<table>
<thead>
<tr>
<th>Title</th>
<th>Automatic generation of Chinese calligraphic writings with style imitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Xu, S; Jiang, H; Jin, T; Lau, FCM; Pan, Y</td>
</tr>
<tr>
<td>Citation</td>
<td>IEEE Intelligent Systems, 2009, v. 24 n. 2, p. 44-53</td>
</tr>
<tr>
<td>Issued Date</td>
<td>2009</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10722/152438">http://hdl.handle.net/10722/152438</a></td>
</tr>
<tr>
<td>Rights</td>
<td>IEEE Intelligent Systems. Copyright © IEEE, Computer Society; ©2009 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE; This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.</td>
</tr>
</tbody>
</table>
An automatic algorithm can generate Chinese calligraphy by quantitatively representing the characteristics of personal handwriting acquired from learning examples.

Handwritten calligraphy is one of the master art forms of Chinese, Korean, Japanese, and Vietnamese culture. Nowadays, however, few people practice handwriting in their daily life, let alone calligraphy. We propose an automatic method to generate calligraphy, with the hope that it may help rekindle interest in traditional Chinese calligraphy in the digital age. For instance, when someone needs calligraphic writing for decoration, such as on walls, clothes, gifts, monuments, statues, or personalized communications, automatically generated calligraphy would provide a clear alternative to mass-produced characters. With an automatic algorithm and instant results, anyone can create calligraphic art, which in turn could educate a wide range of users. For young people in particular, an interactive digital tool on their favorite electronic devices might prove the best way to interest them in calligraphy. We hope the increased use of intelligent tools like ours will heighten the appreciation of Chinese calligraphic art in the future.

Most work on computerized calligraphic handwriting synthesis has focused on English or Latin characters. Automatic imitation of Chinese calligraphic writing is particularly challenging because the Chinese character set is many times larger than English and the writing styles for Chinese characters is much more diversified owing to the complexity of character composition. C.V. Jawahar and A. Balasubramanian studied the synthesis of Indian script in handwriting style. They employed a simple stroke and layout model to successfully reproduce the handwriting of multiple Indian languages. We have adopted a more elaborate model to capture Chinese calligraphic stroke shapes with high fidelity. Since Chinese characters’ shapes are very different from those of Indian characters, we need to introduce a different topology model and an additional stability model to capture the handwriting characteristics of Chinese calligraphers. For other approaches, see the sidebar “Related Research on Handwriting Generation” on page 49.

Representing Stroke Shapes
Strokes are the building blocks of characters; a writer composes a character by writing in-
individual strokes. Therefore, to capture the shape aspect of personal handwriting, we first need to represent the shapes of individual strokes.

In our method, we derive a parametric representation of stroke shapes, which otherwise are often stored as images. Our parametric representation can make the pattern analysis and synthesis operations in the later stages of our system easier and more efficient. For each image of a stroke, we apply a thinning algorithm to extract the skeleton of the stroke, which is a single pixel-wide trajectory. At each pixel of the skeleton, we draw a line following the local normal direction of the skeleton. The length of the line within the stroke’s interior region is treated as the local width of the stroke.

We adopt a shape-generation-based process to compactly represent the shapes of single strokes. Figure 1 illustrates how to generate a stroke shape based on other stroke shapes. Such a process helps us efficiently represent stroke shapes—that is, we only represent the truly unique shapes, from which we can generate others. This can save tremendous storage space since most writers’ stroke shapes are not that different. For the same person, stroke shapes vary even less.

**Representing Character Topology**

Another essential piece of handwriting information consists of the topological relationship demonstrated in a character’s composition, or simply, its topology—that is, the way the shape of a character is formed through proper positioning of individual strokes. Inspired by the bounding-box-based method for describing spatial relationships between multiple strokes, we represent a character’s topology according to the spatial relationships between the bounding boxes of all the character’s strokes. More concretely, for a certain character C, assume it consists of n strokes S₁, S₂, ..., Sₙ. We denote stroke Sᵢ’s bounding box as Bᵢ. For each pair of strokes Sᵢ and Sⱼ in C, we introduce two features, fₓ(Sᵢ, Sⱼ) and fᵧ(Sᵢ, Sⱼ), to represent the horizontal and vertical overlapping between Bᵢ and Bⱼ. On the basis of these two features, we can construct two matrices Fₓ and Fᵧ, both of dimensionality n x n, to capture the spatial relationships among all strokes in character C.

