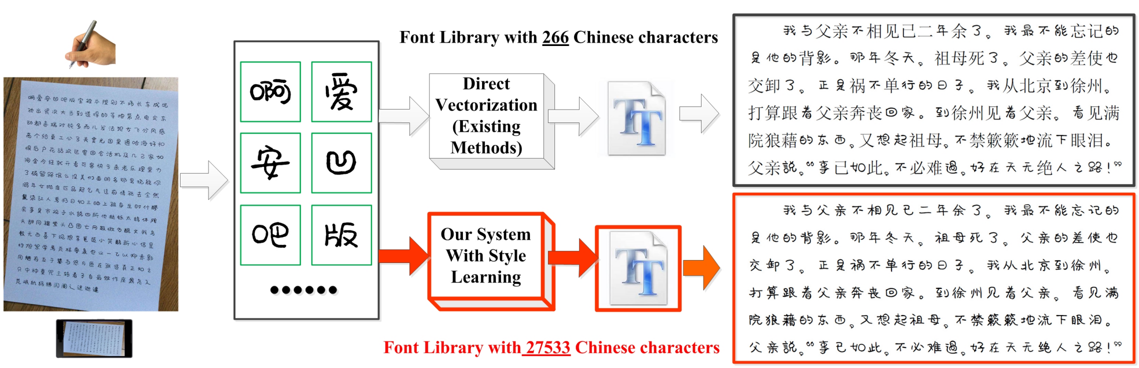
Automatic Generation of Large-scale Handwriting Fonts via Style Learning

Zhouhui Lian* Bo Zhao Jianguo Xiao
Institute of Computer Science and Technology, Peking University, PR China


Font Library with 266 Chinese characters
啊
爱
安
凹
吧
版
••••••
Direct Vectorization (Existing Methods)
Our System With Style Learning
Font Library with 27533 Chinese characters
我与父亲不相见已二年余了。我最不能忘记的
是他的背影。那年冬天。祖母死了。父亲的差使也
交卸了。正是祸不单行的日子。我从北京到徐州。
打算跟着父亲奔丧回家。到徐州见着父亲。看见满
院狼藉的东西。又想起祖母。不禁簌簌地流下眼泪。
父亲說。"事已如此。不必难過。好在天无绝人之路！"
我与父亲不相见已二年余了。我最不能忘记的
是他的背影。那年冬天。祖母死了。父亲的差使也
交卸了。正是祸不单行的日子。我从北京到徐州。
打算跟着父亲奔丧回家。到徐州见着父亲。看见满
院狼藉的东西。又想起祖母。不禁簌簌地流下眼泪。
父亲說。"事已如此。不必难過。好在天无绝人之路！"

(a) Input (b) Preprocessing (c) Font Generation (d) Text Rendering

Figure 1: *Overview of the proposed system and comparison of our approach with an existing font generation method. Using our system, the user only needs to write a small number (e.g., 266) of characters on a blank paper and uploads the photo to our web site. The system can then automatically generate a handwriting font library in the user's personal style with huge amounts (e.g., 27533) of Chinese characters. The paragraph rendered using the font library generated by our system looks exactly as a text written by the user. While, the one rendered by the other font library is even unreadable, due to the mixture of human-written samples and characters rendered in the default font style.*

Abstract

Generating personal handwriting fonts with large amounts of characters is a boring and time-consuming task. Take Chinese fonts as an example, the official standard GB18030-2000 for commercial font products contains 27533 simplified Chinese characters. Consistently and correctly writing out such huge amounts of characters is usually an impossible mission for ordinary people. To solve this problem, we propose a handy system to automatically synthesize personal handwritings for all characters (e.g., Chinese) in the font library by learning style from a small number (as few as 1%) of carefully-selected samples written by an ordinary person. Experiments including Turing tests with 69 participants demonstrate that the proposed system generates high-quality synthesis results which are indistinguishable from original handwritings. Using our system, for the first time the practical handwriting font library in a user's personal style with arbitrarily large numbers of Chinese characters can be generated automatically.

Keywords: handwriting, Chinese, style learning, Fonts

Concepts: •**Computing methodologies** → **Shape modeling;**

1 Introduction

Computer fonts are widely used in our daily lives. Now, texts displayed in almost all books, posters, computers, mobile devices, etc. are rendered using various fonts mainly created by professional companies. Although the number of font products increased rapidly in last two decades, existing resources still can not satisfy the demand of every individual since more and more people want to render texts in their own handwriting styles which are unique and full of personal information.

