An Intelligent System for Chinese Calligraphy

Songhua Xu[†,‡,*], **Hao Jiang**[†,§], **Francis C.M. Lau**[§] **and Yunhe Pan**[†,¶]

† CAD & CG State Key Lab of China, Zhejiang University, Hangzhou, Zhejiang, P.R. China, 310027
‡ Department of Computer Science, Yale University, New Haven, Connecticut, USA, 06511
§ Department of Computer Science, The University of Hong Kong, Hong Kong, P.R. China
¶ Chinese Academy of Engineering, Beijing, P.R. China, 100088

Abstract

Our work links Chinese calligraphy to computer science through an integrated intelligence approach. We first extract strokes of existent calligraphy using a semi-automatic, two-phase mechanism: the first phase tries to do the best possible extraction using a combination of algorithmic techniques; the second phase presents an intelligent user interface to allow the user to provide input to the extraction process for the difficult cases such as those in highly random, cursive, or distorted styles. Having derived a parametric representation of calligraphy, we employ a supervised learning based method to explore the space of visually pleasing calligraphy. A numeric grading method for judging the beauty of calligraphy is then applied to the space. We integrate such a grading unit into an existent constraint-based reasoning system for calligraphy generation, which results in a significant enhancement in terms of visual quality in the automatically generated calligraphic characters. Finally, we construct an intelligent calligraphy tutoring system making use of the above. This work represents our first step towards understanding the human process of appreciating beauty through modeling the process with an integration of available AI techniques. More results and supplementary materials are provided at **http://www.cs.hku.hk/∼songhua/calligraphy**.

Introduction and Related Work

The field of computer vision is predominantly concerned with recognizing shapes and meanings of objects by their images. But there is more than just knowing *what* things are. In everyday life, our visual perception also leads to a sense of *how beautiful* things are. Recently, there is an increased interest in affective computing (Picard 2000).To the best of our knowledge, however, there has not been a solution to beauty appreciation by numerical means. Turing once said in his innovational paper (Turing 1950) that "We do not wish to penalise the machine for its inability to shine in beauty competitions, ..." But witnessing the advancement of photorealistic techniques in computer graphics nowadays, if computers can tell the beautiful from what is not, beauty contests could certainly be open to them as well. Such contests will probably not feature a catwalking computer, but rather computer-generated images of beauty, or these images against real, human-generated ones. This might be going overboard, but to imbue the computer with the ability to recognize beauty will likely win general support. Furthermore, the ultimate intelligent machine in people's mind is one that can create results of beauty on its own, which certainly represents a nice challenge for AI researchers.

We feel entailing the computer in understanding or even producing outputs of beauty that are visible, such as painting, calligraphy and sculpture, is a reasonable first step in the long journey to arrive at the ultimate intelligent machine that can deal with beauty of any form. This paper presents such a "first step" of ours—an approach to the problem of understanding the beauty of Chinese calligraphy and its facsimile by the computer. We picked Chinese calligraphy because of its great importance in Chinese art and history and the many interesting challenges it presents to the computer. Our solution demonstrates the power of integrated intelligence.

The closest related work to this paper is the automatic artistic Chinese calligraphy generation system by Xu et al. (Xu, Lau, & Pan 2003; Xu *et al.* 2004; 2005). Their work, however, is concerned mainly with using constraint based reasoning to generate stylistic calligraphic characters, and paid very little attention on the aesthetic aspects of the generated results, a major issue to be addressed by our paper. Avizzano et al. (Avizzano, Solis, & Bergamasco 2002) proposed a calligraphy tutoring system using a haptic interface. In the field of computer graphics, there has been some work on automatic painting creation which however is mostly done with reference to a given photograph (Hertzmann 2003). Others have explored using a combination of artificial intelligence and interactive techniques to produce painting-style animation (Xu *et al.* 2006). The book by McCorduck (McCorduck 1990) provides a comprehensive treatment on artificial intelligence art. Outside the domain of visual arts, computer music is probably the single most successful research area in which AI techniques have been employed to do or assist

*Contact him at songhua.xu@yale.edu.

music composition (Roads 1985).[1] It is important to notice that existent research on computer music includes both *automatic* music composition and music evaluation, which serves as a good inspiration for our work on calligraphy. Their problems in some respect are easier than ours because there are well established music theories for judging music; that is not the case for visual arts including calligraphy. There is also an abundant collection of work on story generation, believable agents (Bates 1994), interactive story, and the like, which aim at capturing aesthetics computationally. Finally, our work is also remotely related to Knuth's pioneering work on Metafont (Knuth 1986). Knuth's 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 fonts.