In much the same way as we represent stroke shapes through a shape generation process, we also represent character topology via a topology generation process to achieve a storage-optimized representation. Basically, we generate a character’s topology through weighted averaging of a few other characters’ topologies. Since we represent a character’s topology in matrix form, we can realize such a weighted-average-based character topology generation process via standard matrix multiplication and addition. Figure 2 (see next page) illustrates how to generate a new character topology via weighted averaging of a few given character topologies.

**Representing Personal Handwriting Characteristics**

One important difference between personal handwriting and a script generated from a font system is that a human writer would very likely write a certain stroke or character differently each time, while a font system will always generate the same output. We’ll examine how to represent such characteristics of personal handwriting in this section, which consists of representing the shape and topology aspects of personal handwriting stability.

**Measuring Pairwise Shape Distance**

Given two stroke shapes S₁ and S₂, we employ an image-processing-based approach to derive the distance between them. We use I₁ and I₂ to represent the image regions occupied by the two stroke shapes respectively. To keep the computation cost manageable, the approach we use to measure the shape distance is simple but effective. Specifically, we define the distance measure as
where $A(X)$ is the area of the region $X$, and $T(I_2)$ applies the optimal transform to align $I_2$ with $I_1$ to maximize their overlapping area. $\theta^{\text{sha}}(S_1, S_2)$ ranges between 0 and 1. It is 0 when the two stroke shapes are identical. This optimal transform comes from a gradient descent search that applies the optimal translation, rotation, and scaling transformations to maximize the overlapping area between the two stroke shapes. Unlike many shape comparison approaches that compare contours, our shape-based criterion makes direct use of areas, which is more reliable than using shape contours because stroke contour details vary greatly.

![Image](https://example.com/image.png)

Figure 2. Generation-based character topology representation: (a) four sample characters, (b) the stroke bounding boxes indicating those characters’ composition topology, and (c) using the four composition topologies to generate a new topology. In this example, $\gamma_1 = 0.34$, $\gamma_2 = 0.09$, $\gamma_3 = 0.14$, and $\gamma_4 = 0.43$.

**Representing Stability in the Shape Aspect**

Generally, when a writer writes a stroke $S_t$ times, he or she creates $t$ different stroke shapes $S_i (i = 1, ..., t)$. To represent the writer’s shape aspect of handwriting stability in writing stroke $S$, we first find the largest pairwise distance among the $t$ shapes above, assumed to be $\text{Dis}_{\text{max}}^{\text{sha}}$. Then, for each stroke shape $S_i$, we compute its shape stability factor $\phi^{\text{sha}}(S_i)$ as

$$\phi^{\text{sha}}(S_i) = \frac{\phi^{\text{sha}}(S_i)}{\sum_{j=1}^{t} \phi^{\text{sha}}(S_j)},$$

where

$$\phi^{\text{sha}}(S_i) = \frac{\sum_{j=1}^{t} 1}{\sigma \sqrt{2\pi}} \exp \left( \frac{\theta^{\text{sha}}(S_i, S_j) - 2\sigma^2}{-2\sigma^2} \right)$$

and $\sigma$ is $0.1\text{Dis}_{\text{max}}^{\text{sha}}$. The more a person writes stroke $S$ in a certain shape $S_i$, the larger $\phi^{\text{sha}}(S_i)$ becomes.

**Measuring Pairwise Topology Distance**

For two topology relationships $T_1$ and $T_2$, we can derive the pairwise topology distance between them, denoted as $\theta^{\text{topo}}(T_1, T_2)$, according to their matrix representations. Let $F_x(T_1)$, $F_y(T_1)$ and $F_x(T_2)$, $F_y(T_2)$ be $T_1$ and $T_2$’s respective matrix representations using the method introduced in the section “Representing Character Topology.” We can then define $\theta^{\text{topo}}(T_1, T_2)$ as

$$\theta^{\text{topo}}(T_1, T_2) \triangleq \left\| F_x(T_1) - F_x(T_2) \right\| + \left\| F_y(T_1) - F_y(T_2) \right\|$$

Here, $\| \cdot \|$ is the Euclidean matrix norm.