However, building a handwriting font library with large numbers of different characters is not easy. It is not a problem for writing systems (e.g., English) that only contain a few alphabets. Yet, the task becomes tougher when the number of characters included in the font library increases. Take Chinese fonts as an example, the official character set GB18030-2000 consists of 27533 simplified Chinese characters. What's more, shapes and structures of many Chinese characters are very complicated. As we know, to be a qualified font library, not only the glyph of each character should represent correct meanings, but also the style of all glyphs must be consistent. According to a report made by FounderType [Founder 2015], a leading Chinese font producing company, it takes more than 12 months for a group with 3-5 experienced font designers to generate a GB18030-2000 handwriting font library. Thereby, building complete Chinese fonts in their own personal handwriting styles is usually an impossible mission for ordinary people.

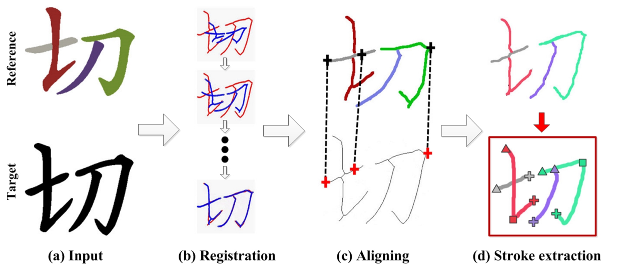
Currently, font designers still rely heavily on personal experiences and manual operations to generate commercial font products with the help of some typeface editing software (e.g., FontCreator [FontCreator 2015] and FontLab [FontLab 2015]). Due to the complexity and particularity of Chinese characters, some companies have developed their own font designing systems to create Chinese fonts other than directly using universal typeface editing software. For instance, HAND, a typeface editing system developed by FounderType [Founder 2015], was specially designed for Chinese fonts by taking unique characteristics (e.g., hierarchical represen-

*Corresponding author. Email: lianzhouhui@pku.edu.cn. This work was supported by National Natural Science Foundation of China (Grant No.: 61472015, 61672056 and 61672043), National Hi-Tech Research and Development Program (863 Program) of China (Grant No.: 2014AA015102) and Beijing Natural Science Foundation (Grant No.: 4152022)

SA '16 Technical Briefs ,, December 05-08, 2016, , Macao
ISBN: 978-1-4503-4541-5/16/12...$15.00
DOI: http://dx.doi.org/10.1145/3005358.3005371

tations) of Chinese characters into account. Indeed, these CAD systems could markedly improve the efficiency of font designing, but lots of times and manual operations are still required to create a commercial Chinese font library. During last two decades, several work that tried to reduce the heavy manual operations by introducing more heuristic rules and automatic processing into the font producing procedure have been reported [Fan 1990] [Lai et al. 1996] [Lian and Xiao 2012] [Lin et al. 2014] [Campbell and Kautz 2014], but these methods are still far from practical uses. Recently, [Suveeranont and Igarashi 2010] presented a system to automatically generate all characters in the font library based on the shape of a single character designed by the user. [Phan et al. 2015] proposed a system called FlexyFont that applies a machine learning method as well as retrieval-based methods to generate the whole font library by giving a small number of samples. However, these two systems both need to manually create the accurate and complex shape models for all training characters, which is unsuitable for handling large-scale font libraries that contain huge amounts of characters with complicated shapes. More recently, [Lake et al. 2015] presented a concept learning method using probabilistic program induction which is able to produce new exemplars in novel writing styles given a single character image. But, their method is unable to handle the tough problem of generating handwritings in the same style as input samples for all other unseen characters. The most relevant work was reported in [Zhou et al. 2011], where Zhou *et al.* developed a system to construct the shapes of 2500 relatively simple Chinese characters by reusing radicals of 522 characters written by a user, and thus built a small-scale Chinese font library in the user's handwriting style. However, limitations in scalability and efficiency hinder the practical uses of their system.

