



Title	Computationally Evaluating and Reproducing the Beauty of Chinese Calligraphy
Author(s)	Xu, S; Jiang, H; Lau, FCM; Pan, Y
Citation	IEEE Intelligent Systems, 2012, v. 27, p. 63-72
Issued Date	2012
URL	http://hdl.handle.net/10722/165874
Rights	Creative Commons: Attribution 3.0 Hong Kong License

Computationally Evaluating and Reproducing the Beauty of Chinese Calligraphy

Songhua Xu, *Oak Ridge National Laboratory*

Hao Jiang and Francis C.M. Lau, *University of Hong Kong*

Yunhe Pan, *Zhejiang University*

Training a computer to evaluate the aesthetics of Chinese characters provides a feedback mechanism to improve the quality of automatically generated calligraphy.

The field of computer vision has predominantly been concerned with recognizing shapes and meanings of objects by their images—that is, figuring out what things are. But in everyday life, our visual perception also induces a sense of how beautiful things are. To the best of our knowledge, attempts

to program a computer to appreciate beauty have been rare. Alan Turing once said, “We do not wish to penalize the machine for its inability to shine in beauty competitions.”¹ But witnessing the rapid advancement of photorealistic techniques in computer graphics nowadays, we feel it is not unimaginable that computers can also distinguish the beautiful, and when that happens, beauty contests could certainly be open to computers as well.

To imbue computers with the ability to recognize beauty is certainly a worthwhile problem for AI researchers. The ultimate intelligent machine is probably one that can create beautiful results on its own, which presents a nice AI research challenge. This article describes our attempt to meet this challenge, the result of which is a system

that can perform beauty appreciation quantitatively over Chinese calligraphy pieces.

Calligraphy and character fonts are closely related. Much research has been done on computerizing Chinese fonts—for example, Ariel Shamir and Ari Rappoport’s study on how to compress Chinese outline fonts.² Computerizing Chinese calligraphy, which is usually done with a brush, is more challenging because the shapes of brush strokes as well as the topology over multiple strokes can be highly complex. A single character alone presents many algorithmic challenges in terms of the shape and spatial layout of the strokes.

In this article, we first explore how to use a numerical method to evaluate the visual quality of calligraphic writings from an aesthetic point of view, and then describe the generation of

Related Work in Character Generation and Aesthetic Evaluation

The most closely related work to that outlined in the main article is the automatic artistic Chinese calligraphy generation system described by some of the authors and their colleagues.¹ That work, however, is concerned mainly with using constraint-based reasoning to generate stylistic calligraphic characters and paid very little attention to the aesthetic quality of the generated results. Chin-Chuan Han and his colleagues proposed an interactive system for grading Chinese calligraphy for writing instruction.² They employed image-processing techniques to extract features related to character position, size, and stroke projections. On the basis of these features, they used hand-coded rules to grade the visual quality of calligraphic writing through fuzzy inference. In contrast, we use a machine-learning approach to automatically grade the visual appearance of calligraphic writing. The grading results of our method closely resemble human aesthetics opinions.

Pak-Keung Lai's team studied the problem of numerically evaluating the beauty of calligraphic characters through a simple heuristics approach.³ They identified four rules in Chinese calligraphy, and then used them to implement a rule-based beauty-grading function. Our automatic calligraphy visual-quality evaluation, on the other hand, is based on a supervised-learning approach. It is generally known that high-level rules for capturing expert knowledge are not always effective and sometimes impossible to derive. Our data-driven approach can lead to a machine evaluation capability better than their rule-based approach in capturing the opinions of human viewers.

Recently, in computer graphics research, a data-driven approach has been proposed for evaluating the attractiveness of human faces.⁴ Success with that approach adds to our confidence in a data-driven approach to evaluating aesthetics of Chinese calligraphic writings. Also related is recent work by some of the current authors and their colleagues that studies the problem of automatically generating Chinese calligraphic writings with style imitation.⁵

Pamela McCorduck's book provides a comprehensive treatment of AI and art,⁶ which motivates our study of the beauty of Chinese calligraphic art from an AI perspective. Other than the visual arts, computer-generated music is probably the most established area for AI techniques being employed in art composition. Current research on computer music includes both automatic music composition and music evaluation. The challenges in that field, however, are in some respects easier than ours, because there are well-established musical rules for judging music, which isn't the case for the visual arts, including calligraphy. Others have pursued work on story generation, believable agents, interactive story, and the like, which aim to capture aesthetics computationally.

Finally, our work is also remotely related to Donald Knuth's pioneering work on Metafont.⁷ That work focuses on the definition and interpretation of fonts, but leaves font creation to the end users. In contrast, our work emphasizes the generation of good-looking character writings.

