are- (i) left margin angle, (ii) Inter-line beginning difference (x-value) (iii) text line slope angle, iv) Inter- line gap (y-value), and (v) neighboring word undulation in text lines.

Fig 3 explains these parameters. Here margin angle (we consider only left margin) is defined by θ, with mean $\overline{\theta}$ and standard deviation σ_θ. Over several text pages written by the author, the mean angle $\overline{\theta}$ and σ_θ can be computed during the training phase. In this model, it is assumed that the writer's margin slope has a Gaussian distribution $N(\overline{\theta}, \sigma_\theta)$, truncated to a practical limit. Alternatively, we can assume a uniform distribution. The left

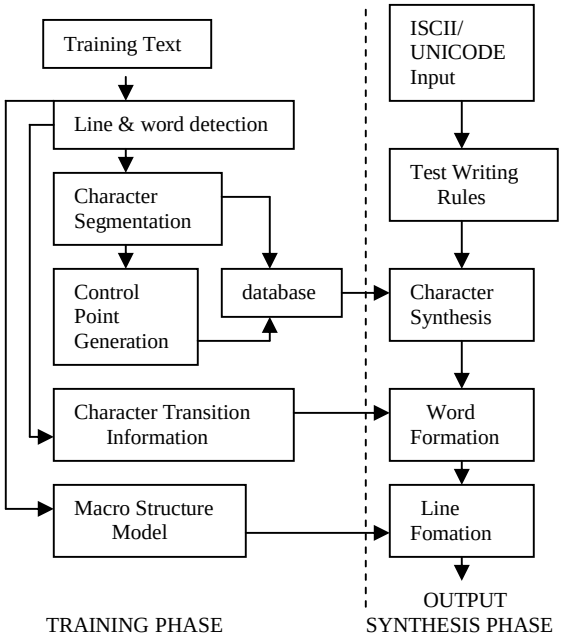


Fig 2. Handwriting Synthesis Approach

margin is computed as follows. A vertical or semi-vertical line l_m is drawn to touch or cross as many text lines as possible, so that the number of black pixels to the left of any individual text line does not exceed a pre-defined number N_{om}. We can make N_{om} dependent on the document page width, so that it is larger for wider page and smaller for shorter page. In a similar manner, $N(\overline{X_L}, \sigma_{XL})$, where X_L is the positional change of first word of current line with respect to the previous one, is also found. Next the line orientation, inter-line gap, and line undulation are modeled. To get orientation of a text line, we find the minimum area rectangular bounding box that contains it. The bounding box may be skewed to get the minimum area. A text line may have deviation from horizontal direction by an angle ϕ, the inter-line gap (Y_L) and over a line, two neighboring words may be vertically mis-aligned by Y_u. Again, we can model them as normally distributed by $N(\overline{\phi}, \sigma_\phi)$, $N(\overline{Y_L}, \sigma_{YL})$, $N(\overline{Y_U}, \sigma_{YU})$, where the parameters of the distribution are estimated from the training samples of handwriting. When used for synthesis, suitable truncation is made on the values.

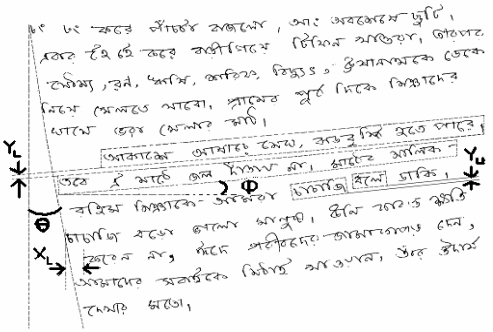


Fig 3. Parameters used in page layout model

3.2 Word Synthesis Model

If a text is entered in UNOCODE or ISCII (Indian Standard Code for Information Interchange), a converter should generate the correct sequence of glyphs, characters, modifiers or allographs to be placed according to writing rules of Bangla script. Unlike English, where one character follows the other, a reasonable number of writing rules exist for Bangla. There are alternatives as well as exceptions to the rules, some of which will be discussed below.