This paper aims to solve the challenging problem of automatic generation of large-scale handwriting font libraries. As shown in Figure 1, using our system to build a large-scale handwriting font library is convenient, the user only needs to write a small amount (as few as 1%) of carefully-selected Chinese characters on blank papers, take pictures and upload those photos to our web site [1]. After receiving these text images, a GB18030-2000 font library in the user's personal handwriting style can be automatically generated by our system in about four hours. Experiments including Turing tests with 69 participants demonstrate that the proposed system is able to accurately learn personal handwriting style, automatically synthesize indistinguishable handwritings, and quickly generate high-quality large-scale font libraries for any ordinary people. To the best of our knowledge, this is the first time that the handwriting font library in a user's personal style with arbitrarily large numbers of Chinese characters can be generated automatically.



(a) Input (b) Registration (c) Aligning (d) Stroke extraction

Figure 2: *Illustration of our stroke extraction algorithm. Key points (triangles, crosses, and squares denote start, end, and corner points, respectively) on each stroke can also be located (d).*

2 Method Description

As mentioned above, we intend to learn handwriting style from a small amount of characters written by an ordinary person, then automatically generate the whole handwriting font library which can have arbitrarily large numbers of characters in the user's personal style. More specifically, during offline processing period, we first manually specify the writing trajectory of each stroke for all (e.g., 27533) characters in the standard "Kaiti" font library which is employed as reference data of our system. Then, input character sets are properly chosen to satisfy requirements of different situations. Online, we first automatically extract stroke trajectories for individual character images segmented from input text photos. Then, we utilize Artificial Neural Networks (ANNs) to learn and reconstruct the user's overall handwriting style which can be decomposed into stroke shape style and stroke layout style. Meanwhile, handwriting details including stroke connectivity and shapes of contours are also properly described and recovered. Finally, a complete personal font library can be generated by vectorizing both images of human-written samples and machine-generated handwritings for all other characters. More details of our method are described below.

2.1 Selecting Input Character Set

In order to imitate a user's handwritings, our system needs to learn the handwriting style by analyzing some samples written by the user. Here comes a critical question that "which characters should be written?". Obviously, without "seeing" enough handwriting samples, neither our system nor even professional calligraphers could be able to precisely mimic the user's handwritings. So, one fundamental requirement of our system is that all types of strokes should appear at least once in the input character set. If better synthesis performance is required, we should select more characters to the input set so that all types of components can also be covered.

To make the font products generated by our system perform better in real text rendering applications (see Figure 1), we could like to have an input character set which is able to cover about 50% or more characters appear in all normal Chinese articles. By calculating the appearing frequency of each Chinese character in a data set we built with 87 billions characters, we obtain its average rate of coverage in a normal Chinese article. It can be observed after sorting coverage rates of all characters in descending order that theoretically the combination of first 190 characters is able to cover about 50% content of any normal Chinese article.

Finally, we can automatically determine the following two input character sets. The first one, which consists of 266 characters, serves as the minimum input character set (MinSet) of our system for practical uses. This input character set not only includes the above-mentioned 190 characters that have highest coverage rates, but also contains other 76 characters to ensure that all 339 categories of strokes can be written at least once. Adding the requirement of covering all kinds of components to the selection criterion adopted in the first case, we obtain another character set that is composed of 775 commonly-seen characters. We call it the optimal input character set (OptSet) of our system due to above-analyzed advantages and its high and robust performance in our experiments.

2.2 Stroke Extraction

After receiving the user's text images, the system automatically segments individual character images from those pictures by implementing region clustering and candidate filtering via some commonly-used heuristic rules. Then, with those individual character images, in order to know how the user wrote them, we must precisely locate the writing trajectory of each stroke on the charac-

[1] http://www.flexifont.com/flexifont/

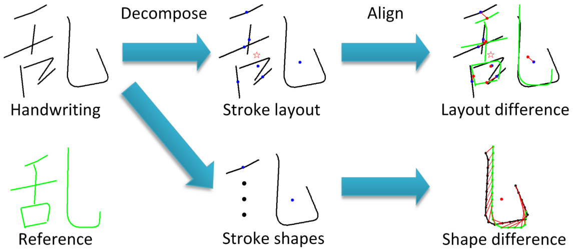


Figure 3: *Describing the handwriting style. Black and green trajectories denote handwriting and reference characters, respectively. Blue and red points are stroke centers of ref and $hand$. Pentacles denote character centers.*