**Representing Stability in the Topology Aspect**

It is also true that when a person writes a character multiple times, he or she will create different topological relationships of the character’s composition. We assume $T_1$, $T_2$, ..., $T_t$ are multiple copies of a character’s composition topology as demonstrated in a writer’s previous handwriting. To define the writer’s topology aspect of handwriting stability, we find the largest pairwise distance among the topology, denoted as $\text{Dis}_{\text{max}}^{\text{topo}}$. Anogous to the definition for the shape aspect of personal handwriting stability, for each topology relationship $T_i (i = 1, ..., t)$, we compute its
stability factor $\theta^{\text{topo}}(T_i)$ as

$$\phi^{\text{topo}}(T_i) = \sum_{j=1}^{i} \hat{\phi}^{\text{topo}}(T_j),$$

where

$$\hat{\phi}^{\text{topo}}(T_i) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left( \frac{\theta^{\text{topo}}(T_i, T_j)}{-2\sigma^2} \right),$$

and $\sigma$ is $0.1D_{\text{topo}}$. The more a person writes a character with a certain topology, the higher the stability for the topology relationship becomes.

**Acquiring Personal Handwriting Characteristics**

As discussed in the previous section, we derive the shape and topology aspects of personal handwriting characteristics according to either the shapes of, or spatial relationships among, individual strokes. Thus, to acquire the characteristics of personal handwriting, we first decompose a character into single strokes by adopting the algorithm proposed by Songhua Xu and his colleagues.\(^2\) Source characters for decomposition come from both personal calligraphy and commercial font systems.

We also must determine whether a person has written the same stroke multiple times. To do this, we associate each decomposed stroke with a property, representing the stroke’s category as determined by conventional Chinese character formation methodology. As Jyh-Yeong Chang and Min-Hwa Wan demonstrated, we can classify strokes largely according to the shapes of their skeletons.\(^3\) We consider stroke samples sharing the same property as multiple samples of the same stroke. Thus we can determine whether a person has written the same stroke multiple times by counting whether multiple samples are classified in the desired category. In our current prototype system implementation, we support 50 of the most frequently used stroke types in Chinese character formation.

Once the training data have been prepared, we use the methods introduced in the previous section to capture individual writers’ personal handwriting characteristics.

**Reasoning on Personal Handwriting Characteristics**

We can use the knowledge of personal handwriting characteristics to reproduce Chinese calligraphy through case-based reasoning. We choose this particular application because of the practical interest and challenges the problem presents—a task that is fairly demanding even for a human. Basically, to reproduce writer W’s handwriting over character C, denoted $C_W$, we infer the character’s topology and the shapes of all its constituent strokes. We then assemble the strokes into a calligraphic rendition of the character. In the reproduction process, we use the font style Kai, defined in China’s national font standard as GB2312, to quantitatively measure the variation of the handwriting style with respect to a standard font. As an overview of the reproduction process, an algorithmic diagram is shown in Figure 3 together with a simplified example in Figure 4 (see next page).

**Determining a Target Character’s Topology**

To reproduce $C_W$, we first extract the topology of the character in its standard font $C_W^{\text{std}}$, represented as

$$\left( E_x^{C_W^{\text{std}}}, E_y^{C_W^{\text{std}}} \right),$$

using the method introduced in the “Representing Character Topology” section. Then, our system retrieves all
characters writer $W$ has previously written. Among them, we discard all characters that contain a different number of strokes than $CW$. We assume after the pruning process that $o$ characters remain, which are denoted as $C_{W}^{1}, C_{W}^{2}, ..., C_{W}^{o}$, with their character composition topology derived and represented as

$$\left\{ F_{x}^{C_{W}^{1}}, F_{y}^{C_{W}^{1}} \right\}, \left\{ F_{x}^{C_{W}^{2}}, F_{y}^{C_{W}^{2}} \right\}, ..., \left\{ F_{x}^{C_{W}^{o}}, F_{y}^{C_{W}^{o}} \right\}.$$ 

Note that each of the matrices must have the same dimensions as $F_{x}^{CW}\text{std}$ and $F_{y}^{CW}\text{std}$.