The organization of the remainder of this paper is as follows. We start with a representation of Chinese calligraphy in Sec. 2. An intelligent user interface for decomposing aesthetic Chinese calligraphy is described in Sec. 3.We discuss how to "grade" the aesthetic quality of a piece of calligraphy in Sec. 4. We explain how this evaluation method is integrated in a constraint-based calligraphy generation system that can produce beautiful calligraphy in Sec. 5.We present a calligraphy writing tutoring system based on our method in Sec. 6, and show some representative experiment results in Sec. 7.We conclude the paper in Sec. 8.

Representing Chinese Calligraphy

Choosing a suitable form of knowledge representation is key to the success of an intelligent system design. Following the practice in computer vision, having a parametric curve-based representation may provide an efficient and effective support for solving our domain specific problem. Such a representation gives two immediate benefits: 1) The representation is not per-pixel based, and thus is robust in the face of local noises; and it has more descriptive power on the overall shape and tends to better reflect high-level visual features; 2) the automatically generated shapes would look more elegant because a parametric representation naturally can overcome the shape aliasing problem. Inspired by (Xu *et al.* 2005), we adopt a hierarchical representation which iteratively constructs the representation of a character from individual strokes level after level.

Intelligent User Interface for Aesthetic Calligraphy Decomposition

To derive the above representation, we introduce a two-phase, semi-automatic processing routine. In the first phase, we combine several decomposition algorithms which are based on various AI techniques to perform a best-effort automatic stroke extraction. In the second phase, through an intelligent user interface we provide to the user, some remedial user interaction would convert the best-effort result to the desired hierarchical parametric representation. We could not rely on a fully automatic processing routine because aesthetic calligraphy tends to be highly cursive and severely distorted, which renders an automatic decomposition process based on available pattern recognition techniques practically impossible. Our intelligent user interface allows the user to optionally correct or refine the automatic stroke decomposition results, especially the difficult cases. The "intelligence" of this UI component lies in its ability to solve the last bits of the puzzle based on the user's very sketchy hints. We show one example, in Fig. 1, to explain this process step by step.

Best-effort automatic stroke decomposition

Starting from a given image of a calligraphic character, Fig. 1(a), we tentatively extract its skeleton (b) using a thinning algorithm (Neusius & Olszewski 1994). After that, we employ a series of analyzing algorithms (Liu, Huang, & Suen 1999; Shin 2004; Rocha & Pavlidis 1994) to perform automatic stroke segmentation. As shown in (c)–(g), this automatic effort identifies three strokes, but it leaves many segments unclassified due to the severe deviation of the shape of the aesthetic character from the regular one. We then turn to a stroke library for stroke identification by shape comparison (Sundar *et al.* 2003; van Eede *et al.* 2006). This additional shape matching process extracts two more strokes (h). There remain still some unidentified strokes in the sub-graph after all these efforts. At this point, we turn to the intelligent user interface.

Intelligent user interface for the difficult cases

We ask the user to draw just a few suggestive sketches to help the computer. We have developed an intelligent user interface for this purpose. The interface does not expect strokes to be drawn in high accuracy, as can be seen in Fig. 1(i). Inspired by (Liu, Kim, & Kim 2001), given the user's sketches, we adopt a heuristic A* search to look for an optimal stroke match between the user input and the remainder of the sub-stroke graph (j). During this search process, we also assign probability values to stroke attributes and their spacial relationships in a fashion similar to (Liu, Kim, & Kim 2001). With this algorithmic processing in the background, our intelligent user interface gets all the strokes decomposed (k),(l). This figure as well as the additional results in our project website show that our intelligent user interface can successfully help decompose aesthetic calligraphy.

Grading the Calligraphy

In this section, we discuss how to rate the aesthetic quality of calligraphic characters, according to both the shapes of their constituent strokes and the spatial relationship be-

[1] In IJCAI 2007, there is an independent track called MUSIC-AI 2007 devoted to the topic.