ters. The key idea of our method is to utilize the Coherent Point Drift (CPD) [Myronenko and Song 2010] algorithm to implement non-rigid registration between the skeletons of reference and target character images (see Figure 2). Since strokes have been extracted manually from the reference data, after establishing correspondence between reference and target data, the writing trajectory of each stroke on the target images can be obtained automatically (see Figure 2(d)). More specifically, let $\mathbf{X}_{N\times 2} = (\mathbf{x}_1, ..., \mathbf{x}_N)^T$ be the target point set with N points evenly sampled on the skeleton of a user-written character, and $\mathbf{Y}_{M\times 2} = (\mathbf{y}_1, ..., \mathbf{y}_M)^T$ be the reference point set (i.e., Gaussian Mixture Models (GMM) centroids) with M points sampled on the corresponding reference character's writing trajectory. Registering the reference point set with target point set is equivalent to determining the locations of GMM centroids (i.e., θ) and the equal isotropic covariances of GMM distributions (i.e., σ^2) by minimizing the following objective function

$$E(\theta, \sigma^2) = -\sum_{n=1}^{N} \log(\sum_{m=1}^{M} \frac{1}{M}\frac{1}{2\pi\sigma^2} \exp^{-\frac{\|\mathbf{x}_n - \mathbf{y}_m\|^2}{2\sigma^2}} + \frac{1}{N}). \tag{1}$$

Here, we adopt the EM algorithm to solve this problem.

2.3 Overall Style Learning

In our system, the overall handwriting style is represented as the difference between trajectories of reference characters (ref) and handwritten characters ($hand$). We decompose the Chinese character into a lower-level concept and structure, namely the stroke shape (SS) and stroke layout (SL) (see Figure 3). Hence, the overall handwriting style can be decomposed into the stroke shape style (SSS) and stroke layout style (SLS), which are represented by the differences of stroke shapes and stroke layouts (DSS and DSL), respectively, between ref and $hand$.

To calculate the DSS and DSL, we first sample the same number (N_P) of points $P_{ij}(k) = (x_{ij}(k), y_{ij}(k)), k = 1, 2, ..., N_P$ on the trajectory of each stroke for all reference and handwritten characters. Thus, strokes can be represented as points along stroke trajectories (for simplicity, unless otherwise specified, in this section we directly use "stroke" to denote the "stroke trajectory"), e.g., $S_{ij} = (P_{ij}(1), P_{ij}(2), ..., P_{ij}(N_P))$. While, characters are represented as vectors of strokes $C_i = (S_{i1}, S_{i2}, ..., S_{iN_{Si}})$, where N_{Si} denotes the number of strokes in character C_i. Then, the stroke center SC and character center CC can be easily computed. To better describe the shape of a stroke, we calculate the normalized stroke shape, which consists of relative positions of points on the stroke to the stroke's mass center. Similarly, the normalized stroke layout can be obtained by calculating the relative positions of stroke centers to the character center. Thus, the normalized SS and SL for each stroke can be computed by $SS_{ij} = S_{ij} - SC_{ij}$ and $SL_{ij} = SC_{ij} - CC_i$, respectively. As mentioned above, the stroke shape style and stroke layout style are represented by DSS and DSL, which can be calculated as $DSS_{ij} = SS_{ij}^{hand} - SS_{ij}^{ref}$ and $DSL_{ij} = SL_{ij}^{hand} - SL_{ij}^{ref}$, where SS_{ij}^{hand}, SL_{ij}^{hand} denote the normalized SS, SL of a handwritten character, and SS_{ij}^{ref}, SL_{ij}^{ref} represent the normalized SS, SL of the corresponding reference.