Then we can derive the distance between the topology of the $i$th character, $C_{W}^{i}$, and that of $C_{W}\text{std}$ as

$$\xi_{i} \triangleq \left\| F_{x}^{C_{W}^{i}} - F_{x}^{C_{W}\text{std}} \right\| + \left\| F_{y}^{C_{W}^{i}} - F_{y}^{C_{W}\text{std}} \right\|.$$ 

In this topology distance measurement process, we also account for all possible stroke combination options where adjacent strokes are merged into a single stroke, as is often the case in calligraphic writings. We then identify five characters that bear the five most similar topology relationships to that of $C_{W}\text{std}$ in terms of the smallest topology distances. We denote them as $C_{W}^{1}, C_{W}^{2}, ..., C_{W}^{5}$, and their respective overall topology distances to $C_{W}\text{std}$ as $\xi_{1}, \xi_{2}, ..., \xi_{5}$. Each of the five topology relationships is associated with a handwriting stability factor as derived in the subsection “Representing Stability in the Topology Aspect.” We denote them as $\phi_{1}^{\text{topo}}, \phi_{2}^{\text{topo}}, ..., \phi_{5}^{\text{topo}}$ respectively. To determine the topology of target character $C_{W}$, we randomly choose one of the five characters’ topologies, with the probability of choosing the $i$th character’s topology being $\nu_{i}$. We calculate $\nu_{i}$ as

$$\nu_{i} = \frac{\phi_{i}^{\text{topo}}}{\xi_{i} + \epsilon} \sum_{j=1}^{5} \frac{\phi_{j}^{\text{topo}}}{\xi_{j} + \epsilon},$$

where $\epsilon$ is a small positive number to avoid the divide-
Many research results exist on handwriting recognition. The problem we study in the main article is the reverse problem of handwriting generation—we want to generate characters in one’s handwriting style as precisely as possible. Recently, Ying Wang and her colleagues contributed an algorithm for Chinese handwriting synthesis that’s a degenerate case of our algorithm: their algorithm did not model the habitual variations of personal handwriting.¹ We introduce a statistical shape-modeling method for capturing the character of personal handwriting. By modeling the uncertainties in personal handwriting, we can produce the most faithful facsimile of a person’s handwriting.

A body of work exists for solving the computational problems arising from processing calligraphic characters. Although the pioneering work by Donald Knuth on TeX and Metafont was not directly dedicated to handwritten-calligraphy processing, many of the modeling and computational solutions in his approach serve as a good reference for much of the later work in the area.² Pak-Keung Lai and his colleagues studied the problem of numerically evaluating the beauty of calligraphic characters through a heuristic approach.³ Toshinori Yamasaki and Tetsuo Hattori tackled the Japanese calligraphy generation problem by composing calligraphic characters from fundamental brush strokes in a hierarchical fashion; their work provides a good reference for the calligraphic-generation component in our work.⁴ Junji Mano and his colleagues utilized fuzzy spline curves to generate Japanese character calligraphy through an interpolation-based approach.⁵ A recent piece of work related to our article is an intelligent system for Chinese calligraphy generation capable of generating novel calligraphic writings based on a few parameterized calligraphic samples learned by the computer.⁶ This generation process is realized through a constraint-based spatial-reasoning process. In contrast, the focus of our research here is to capture and mimic the appearance of Chinese calligraphy through representing and reasoning about a certain writer’s personal handwriting characteristics. We acquire such knowledge from exploiting the underlying relationships existing in the writer’s calligraphic characters.

Our calligraphic-writing-generation work is also related to shape generation using generative models. Aleksandr Dubinsky and Song Chun Zhu introduced a multiscale generative model for shape generation aimed at animation applications.⁷ Recently, Nhon Trinh and Benjamin Kimia devised a symmetry-based generative model for shape generation, which can generate an arbitrary shape with relatively few parameters.⁸ But they are not concerned with how to constrain the generation parameters to achieve style imitation during the shape generation process. People have also used generative models for recognition. As exemplified by Michael Revow and his colleagues, a body of literature discusses methods for handwriting recognition by first capturing the handwriting styles or allographic shape variations, such as through clustering or by building generative models.⁹ Such a phase is relevant to the handwriting synthesis work studied in this article. Most recently, Songhua Xu and his colleagues proposed an interpolation-based approach for producing calligraphic writings. Compared with the algorithm we introduce in this article, their algorithm is more costly computationally, thus making it difficult for real-time applications or deployment on devices with limited computing power, such as cell phones and PDAs.¹⁰