References

1. S. Xu et al., "Automatic Generation of Artistic Chinese Calligraphy," *IEEE Intelligent Systems*, vol. 20, no. 3, 2005, pp. 32–39.
2. C.-C. Han, C.-H. Chou, and C.-S. Wu, "An Interactive Grading and Learning System for Chinese Calligraphy," *Machine Vision and Applications*, vol. 19, no. 1, 2008, pp. 43–55.
3. P.-K. Lai, D.-Y. Yeung, and M.-C. Pong, "Chinese Glyph Generation Using Character Composition and Beauty Evaluation Metrics," *Proc. Int'l. Conf. Computer Processing Oriental Languages, Chinese and Oriental Languages Society (COLCS)*, 1995, pp. 92–99.
4. T. Leyvand et al., "Data-Driven Enhancement of Facial Attractiveness," *Proc. Siggraph 2008, ACM*, 2008; doi:10.1145/1399504.1360637.
5. S. Xu et al., "Automatic Generation of Chinese Calligraphic Writings with Style Imitation," *IEEE Intelligent Systems*, vol. 24, no. 2, 2009, pp. 44–53.
6. P. McCorduck, *Aaron's Code: Meta-Art, Artificial Intelligence and the Work of Harold Cohen*, W.H. Freeman & Co., 1990.
7. D.E. Knuth, *The Metafont Book*, Addison-Wesley, 1986.

good-looking Chinese calligraphic writings. Our work focuses on the aesthetics evaluation and automatic generation of single calligraphic characters, which are the basic building blocks of any Chinese calligraphic writing. (See the "Related Work" sidebar for other work done in this area.) Our experimental results show that our effort at the character level can effectively improve the visual quality of automatically generated multi-character Chinese calligraphy.

Calligraphy Representation and Acquisition

We adopt the hierarchical Chinese calligraphy representation scheme we

have described before,³ which represents the shape of a character at multiple levels. The method is fully automatic, which unfortunately would fail when processing characters in cursive styles. We therefore set out to devise a more generally applicable method based on a two-phase, semiautomatic routine.

In the first phase, we combine several decomposition algorithms to perform a best-effort automatic stroke extraction (see Figure 1). Starting from a given image of a calligraphic character, we tentatively extract its skeleton using a thinning algorithm.⁴ Next, we employ an adapted version

of Jairo Rocha and Theo Pavlidis's algorithm to automatically segment the strokes.⁵ For characters written in regular styles that don't deviate drastically from standard writing styles, this automatic stroke segmentation works effectively and efficiently, typically taking less than three seconds.

For highly cursive writing styles, however, this approach tends to fail. It might only be able to identify a few strokes because of the severe deviation of a cursive character shape from its standard appearance. In that case, we turn to a stroke library for stroke identification through stroke shape matching,⁶ which generally can

extract strokes more completely. We could not devise a fully automatic routine for stroke extraction, however, because artistic calligraphy tends to be highly cursive and severely distorted, which is beyond the capabilities of existing pattern recognition techniques. Hence, we provide an intelligent user component interface to let users correct or refine the automatic stroke decomposition results. To use the interface, the user only has to sketch the stroke trajectories, which don't need to be highly accurate. Given these user sketches, we adopt a heuristic-based search procedure to look for an optimal stroke match between the user-sketched strokes and the extracted stroke skeleton. This process subsequently leads to all the strokes being successfully decomposed and extracted.

Calligraphy Aesthetics Evaluation

Our calligraphic character aesthetics evaluation is supervised-learning based. We first collected several calligraphic character writing samples from both calligraphy copybooks and student practice books, and they formed the training set for our learning algorithm. Having established the sample collection, we invited five calligraphists to rate the visual quality of each sample character. For simplicity, we asked them to use only three labels—"good," "so-so" and "bad." For each labeled character, we computed three values (x , y , and z) using a Bayesian estimation method. Each value indicated the probability that a character's appearance would be judged good, so-so, or bad on a 0 to 100 percent scale. We can also convert this probabilistic aesthetics evaluation result into a single numerical score:

$$\begin{aligned} \text{Overall Score} &\triangleq x + 0.5y + 0z \\ &\triangleq x + 0.5y. \end{aligned} \quad (1)$$

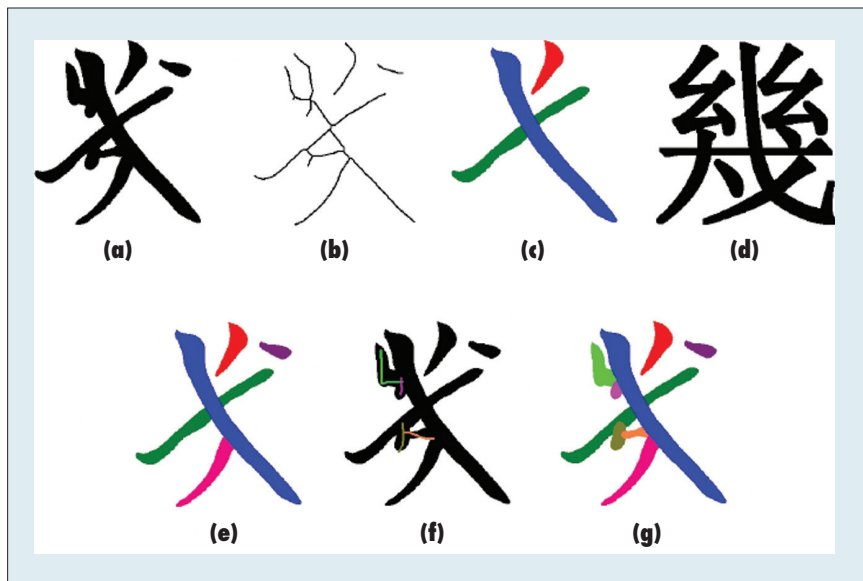


Figure 1. Stroke extraction. Given (a) an input calligraphic character, (b) we first extract its skeleton. In this case, (c) automatic stroke segmentation extracts only three strokes, because of (d) the severe deviation of the input character from its standard style. (e) Matching against a stroke library extracts two more strokes. (f) After a user sketches a few suggestive stroke trajectories, (g) our intelligent user interface extracts all the strokes.