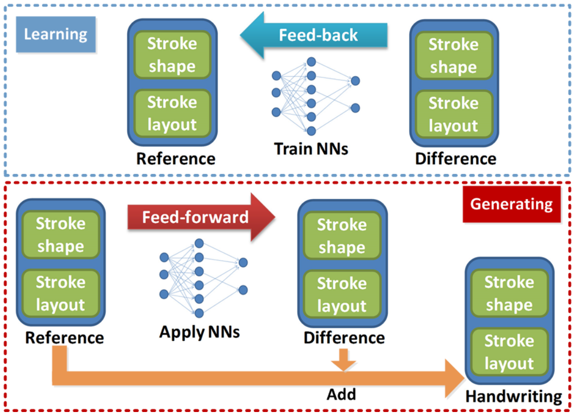


Figure 4: *Illustration of overall style learning and handwriting synthesis in our system.*

Figure 4 depicts an overview of how to learn and reconstruct the user's overall handwriting style using our method, which consists of the following two procedures: style learning and handwriting synthesis. In the style learning procedure, neural networks are utilized to capture overall handwriting style. Stroke shapes of ref serve as input for learning stroke shape style, and the difference of stroke shapes between ref and $hand$ serves as output. Similarly, when learning stroke layout style, the input is the stroke layout of ref and the output is the difference of stroke layouts between ref and $hand$. Here, we choose the stroke-wise data structure, which means that the input and output are SS_{ij} and DSS_{ij}, respectively. Artificial neural networks are well suited to find the difference (style) of strokes between ref and $hand$. In our system, we adopt the Feed-forward neural network (FFNN) to learn stroke layout style and the Stroke-wise Learning with FFNN (SWL-FFNN) to learn stroke shape style. Specifically, only one hidden layer is utilized in all NNs and the structures of all networks are experimentally chosen as $I * H * O = 40 * 40 * 40$, where I, H and O denote the numbers of units in input, hidden and output layers, respectively. In the handwriting synthesis procedure, new reference characters are input into trained NNs to estimate DSS_{ij} and DSL_{ij}. Then, we obtain SS_{ij}^{hand}, SL_{ij}^{hand} by applying DSS_{ij} and DSL_{ij} to SS_{ij}^{ref}, SL_{ij}^{ref}, i.e., $SS_{ij}^{hand} = DSS_{ij} + SS_{ij}^{ref}$ and $SL_{ij}^{hand} = DSL_{ij} + SL_{ij}^{ref}$. After setting the character center CC_i, positions of stroke centers and sampled points on strokes can be located using calculated SS_{ij}^{hand} and SL_{ij}^{hand}.

2.4 Recovering Handwriting Details

As described above, after capturing the user's overall handwriting style, writing trajectories of all characters can be easily generated. Then, the simplest way of creating synthesized handwritings is to render the trajectories with the average stroke width of handwritten characters. However, as shown in Figure 2(a), the stroke width of each point on a writing trajectory may change greatly according to different handwriting behaviors, especially in the start and end regions of a stroke. Moreover, the connectivity of two

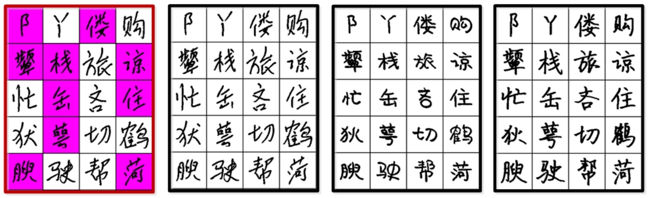

(a) Answer paper (b) Test - User 1 (c) Test - User 2 (d) Test - User 3

Figure 5: *Examples of a region of test papers in three handwriting styles and their corresponding answer paper (a) utilized in Turing tests. Characters in color blocks are generated by our system and others are human-written characters.*

sequential strokes is also an important feature of the user's handwriting style. To capture the handwriting details on the contour of a stroke, we emit a number of (e.g., 11) rays evenly distributed in the half region opposed to the writing direction to obtain the relative positions of points on a stroke's contour around the stroke trajectory's start point (same for the end point). In this way, details on the contour can be recovered when rendering the trajectory of a stroke that belongs to the same category. To describe the connecting property for each pair of sequential strokes, we calculate a 339 × 339 matrix M_c in which the element m_{ij}^c denotes the probability of drawing trajectory between the end point of stroke i and the start point of stroke j. When generating synthesis results, if the value of m_{ij}^c is larger than a random number P_c ($P_c \in [0, 1]$), a natural and smooth line with proper width values will be created to connect the end point of stroke i with the start point of stroke j.