3.2.1 Text writing rules in Bangla

In Bangla two or more consonants may combine to form a compound character that may be very different in shape from the participating characters. A vowel following a consonant can take a new shape called vowel modifier. This modifier may appear to the left, right (or partly left and partly right), top or bottom of a consonant. Similarly, some consonants form consonant modifiers instead of compound shape.

Some of the Bangla writing rules are as follows: (1) Full shaped vowel A (a) can appear only as the first character of a word, (2) all other vowels can also appear in basic form at the first place of a word; otherwise B (A) and D (Long-E) get modified into া and ী respectively, and are

placed to the right of the consonant, (3) C (Short-E) takes the form of with a consonant, but is placed to its left, (4) E (Short-U), F (Long-U), G (Ree) also get modified with a consonant and the modifier is placed at the bottom of the associating consonant, (see rule alternative (i) also) (5) k (Semi-vowel-J) turns into É when compounded with another consonant of a word, (6) J (O) becomes , which is a combination of (E-modifier) and (A-modifier); the associating consonant resides between these two modifiers, (7) if l (R) is compounded with a consonant, it turns into *ref* and appears above the consonant, (8) If there are two consecutive vowel sounds after a consonant, the first takes modified form, while the second takes basic alphabetic form: e.g. Ll + BCh => Ll¡Ch. As stated before, there are exceptions to the rules. Some exceptions/alternatives are- (i) Modifiers like Short-U-modifier () and Long-U-modifier () may appear at the bottom of the previous consonant: / ; but, these modifiers themselves can take new shapes appearing to the right side of associated consonant: l¦/l©, (ii) some Compounds may be expressed in two ways. In one way it is amalgamated into a complex shape. In the other way the consonants are combined by '*hasant*' sign: N + e => NÀ or NÚe, (iii) Compounds can be written in different ways: a + a => or š, (iv) example of exception to the rule (8) is: AaHh etc.

3.2.2 Character synthesis

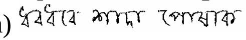
For our character synthesis we used cubic splines combined with a paint-brushing whose thickness is estimated a priori from the writer's handwriting stroke thickness. The brush thickness is estimated by black run length histogram on the text computed along the horizontal direction. The highest frequency of run length gives the brush thickness as shown in fig 4. There may be some blobs in the characters, which are simulated by ellipse of various ratios at blob points. These and spline control points are stored in database. The following cubic spline relation has been used to synthesize the curves.

$$\mathbf{P(t)} = \sum_{i=1}^{4} \mathbf{B_i t^{i-1}}, \quad \mathbf{t_1 \le t \le t_2} \qquad (3\text{-}1)$$

where t_1 and t_2 are the parameter values at the beginning and end of the segment. P(t) is the position vector of any point on the cubic spline segment. P(t)=[x(t) y(t)] is a vector valued function.

$$\mathbf{x(t)} = \sum_{i=1}^{4} \mathbf{B_{i_x} t^{i-1}}, \quad \mathbf{t_1 \le t \le t_2} \qquad (3\text{-}2)$$

The constant coefficients B_i are determined by specifying four boundary conditions for the spline segment. Some typical synthesized characters and modifiers/symbols are shown in fig 5. Note the uniformity of stroke thickness and straightness (e.g. last column) in the synthesized version, compared to the original handwriting.

a) 

b)

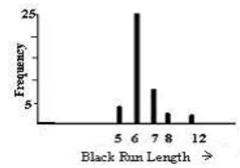


Fig 4. a) Sample Text b) Histogram of Black Run Length

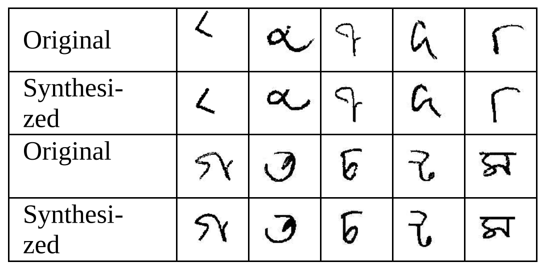


Fig 5. Original versus Synthesized Characters

Since Bangla handwriting is often cursive, continuity from one character to the next needs to be synthesized. This is also done by cubic splines using the *continuity* control points stored in the database for individual characters.