Thus, the value domain of the converted numerical score is between 0 and 100 percent because $x + y + z = 100$ percent.

On the basis of our discussions with some practicing calligraphists, we took into account three aspects when designing our calligraphy aesthetics evaluation algorithm: the shapes of individual strokes, the topological relationship between these strokes, and the style consistency among the strokes.

Evaluating Individual Stroke Shape

Learners of calligraphy must start with writing decent-looking single strokes. The reason is simple: a single ugly stroke can destroy the beauty of the entire character. Accordingly, when grading the aesthetics of a character, we first estimate a visual appearance score for each of its constituent strokes.

In the covering-ellipse stroke representation,³ each stroke contains a series of points on its skeleton, and each point has a corresponding covering ellipse. We denote a stroke's skeleton

as K , which is a discrete 2D curve comprising all the pixels on the stroke skeleton. Without loss of generality, we assume the stroke has n pixels—that is, $K(1)$ is the first pixel on the stroke skeleton, and $K(n)$ is the last pixel of the skeleton. Let Kx and Ky consist of all the x and y coordinates of $K(1), K(2), \dots, K(n)$ in that order; in other words, $Kx = \{K(1).x, \dots, K(n).x\}$ and $Ky = \{K(1).y, \dots, K(n).y\}$. Kx and Ky are two discrete 1D curves.

We also denote the covering ellipse corresponding to $K(i)$ as $E(i)$. Then the lengths of the major radii of $E(1), \dots, E(n)$ form a 1D curve Ma . Similarly, the lengths of the minor radii of $E(1), \dots, E(n)$ form a 1D curve Mi . Last, for each skeleton pixel $K(i)$, we also compute a minimum distance, $D(i)$, from the pixel to a point on the stroke contour (see Figure 2). All the $D(i)$ values together compose an offset distance sequence $D = \{D(1), \dots, D(n)\}$, which is another 1D discrete curve.

Now we have a set of 1D curves, $\omega = (Kx, Ky, Ma, Mi, D)$. We then compute the associated gradient

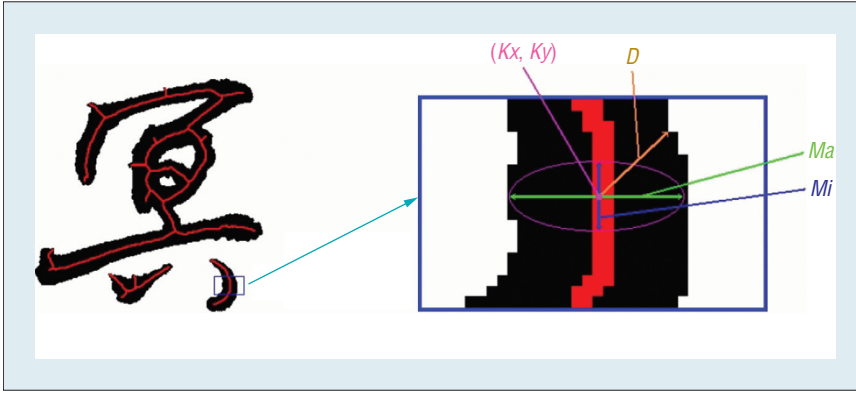


Figure 2. Deriving features of a single stroke. (Kx, Ky) are the coordinates of all the pixels on the stroke's skeleton K . Ma and Mi are curves formed by the lengths of the major and minor radii of the covering ellipses. D is the distance from a skeleton pixel to the edge of the stroke contour.

curves for each of them, getting another set of 1D curves, $\omega' = (Kx', Ky', Ma', Mi', D')$. With both ω and ω' , we can compute the shape features of the curves for use in both the learning and the grading processes. For each curve C , which is a 1D signal, we obtain its largest element (C_{\max}), the average value (C_{ave}), and its median value (C_{med}). The set of features extracted is derived as

$$F \triangleq \{C_{\max} | C \in \Theta\} \cup \{C_{\text{ave}} | C \in \Theta\} \cup \{C_{\text{med}} | C \in \Theta\} \cup \theta \quad (2)$$

where $\Theta \triangleq \omega \cup \omega'$, and θ is defined by

$$\theta \triangleq \theta_{\text{ave}} \cup \theta_{\max} \cup \theta_{\text{med}}$$

$$\theta_{\text{ave}} \triangleq \left\{ \frac{C'_{\max}}{C_{\text{ave}}}, \frac{C'_{\text{ave}}}{C_{\text{ave}}}, \frac{C'_{\text{med}}}{C_{\text{ave}}} \mid C \in \omega \right\}$$

$$\theta_{\max} \triangleq \left\{ \frac{C'_{\max}}{C_{\max}}, \frac{C'_{\text{ave}}}{C_{\max}}, \frac{C'_{\text{med}}}{C_{\max}} \mid C \in \omega \right\}$$

$$\theta_{\text{med}} \triangleq \left\{ \frac{C'_{\max}}{C_{\text{med}}}, \frac{C'_{\text{ave}}}{C_{\text{med}}}, \frac{C'_{\text{med}}}{C_{\text{med}}} \mid C \in \omega \right\}$$

In our experiments, those features in θ_{ave} make the most difference in our calligraphic curve aesthetics learning and grading processes.