3 Experiments

In our experiments, original handwritings of different users are collected through our personal font generation web site mentioned above. The algorithms are implemented in Matlab on a PC with a 3.5GHz Intel i7-5930K CPU and 32.0 GB RAM. The OptSet that consists of 775 characters is chosen as the input character set in this section. Although the MinSet with only 266 characters can also be adopted as the input character set, and in fact considerably good results (see Figure 1) are already obtained by using the MinSet, we still recommend utilizing the OptSet to guarantee better and more stable performance in real applications where strange, very cursive, and even incorrect handwritings might also be input to our system. Averagely it takes only about 20-30 minutes for an educated Chinese people to correctly write out all characters in the OptSet and about 4 hours for the system to automatically generate a GB18030-2000 font library in the user's handwriting style. We choose the handwriting data provided by three users (i.e., User 1, 2 and 3) of our web site that have quite different handwriting styles (see Figure 5 and supplementary materials) as the input data of our system.

To quantitatively measure the similarity of styles between human-written characters and synthesized handwritings generated by our system, Turing tests are conducted. Specifically, we build a web site that shows a random test paper (see Figure 5(b-d) for some examples of test papers and Figure 5(a) for the corresponding answer paper) for each participant, on which 100 machine-generated characters and 100 human-written characters in a user's personal style are randomly chosen and placed. Meanwhile, 50 randomly-chosen characters written by the user are also displayed to the participant as reference. Each participant is asked to pick out as many characters as possible, which they think are imitated by computers, with sufficient time. Obviously, if the writing style of a machine-generated character is different against original handwritings, it will be quite easy for educated Chinese people to find it. Meanwhile, a machine-generated character is hard to be picked out from the test paper if it not only looks like a human-written character but is also similar as the character written by the same person. Thereby, the Turing tests we conduct here can illustrate whether synthesis results possess the required personal handwriting style or not.

We invited 69 educated Chinese people with different ages (16-51) and occupations (e.g., students, teachers, etc.) to participate in our Turing tests via internet. The average accuracy of distinguishing machine-generated characters from original ones is 51.25%, which is close to the accuracy of random guess (50%). Results of our Turing tests verify that synthesized handwritings generated by our system are hard to be distinguished from the corresponding user's original handwritings. This is because, as shown in Figure 5, no only the overall style but also many important details of the users' handwritings can be well imitated by the proposed system.

4 Conclusion

This paper presented a novel system that is able to learn the handwriting style from a small number of input samples written by an ordinary person and generate the personal handwriting font library, which can have arbitrarily large numbers of Chinese characters, for the user. Experimental results demonstrated that our system can be used to automatically generate high-quality handwriting font libraries which include huge amounts of machine-generated characters that are indistinguishable from original handwritings.

References

Campbell, N. D., and Kautz, J. 2014. Learning a manifold of fonts. *ACM Transactions on Graphics 33*, 4, 91.

Fan, J. 1990. Interlligent chinese character design and an experimental system iccds. *JCIP 4*, 3, 1–11 (In Chinese).

FontCreator. 2015. High logic. *http://www.high-logic.com/*.

FontLab. 2015. Fontlab. *http://www.fontlab.com/*.

Founder. 2015. Founder group. *http://www.foundertype.com/*.

Lai, P.-K., Yeung, D.-Y., and Pong, M.-C. 1996. A heuristic search approach to chinese glyph generation using hierarchical character composition. *Computer Processing of Oriental Languages 10*, 3, 307–323.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. 2015. Human-level concept learning through probabilistic program induction. *Science 350*, 6266, 1332–1338.

Lian, Z., and Xiao, J. 2012. Automatic shape morphing for chinese characters. In *Proc. SIGGRAPH Asia 2012 TB*, 2.

Lin, J.-W., Wang, C.-Y., Ting, C.-L., and Chang, R.-I. 2014. Font generation of personal handwritten chinese characters. In *Proc. IGIP 2014*.

Myronenko, A., and Song, X. 2010. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 12, 2262–2275.

Phan, H., Fu, H., and Chan, A. 2015. Flexyfont: Learning transferring rules for flexible typeface synthesis. *Computer Graphics Forum 34*, 7, 245–256.

Suveeranont, R., and Igarashi, T. 2010. Example-based automatic font generation. In *Proc. Smart Graphics*, 127–138.

Zhou, B., Wang, W., and Chen, Z. 2011. Easy generation of personal chinese handwritten fonts. In *Proc. ICME 2011*, 1–6.