In handwriting, the character size may vary from place to place. Though all parameters and control points are generated and stored on normalized characters, we allow small variation in size at the synthesis stage. Again, while synthesizing the current size of a character is decided randomly by a value that is taken from truncated normal distributed with zero mean and small standard deviation. The truncation allows only 10% increase in size.

4. Synthesizer Training

Since Bangla has a large number of character shapes, a few pages (about ten) of handwriting should be taken from the writer for database generation. The text needs to be

carefully chosen so that all basic characters, modifiers, compound characters are there in compact subset of meaningful sentences.

These pages are scanned at 400 dpi and binarized to get the two-tone electronic version. The synthesis parameters are extracted from these images in semi-automatic manner. For example, the text lines and words are identified by algorithm described in another paper, but the character information are interactively obtained from the words. This is so because accurate and automatic character segmentation is a daunting task and our segmentation module has only 70% accuracy.

Using the line and word information, the page layout model parameters are computed by five sub modules corresponding to the five pairs of the parameters of the normal distributions described in Sec 3.1.

The manually segmented characters are first normalized in size by contracting/expanding in bounding box of uniform height, but variable width. Then selection of the control parameters for spline-based synthesis is initiated. Using the skeletonized version of the character, the end points, crossing points and points of high-curvature are found. These points are called the *fixed points.* See fig 6 for example. The fixed points as well as some other uniformly spaced points on the character are used to initiate the spline for synthesis. Thus we can generate automatically a synthesized version of the character. Now this version is matched with the training character and the error in mismatch is found. To reduce the error control points other than the fixed points are shifted over the skeleton. Again the character is synthesized and compared with the original. This process is continued till the error is less than a predefined threshold or a

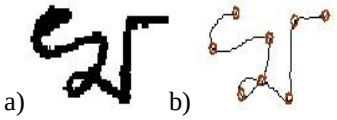


Fig 6. a) Original character b) skeleton with *fixed points*

predefined number of iterations are executed. The set of semi-optimal control points obtained in this way are stored in the character information database.

Blob regions are detected as approximation to circular shape. Blobs are normally thicker than the average stroke thickness by about 10% to 30%. We utilized the uniform thickness (t_v) to make special considerations for blobs. We use ellipses for blobs where the length of minor axis ranges from t_v to $2t_v$, and the length of major axis lies between t_v and $4t_v$ such that major axis length is greater than or equal to the minor axis length. Also orientation of the major axis of such ellipse should be in the direction (with small variation) of the curvature of the edge of the character (see third word of the first line of Fig 8b). At times these ellipses may turn into circles when the length of major and minor axes becomes equal (see the second and fourth word in the second line of Fig 8.b).

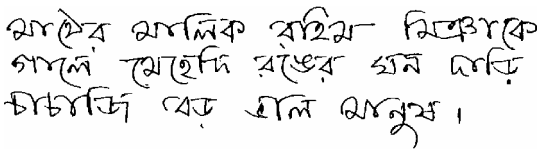
There are a good number of compound characters that are rarely found in text. For the sake of completeness of synthesizer, these as well as some difficult-shaped compound characters are separately written by the candidate, normalized and stored as such. Some of them are shown in Fig 7. While synthesis they are not generated by spline and used directly. All other information are put in appropriate database.