To provide labels for the training examples, we ask the calligraphists to grade each stroke (see Figure 3). After computing the probabilistic grades from these manual labels, we

feed a training set containing a total of 2,500 labeled single-stroke examples collected from about 500 single characters into a four-layered back-propagation neural network and trained it iteratively. The input to our neural network is the feature set F derived in Equation 2. The output is a probabilistic evaluation result in the form of (x, y, z) , where x, y , and z are probabilistic values, each in the range [0%, 100%], which indicate whether the stroke is good, so-so, or bad, respectively. We can also synthesize a scalar score on the basis of these three probabilistic values using Equation 1. During the neural-network-training process, we use ten-folded cross-validation to avoid overfitting. Table 1 shows a comparison of the human grading results with those from our algorithm.

Evaluating Stroke Spatial Layout

As important as the appearance of single strokes is the way the strokes are arranged to compose a character. The visual qualities of the individual strokes interact to form the overall visual impression of the whole character. For Chinese characters, these spatial arrangements not only affect the aesthetic appearance but also can lead to different interpretations as to what the characters are. Sometimes a small change of the spatial

relationship between strokes can result in an entirely different character, not merely the same character written in a different style. This poses a difficult challenge to our algorithmic design.

Assume α is a Chinese character whose stroke layout is to be graded. For every pair of its strokes (a, b) , we compute the maximum, minimum, and mean distances— $l_{\max}(a, b)$, $l_{\min}(a, b)$, and $l_{\text{mean}}(a, b)$ —from a point on one stroke to a point on the other. These values can describe both the topological and the spatial relationship between the strokes. For example, we can easily determine whether the two strokes intersect and how much they overlap, if at all. However, these three values might not tell us the strokes' relative position precisely, which is important for the whole visual appearance or for determining the character's identity. To capture that, inspired by our previous work,³ we draw a bounding box for each stroke (a bounding box of a stroke is the minimum rectangle that includes all the interior area of that stroke), and then compute the horizontal, vertical, and planar overlap between two strokes' bounding boxes (see Figure 4). We denote these three types of overlap as $B_h(a, b)$, $B_v(a, b)$, and $B_p(a, b)$ respectively, which are computed by

$$\begin{cases} B_h(a, b) \triangleq (X(a) - X(b)) / (\text{Width}(a) + \text{Width}(b)) \\ B_v(a, b) \triangleq (Y(a) - Y(b)) / (\text{Height}(a) + \text{Height}(b)), \\ B_p(a, b) \triangleq (I(a) \cap I(b)) / (I(a) \cup I(b)) \end{cases}$$

where $\text{Width}(s)$ and $\text{Height}(s)$ denote the width and height of the bounding box for stroke s , $X(s)$ and $Y(s)$ are the horizontal and vertical coordinates of the bounding box's center, and $I(s)$ denotes the interior area of stroke s . Assuming the character α has n strokes, doing the above gives us six $n \times n$ matrices $M_{\max}(\alpha)$, $M_{\min}(\alpha)$, $M_{\text{mean}}(\alpha)$, $M_h(\alpha)$, $M_v(\alpha)$, and $M_p(\alpha)$.

The element in row i and column j of each matrix is $I_{\max}(S_i, S_j)$, $I_{\min}(S_i, S_j)$, $I_{\text{mean}}(S_i, S_j)$, $B_b(S_i, S_j)$, $B_v(S_i, S_j)$, and $B_p(S_i, S_j)$, respectively, where S_i and S_j are strokes i and j of α .

The next step is to derive the features of these matrices. To grade the spatial layout of α , we first find its corresponding standard character writing style $\tilde{\alpha}$. Here we use the Kai style (GB2312 in China's national font standard) as the standard Chinese character writing style. We then derive the previous six matrices for $\tilde{\alpha}$, resulting in $M_{\max}(\tilde{\alpha})$, $M_{\min}(\tilde{\alpha})$, $M_{\text{mean}}(\tilde{\alpha})$, $M_b(\tilde{\alpha})$, $M_v(\tilde{\alpha})$, and $M_p(\tilde{\alpha})$. For each of these six feature matrices on $\tilde{\alpha}$, denoted as $M(\tilde{\alpha})$, and its corresponding feature matrix $M(\alpha)$ on the standard writing style $\tilde{\alpha}$, we derive their difference matrix $M^{\text{dif}}(\alpha) = M(\alpha) - M(\tilde{\alpha})$. Once $M^{\text{dif}}(\alpha)$ is calculated, we compute its maximum element value φ_{\max} , minimum element value φ_{\min} , maximum absolute value $\varphi_{\max a}$, mean element value φ_{mean} , median element value φ_{med} , and first three eigenvalues (λ_1 , λ_2 , and λ_3). Thus, we derive a total of $6 \times 8 = 48$ features for character α .