**References**


**Determining a Target Character’s Stroke Shapes**

To determine the shape of a stroke we will reproduce, we need to introduce the concept of stroke context. For a certain character’s stroke S, we first find its nearest four strokes, denoted S₁, S₂, ..., S₄. Here the distance between two strokes is defined as the shortest distance between two points, one on
each stroke. We then construct two $1 \times 4$ vectors,

$$V_x(S) = \left[ f_x(S, S_{1}^w), \ldots, f_x(S, S_{4}^w) \right],$$

and

$$V_y(S) = \left[ f_y(S, S_{1}^w), \ldots, f_y(S, S_{4}^w) \right].$$

to describe the local topology context around stroke $S$ using the functions $f_x$ and $f_y$ introduced in the section “Representing Character Topology.” Given these two context description vectors, we can measure the contextual difference between two strokes $S_1$, $S_2$ as

$$\theta_{vx}(S_1, S_2) = \| V_x(S_1) - V_x(S_2) \| + \| V_y(S_1) - V_y(S_2) \|.$$  

With this metric, we can further define the overall difference between strokes $S_1$ and $S_2$ as $\theta_{v}(S_1, S_2) = \theta_{vx}(S_1, S_2^o) + \theta_{vsha}(S_1, S_2)$ using $\theta_{vsha}(\cdot, \cdot)$, as introduced in the subsection “Measuring Pairwise Shape Distance.”

With the metric $\theta_{v}(\cdot, \cdot)$, we can determine the shapes for the strokes constituting the target character $C_{w}$, again through reference to $C_{w}^{std}$. We assume that $C_{w}^{std}$ contains $n$ strokes—that is, $S_{1}^w$, $S_{2}^w$, ..., $S_{n}^w$. For each stroke $S_{i}^w$, we look for all the strokes created by writer $w$ that have the same stroke type as $S_{i}^w$ (see the section “Acquiring Personal Handwriting Characteristics” regarding stroke type property). Assume that the retrieved strokes are $S_{i}^1$, $S_{i}^2$, ..., $S_{i}^o$. We choose five strokes among them that yield the smallest overall distances to $S_{i}^w$ using the metric $\theta_{v}(\cdot, \cdot)$. We denote their respective distances as $\theta_{i}^{v}$, $\theta_{i}^{vsha}$, ..., $\theta_{i}^{vsha}$. Using the procedure introduced in the subsection “Representing Stability in the Shape Aspect,” we can also derive these stroke shapes’ writing stabilities as $\phi_{i}^{vsha}$, ..., $\phi_{i}^{vsha}$. Finally, to determine the shape of stroke $S_{i}^w$, we randomly choose from one of the above five stroke shapes with the probability of $\mu_{i}$ to choose the stroke shape $i$. We calculate $\mu_{i}$ as

$$\mu_{i} = \frac{\phi_{i}^{vsha}}{\theta_{i}^{v} + \epsilon} \sum_{j=1}^{5} \frac{\phi_{j}^{vsha}}{\theta_{j}^{v} + \epsilon}.$$  

In this way, we determine the shape of the stroke by identifying and randomly reusing the contextually most similar stroke created by the writer. Through this procedure, we determine all stroke shapes needed for composing target character $C_{w}$. We then assemble them according to the character composition topology determined in the subsection “Measuring Pairwise Shape Distance.”

**Experiment Results and User Evaluation**

To experiment with our algorithm, we developed a prototype system for reproducing calligraphic characters. Figure 5 analyzes the behavior of our algorithm by comparing the deviation of the top three results of our algorithm with authentic handwriting. Figure 6 shows the results of four calligraphic reproduc-
tion experiments. The characters by the four calligraphists are shown along with the respective results produced by our algorithm. In these four experiments, only 24 other characters written by the respective calligraphists are accessible to our algorithm for knowledge acquisition. Our facsimile results are not identical to the ones written by the calligraphers, which is not surprising. Nevertheless, for many characters, the results appear fairly close to the authentic handwriting.

Figure 7 shows the results of yet another experiment imitating a famous 11th-century Chinese poem. Owing to space constraints, more experimental results are available on our project Web site (www.cs.hku.hk/~songhua/facsimile).

For a more objective evaluation, we also carried out a medium-scale user survey with 100 Chinese university students. We devised an online user survey (available on our project Web site) and first showed them 20 characters of original handwriting by a calligrapher. Then we showed them 50 characters, saying that approximately half were authentic handwriting and half were created by computer. We asked them to try to identify the reproductions. The average accuracy of this “Turing test” survey achieved by our subjects is 52.58 percent. Considering that a purely random guess would achieve an accuracy of 50 percent, such an accuracy confirms the high quality of our machine reproduction results, which we find encouraging. We show the characters used in the survey as well as our subjects’ performance in Figure 8 (see page 53).