Fig 7. Some complex compound characters

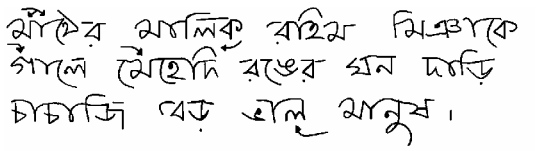
5. Results and discussion

We applied our proposed technique on some sample text one of which is shown in Fig 8.a. Fig 8.b shows the synthesized version. For the given text θ is approximately zero; $\overline{X_L}$ = 1, σ_{XL} = 14.14; $\overline{Y_L}$ = 83.5, σ_{YL} = 9.19; $\overline{\phi}$ = 3.7 degrees, σ_{ϕ} = 0.12; $\overline{Y_U}$ = 17.43, σ_{YU} = 9.72. Moreover, average inter word space is 60.42 and standard deviation is 7.47. We randomly selected this space and as a consequence inter-line and inter-word spaces have been changed a little bit. In Fig 8 b), first word in the first line displays small change in character-shape and continuation of characters using splines. Some characters have more than one instance and we select these instances randomly while synthesis. Second word in the second line shows instances of characters and modifiers, which are repeated. For some characters we change the size by a small amount to incorporate variation. This is shown in the last character of the third word in the third line. The first character of the first word of the second line clearly points out the uniformity of thickness (t_v) that is obtained by black run length and used here.

We took handwriting samples from fifteen persons to synthesize their styles. Each of them was presented the synthesis of a new page of text in his/her style and was asked to rank the result in one of four categories namely very good, satisfactory, tolerable and bad. Nine persons considered the machine output very good, four persons considered the result satisfactory while two persons considered the result tolerable. These two persons used to write in highly cursive manner and have variable character shapes and



a)



b)

Fig 8 a) original text b) its synthesized version (arrows show the change in characters)

inter-character transition that depends on the character pair concerned. The training sample that we took from them did not contain all such shapes and hence the synthesizer was unable to generate them in a desired way.

6. References

[1] C V Jawahar and A Balasubramanian, Synthesis of online handwriting in Indain languages, *Proc. Workshop on Frontiers of Handwriting Recognition*., pp. 21-26, 2006.

[2] D-H Lee and H-G Cho, A new synthesizing method for handwriting Korean scripts., *Int J Pattern Recognition and Artificial Intelligence*, Vol 12, No 1, pp. 46-61, 1998.

[3] H S Baird, The state of the art of Document image degradation modeling.,in B B Chaudhuri *Ed Digital Document Processing*, Springer Verlag, London, pp 261-279, 2007.

[4] I. Guyon, Handwriting synthesis from handwritten glyphs., *Proc Int Workshop on Frontiers of Handwriting Recognition*., pp. 140-253, 1996.

[5] J. Wang, C. Wu, Y-Q Xu, H-Y Shum and L Ji, Learning-based cursive handwriting synthesyis., *Proc. Workshop on Frontiers of Handwriting Recognition*., pp. 157-162, 2002.

[6] L.R.B. Schomaker, A.J.W.M Thomassen, and H.L. Teulings, "A computational model of cursive handwriting," in *Computer recognition and Human Production of Handwriting*, pp. 119-130, 1989.

[7] Ondrej Velek, Cheng-Lin Liu, and Masaki Nakagawa, "Generating realistic kanji character images from on-line patterns," in *proc. Of ICDAR*, pp. 556-560, 2001.

[8] S Srihari, H Cha, S-H Arora and S. Lee, Individuality of handwriting, *Journal of Forensic Sciences*, Vol, 47, No 4, pp. 1-17, 2002.

[9] S. Setlur and V Govindaraju, Generating manifold samples from a handwritten word, *Pattern recognition*, Vol. 15, pp. 901-905, 1994.

[10] T Varga and H Bunke, Generation and use of synthetic trining data in cursive handwriting recognition, *Proc 1st Iberian Conf. On Pattern Recognition and Image Analysis*., pp 336-345, 2003.

[11] Wacef Guerfali and R. Plamondon, "The delta lognormal theory for the generation and modeling of cursive characters ", in *Proc. Of ICDAR*, pp. 495-498, 1995.

[12] Y. Singer and N. Tishby, "Dynamical encoding of cursive handwriting," in *Proc. IEEE Conf. CVPR*, pp. 341-346, 1993.

[13] Y. Zheng and D Doermann, Handwriting matching and its application to handwriting synthesis, *Proc. Int. Conf. Document analysis and Recognition*., pp 861-865, 2005.