Again, we use a four-layered back-propagation neural network in our grading process (see Figure 5). By working with the layout difference between a particular calligraphic writing style and its corresponding standard writing style, we don't have to bother with much of the layout details associated with a particular character composition structure. That is, the features we extract are more indicative of the writing style of an individual writer, and become largely unaffected by a character's own particular stroke composition topology. This practice is inspired by our prior work.³

We feed our grading neural network more than 500 character samples for the 100 most frequently used characters, all of which come from

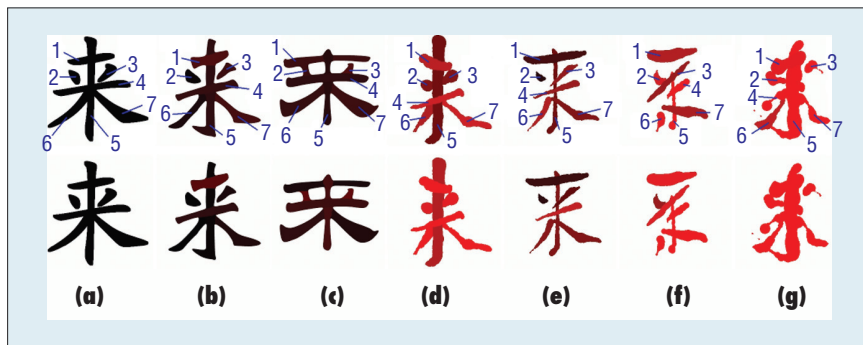


Figure 3. Grading of strokes by the calligraphists. The top row shows how our algorithm evaluated the strokes in these characters, and the bottom row shows the human evaluation. A black stroke is one that looks "good," and a red one is "bad." In general, our algorithm's grading results agree well with those made by human calligraphists. Table 1 shows the overall scores for each stroke.

Table 1. Comparison of algorithmic and human stroke grading of the strokes in Figure 3 (%).*

Character	Stroke no.						
	1	2	3	4	5	6	7
(a)	100	100	100	100	100	100	100
	100	100	100	100	100	100	100
(b)	67.1	95.4	74.2	77.6	80.9	88.7	71.9
	65	100	90	80	95	100	85
(c)	74.2	73.6	68.7	79.8	81.8	76.2	74.7
	90	60	60	75	70	80	85
(d)	26.0	49.5	40.9	21.7	54.1	31.1	20.4
	5	15	10	0	30	5	0
(e)	71.5	82.7	41.0	40.4	55.2	31.1	48.0
	75	80	60	25	50	45	50
(f)	28.8	16.8	38.1	9.5	7.0	12.5	34.4
	10	40	15	0	0	5	5
(g)	6.6	11.7	0	1.4	0	22.9	4.3
	0	0	0	0	0	5	0

*For each character, the algorithmic grade is in the top row and the human expert grade on the bottom.

different people with different experience levels in calligraphy; the samples also include a few ugly or naively written ones. As we did with individual stroke grading, we assign only the labels good, so-so, and bad. Thus the input dimension to our neural network is 48, and its output dimension is 3. We also generate an overall score using Equation 1. We train the neural network using 10,000 iterations, and we use ten-folded cross-validation to avoid overfitting during the training process.

Figure 6 shows the results for spatial-layout grading on some characters in the testing set whose human-applied labels were unknown to our grading neural network. Table 2 compares the grades from our algorithm with those from human calligraphists for 10 other characters in the test. We set the threshold for visual acceptance at a score of 70 percent. This might sound like a high bar, but our main goal here is to ensure no poor-looking Chinese calligraphy generation results will be output as the end result.

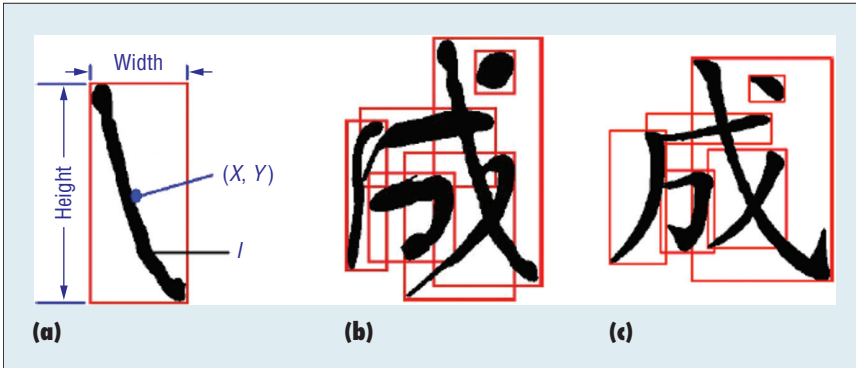


Figure 4. Stroke bounding boxes. (a) An individual stroke's bounding box and its key features, where (X, Y) are the coordinates of the box's center, and l is the interior area of the stroke; (b) all the bounding boxes of a single character; and (c) the bounding boxes of the same character written in standard style.

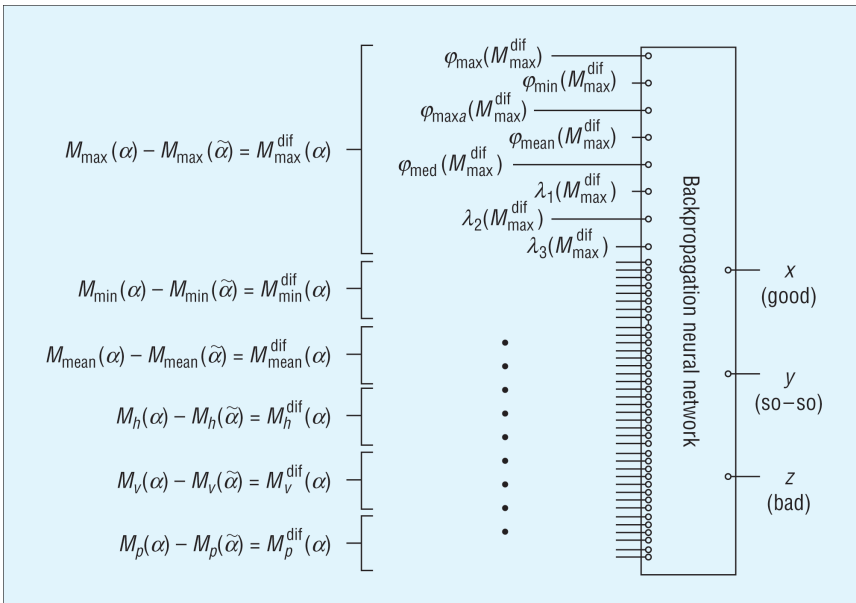


Figure 5. Our grading system. The equations on the left involve the six feature matrices for the character being graded (α) and the corresponding matrices for the same character in a standard font $(\tilde{\alpha})$. The element values and eigenvalues for the top result are shown, although we calculate them for the other results as well.