We currently choose sequences of strokes at random from sample characters created by a writer, assuming stroke independence. However, this sequence may not correspond with the order in which the writer normally produces strokes. This could be an issue if a person writes hastily and adjacent strokes get connected so that shapes and ink distributions would differ if written in a different order. But in this work, we mostly focus on those handwriting styles that are done in a slow and careful manner, which largely exhibit a strong independence between stroke orders. In the future, when we extend our work to imitate cursive styles, our stroke order independence assumption might lead to failures and would need new algorithmic innovations to fix the problem.

Knowledge of personal handwriting characteristics, once acquired and properly represented, enables us to try a number of interesting applications. Other potential applications include writer authentication, personal handwriting tutoring, handwriting beautification, and even personal emotion detection according to one’s handwriting.
References


For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.

Figure 7. Reproducing a personal handwriting rendition of a famous 11th century Chinese poem: (a) Characters written by a modern calligrapher are used as learning examples by our algorithm. (b) Facsimile results of our algorithm. (c) We show the original handwriting over these characters as ground truth data for comparison. Notice that the original characters are included in the testing set, for example, (II-21) and (II-22); our algorithm automatically reuses them. On the basis of this experiment, we conducted a medium-scale online user study to evaluate how well a person could tell our machine imitated results from the original handwriting. Results of this study are reported in Figure 8.
The Authors

Songhua Xu is a research associate at Zhejiang University’s College of Computer Science and Technology, an honorary assistant professor and an honorary researcher at the University of Hong Kong’s Department of Computer Science, and a graduate research fellow at Yale University’s Pathology Informatics program. His research interests include AI, computer graphics, human-computer interaction, information retrieval, and biomedical informatics. Xu received his BEng summa cum laude in computer science and technology from Zhejiang University. Contact him at songhua.xu@yale.edu.

Hao Jiang is a postgraduate student in the University of Hong Kong’s Department of Computer Science. His research interests include AI, computer graphics, the Web, and robotics. Jiang received his MEng in computer science and technology from Zhejiang University. Contact him at hjiang@cs.hku.hk.

Tao Jin is a research assistant in the University of Hong Kong’s Department of Computer Science. His research interests include AI, computer graphics, the Web, and information retrieval. Jin received his MEng in computer science and technology from Zhejiang University. Contact him at tjin@cs.hku.hk.

Francis C.M. Lau is a professor of computer science at the University of Hong Kong. His research interests include parallel and distributed computing, wireless networks, and AI. Lau received his PhD in computer science from the University of Waterloo. Contact him at fcmlau@cs.hku.hk.

Yunhe Pan is a professor at Zhejiang University, a fellow and deputy president of the Chinese Academy of Engineering, as well as an honorary professor at the University of Hong Kong. His research interests include AI, computer graphics, intelligent computer-aided design, and computer art. Pan received his MEng in computer science and engineering from Zhejiang University. Contact him at panyh@zju.edu.cn.

Figure 8. Visualization of the results of our user evaluation of the facsimile experiment reported in Figure 7. (a) Raw data recording each participant’s performance. Each row in the bitmap corresponds to a character, and each column corresponds to a user. A total of 100 Chinese users participated in this study, hence 100 columns. Each user was asked to tell the authenticity of handwriting for 50 characters. Hence, the bitmap has 50 rows. A black block indicates a correct answer, and white an incorrect answer. The 25 reproductions are (II-2), (II-3), (II-4), (II-5), (II-6), (II-7), (II-8), (II-9), (II-10), (II-13), (II-14), (II-15), (II-17), (II-18), (II-19), (II-20), (II-21), (II-24), (II-25), (II-26), (II-27), (II-29), (II-33), (II-34), and (II-36) in Figure 7. The 25 characters written by the calligrapher are randomly selected. (b) The average accuracy achieved by each participant in answering the handwriting authentication questions. (c) The average accuracy achieved by the user group. Here, the lower the human authentication accuracy, the more faithfully our algorithm has reproduced the corresponding character. (d) The distribution of the participants according to their accuracy. The x-axis shows the accuracy, and the y-axis indicates the percentage of participants who achieved it.