Because there is a huge generation candidate space, as we explain later, using a strict threshold with some false negatives is more desirable than using an overly relaxed threshold that might lead to false positives.

Evaluating Consistency of Stroke Styles

An ugly-looking character can consist of good-looking strokes, depending on whether the styles of different strokes cooperate harmoniously. Normally, strokes in the same style would lead

to a good-looking character in that style if the individual strokes and their spatial layout are both good. In some cases where the styles of strokes are different but consistent, the strokes could still form a good-looking character. We thus need to evaluate the style consistency among strokes.

For simplicity, we assume there are m well-recognized writing styles commonly found in copybooks authored by professional calligraphists. For each stroke in a character to be graded, we create an m -dimensional

vector measuring the probability for the stroke to be in each sample writing style—for example,

$$G(S) = \{g_i(S) | i = 1, \dots, m\},$$

defines a set $G(S)$ in which $g_i(S)$ is the probability that stroke S is written in the i th writing style. In our experiments, we fix the number of sample writing styles at six ($m = 6$) by selecting the six most frequently used styles in the Chinese font library. For a given stroke S , to determine its corresponding $g_1(S), \dots, g_6(S)$, we utilize the “cost of matching measurement” proposed by Rocha and Pavlidis, which defines the cost of transforming one stroke shape to match another. Assume the corresponding stroke shapes of S in the six most frequently used Chinese font styles are S_1, \dots, S_6 . We denote the cost of matching measurement between S and S_i as $CM(S, S_i)$. Then we compute $g_i(S)$ as

$$g_i(S) = \frac{CM^{-1}(S, S_i)}{\sum_{j=1}^6 CM^{-1}(S, S_j)}.$$

For all the strokes contained in a character, S^1, S^2, \dots, S^n , we deduce their corresponding style signatures, $g(S^1), g(S^2), \dots, g(S^n)$, as we just described. We then compute the mean signature $\bar{G} = \{\bar{g}_1, \dots, \bar{g}_6\}$ by averaging all the corresponding components in each stroke's style signature. Now we can extract the style inconsistency SI for each stroke. For stroke S^1 whose style signature is $\{g_1(S^1), g_2(S^1), \dots, g_6(S^1)\}$, we derive $SI(S^1)$ as

$$SI(S^1) \triangleq \sum_{j=1}^6 |g_j(S^1) - \bar{g}_j|.$$

We also use a backpropagation neural-network approach to grade a character's style consistency. The input of our neural network is a 3D

vector that includes the three largest style inconsistency values of the character's constituent strokes. The network's output is three real numbers between 0 and 1, denoting the probabilities (as before) that the consistency between strokes in the character is good, so-so, or bad.

To collect the training set, we develop a program that randomly extracts good-looking strokes from sample calligraphic writings and assembles them into characters using the original sample character's stroke composition layout. We then ask our

calligraphists to rate each training sample with one of the three labels. Next, we feed these labeled, random samples into the neural network for training using 10,000 iterations. We also use ten-folded cross-validation to avoid overfitting during the process. After deriving the probabilities of the character's style consistency in each of the three categories, we use Equation 1 to synthesize an overall style consistency score. Table 3 shows some evaluation results of the stroke style consistency grading experiments.

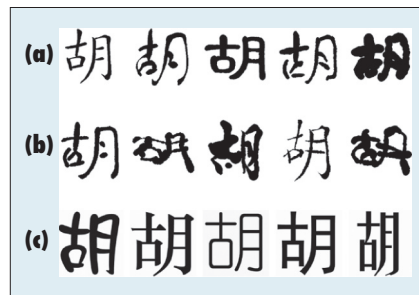


Figure 6. Sample characters. (a) From left, three positive and two negative examples from our training set; (b) characters rated unacceptable by spatial relationship analysis; (c) aesthetically acceptable results (scores > 70 percent) according to our grading algorithm.

Table 2. Comparison of algorithmic and human spatial-layout grading.

Character	Labels and overall score (%)*				Character	Labels and overall score (%)*			
	Good	So-so	Bad	Overall		Good	So-so	Bad	Overall
怒	100	0	0	100	怒	55.0	45.0	0	77.5
	100	0	0	100		90	10	0	95
怒	61.6	38.4	0	80.8	怒	0	54.3	45.7	77.2
	70	30	0	85		50	50	0	75
怒	86.8	13.2	0	93.4	怒	0	54.3	45.7	77.2
	100	0	0	100		20	60	20	50
怒	35.5	60.0	4.5	65.5	怒	0	50.7	49.3	75.4
	40	60	0	70		20	60	20	50
怒	31.3	60.0	8.7	61.3	怒	98.4	1.6	0	99.2
	70	30	0	85		100	0	0	100

*For each character, the algorithmic grade is in the top row and the human expert grade is on the bottom.

Table 3. Comparison of algorithmic and human style consistency grading.

Character	Labels and overall score (%)*				Character	Labels and overall score (%)*			
	Good	So-so	Bad	Overall		Good	So-so	Bad	Overall
江	100	0	0	100	江	0	0	100	0
	100	0	0	100		0	20	80	10
江	3.9	61.2	34.9	34.5	江	0	9.3	90.7	4.7
	20	60	20	50		0	40	60	20
江	91.0	9.0	0	95.5	江	0	12.0	88.0	6.0
	70	30	0	85		0	50	50	25
江	62.7	37.3	0	81.4	江	58.6	41.4	0	79.3
	100	0	0	100		40	60	0	50

*For each character, the algorithmic grade is in the top row and the human expert grade is on the bottom.



Figure 7. Example characters used for synthesizing an overall character score. Table 4 shows the scores each character received. The scores produced by our algorithm on the aesthetics of the whole character agree closely with the human scores. This shows that our algorithm can grade the visual quality of calligraphic writings satisfactorily.

Table 4. Comparison of algorithmic and human overall character scores for the strokes in Figure 7 (%).*

Character	Spatial layout				Stroke consistency				Whole character				Human scores			
	Good	So-so	Bad	Overall	Good	So-so	Bad	Overall	Good	So-so	Bad	Overall	Good	So-so	Bad	Overall
(a)	83.5	16.5	0	91.8	93.5	6.5	0	96.8	67.8	32.2	0	83.9	70	30	0	85
(b)	75.7	24.2	0	87.8	62.1	37.9	0	81.1	71.1	28.9	0	85.6	80	20	0	90
(c)	0	10.7	89.3	5.4	26.7	55.5	17.8	54.5	0	16.4	83.6	8.2	0	0	100	0
(d)	100	0	0	100	100	0	0	100	100	0	0	100	100	0	0	100
(e)	66.2	33.8	0	83.1	87.0	13.0	0	93.5	34.7	58.9	6.4	64.2	20	80	0	60
(f)	9.6	60.2	30.2	39.7	14.9	46.2	38.9	38.0	14.2	57.5	28.3	43.0	20	60	20	50
(g)	0	26.5	73.5	13.3	28.6	66.4	5.0	61.8	0	57.0	43.0	28.5	0	10	90	5
(h)	100	0	0	100	100	0	0	100	100	0	0	100	100	0	0	100

*The scores produced by our algorithm on the aesthetics of the whole character agree closely with the human scores, both of which are displayed in bold.

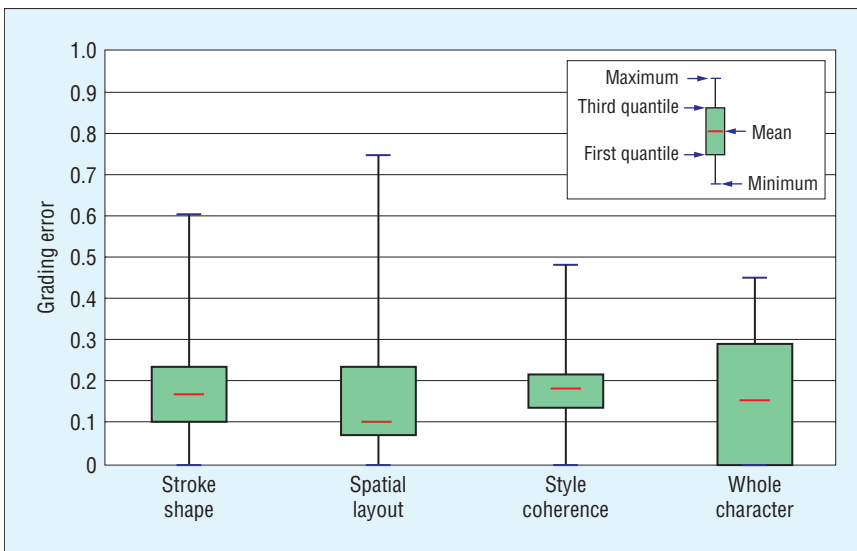


Figure 8. Statistical distributions of the errors of our algorithm’s grading results. The box-and-whisker diagram illustrates the variance between our algorithm’s grading results and the scores given by human calligraphists.

Evaluating a Whole Character

Finally, we evaluate the aesthetics of a whole character through a decision

tree that produces the composite scores of characters having a certain number of strokes. The decision tree’s

input includes the character’s topological score, its style consistency score, and the single-stroke appearance scores for all the strokes in the character. For a character with n strokes, the input to the decision tree has $3n + 6$ dimensions: the n strokes produce $3n$ single-stroke appearance scores because each stroke’s aesthetics grade is associated with three probabilistic values, and similarly, the character’s topological score and style consistency score each produce three dimensions.

The examples used for training each of these decision trees are 50 characters; each character is written in six different styles and labeled by a human calligraphist. Figure 7 shows a selection of the characters used as examples, and Table 4 compares the human labeling results with the aesthetics scores produced by our algorithm.

Figure 8 reports our evaluation of the accuracy of our calligraphy aesthetics grading on a set of 200 characters

THE AUTHORS

previously unseen by our algorithm. The mean grading error is less than 0.2 in the range of 0 to 1 for the whole-character grading results. Considering the fact that human calligraphists sometimes do not even agree with one another exactly on the aesthetics of a piece of calligraphic writing, such performance from our algorithm is quite satisfactory.

Automatic Generation of Aesthetic Calligraphy with Visual-Quality Feedback

We previously introduced a system that can generate a variety of stylistic calligraphic characters following an analogous-reasoning approach.³ However, only a subset of the generated results is truly aesthetically pleasing, owing to the lack of a powerful built-in judging mechanism. Using our proposed calligraphy grading method, we can add an elaborate and practical quality control module to that system.

We have integrated these algorithms into our experimental system. We use our grading algorithm to evaluate each font produced by the prior generation system. According to the overall visual-quality score, the generation parameters are varied from their original settings to yield a new font with a higher aesthetics score. The process repeats just like the case of a typical optimization problem. Assuming there are n parameters involved in the algorithmic process to generate a character C , a scalar function $f(C)$ with n input variables can be formulated whose output value is the overall aesthetics score of the whole character. The task of calligraphy beautification is thus reduced to finding a point in the n -dimensional input variable space that can maximize the function value. We employ a gradient-descent method that iteratively optimizes

Songhua Xu is a research staff member and a Wigner Fellow at Oak Ridge National Laboratory. His research interests include AI, computer graphics, human-computer interaction, information retrieval, and biomedical informatics. Xu has a PhD in computer science from Yale University. Contact him at xus1@ornl.gov.

Hao Jiang is a postgraduate student in the Department of Computer Science at the University of Hong Kong. His research interests include AI, human-computer interaction, computer graphics, the World Wide Web, and robotics. Jiang has an MEng in computer science and technology from Zhejiang University. Contact him at hjiang@cs.hku.hk.

Francis C.M. Lau is a professor in the Department of Computer Science at the University of Hong Kong. His research interests include parallel and distributed computing, wireless networks, and AI. Lau has a 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, and an honorary professor at the University of Hong Kong. His research interests include AI, computer graphics, intelligent CAD, and computer art. Pan has an MEng in computer science and engineering from Zhejiang University. Contact him at panyh@zju.edu.cn.

the target function to search for the best possible quality improvement for the initial calligraphic writing. Figure 9 shows some examples of how our grading method can incrementally improve a calligraphic character.

Integrating our calligraphy aesthetics evaluation algorithm into the automatic calligraphy generation system results in a significant improvement in the quality of the calligraphy that the computer generates and outputs. Thus, our system is an “aesthetic calligraphy generation system.” Figure 10 shows a verse of poetry written in automatically generated calligraphy characters, both with and without our appearance grading method being employed as a feedback component in the generation process.

Our work attempts to evaluate the aesthetics of Chinese calligraphic characters through an integrated intelligence approach. Our approach is learning based, and it sheds light on the possibility of numerical methods to assess beauty. We have obtained encouraging results from the experiments on the quantitative

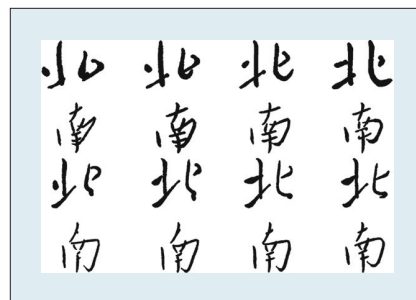


Figure 9. Examples of improvements from our grading method. Each row shows four versions of the same character. The aesthetics scores increase from left to right, corresponding to incremental improvements in their automatic generation parameters based on the results of our grading algorithm.

accuracy of our algorithms. In the future, we plan to investigate other alternative machine-learning approaches and feature designs to explore the optimal algorithm design for evaluating calligraphy aesthetics. ■

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC), grant 60903132. Songhua Xu partly performed this research as a Eugene P. Wigner Fellow and staff member at the Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the US Department of Energy under contract



Figure 10. Twenty versions of an ancient poetic verse from the Tang Dynasty in the Eighth Century. The first five columns are learning samples from human calligraphy. The center section shows 10 automatic calligraphy versions using our visual-quality grading algorithm for the feedback. The last five versions are automatic-calligraphy but without our grading component as feedback. Chinese scholars participating in the experiment judged the visual quality of the center 10 examples as clearly and significantly better than those on the right.


DE-AC05-00OR22725. This manuscript has been authored by UT-Battelle, LLC. The publisher, by accepting the article for publication, acknowledges that the US Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US Government purposes.

References

1. A.M. Turing, "Computing Machinery and Intelligence," *Mind*, vol. 59, 1950, pp. 433-460.

2. A. Shamir and A. Rappoport, "Compacting Oriental Fonts by Optimizing Parametric Elements," *The Visual Computer*, vol. 15, no. 6, 1999, pp. 302-318.
 3. S. Xu et al., "Automatic Generation of Artistic Chinese Calligraphy," *IEEE Intelligent Systems*, vol. 20, no. 3, 2005, pp. 32-39.
 4. C. Neusius and J. Olszewski, "A Non-iterative Thinning Algorithm," *ACM Trans. Mathematical Software*, vol. 20, no. 1, 1994, pp. 5-20.
 5. J. Rocha and T. Pavlidis, "A Shape Analysis Model with Applications to a

Character Recognition System," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 4, 1994, pp. 393-404.
 6. M. van Eede et al., "Canonical Skeletons for Shape Matching," *Proc. Int'l Conf. Pattern Recognition*, 2006, pp. 64-69.

 Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.