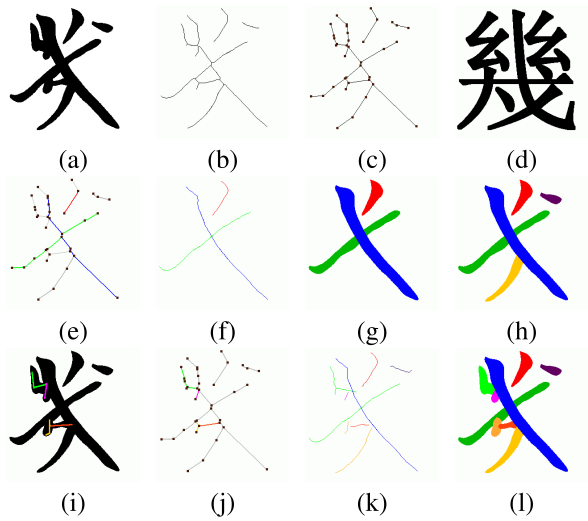


Figure 1: An example of decomposing a calligraphic character by our two-phase decomposition method which utilizes an automatic analysis mechanism and an intelligent user interface. The input calligraphy character (a), its thinning result (b), its sub-stroke graph (c), the shape of the character written in the Kai style (d), the automatic stroke segmentation result by stroke topology analysis and matching, which only extracts three strokes (e). The trajectories of the stroke are smoothed (f), and their shapes extracted (g). After performing the shape matching, we have two more strokes identified (h). The remaining strokes are too difficult to be extracted by any automatic algorithm due to their great deviation from the standard way of writing. We turn to the intelligent interface, with which the user only sketches four simple strokes (i), according to which we can have all the strokes in the sub-stroke graph identified (j); the corresponding stroke trajectories (k), and shapes (l), can thus be all extracted.

tween these strokes. Given an input calligraphic character, our grading algorithm will give a score in the range of [0, 100]. Having the system generate only a single score renders easy reading by humans, and is good enough for our present experiments. The grading functionality is achieved through a learning based approach by studying existent sample calligraphies in multiple styles to numerically explore the appropriate rules for grading the visual quality of an input. We employ the classical back-propagation neural network as the underlying learning device. To train the net, we feed it with samples of both beautiful and ugly characters, of which the beautiful ones come from printed calligraphy books and the ugly ones from novices' writings. Using the decomposition algorithm explained in the previous section, we derived their optimal parametric representations, which are then processed to extract the high-level geometric features for efficient learning.

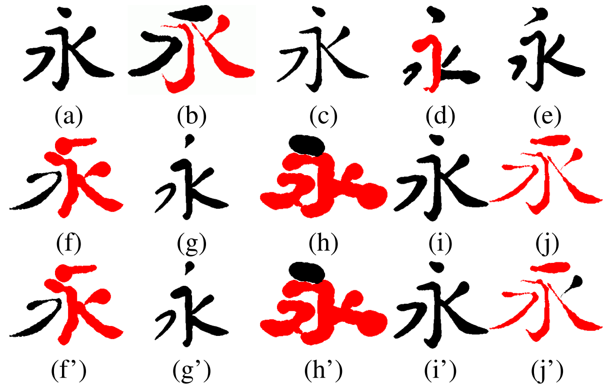


Figure 2: Single stroke grading. (a)–(e) are five example characters among a set of ten sample characters, from which fifty single strokes are extracted to form the training set for single stroke grading. The red ones are aesthetically unacceptable strokes and the black ones are acceptable strokes. (f)–(j) are five new characters unknown to the training set. They are colored according to the grading results produced by our algorithm. (f')–(j') are the corresponding human expert graded results. Our algorithm only makes one mistake for a stroke, in (j), where the stroke is rejected because it is not connected; the human expert feels in this situation the stroke still looks beautiful even though it is against some general rule of calligraphy.

Grading Individual Strokes

In the first step, we grade individual strokes. This agrees with the general learning process where one of the basic learning objectives for calligraphy beginners is to learn to write decent looking single strokes. The reason is simply that a single ugly stroke is enough to destroy the overall beauty of a character. So when grading a character, we first estimate a visual appearance score for each of the individual strokes.

In our ellipse based representation of a stroke, each stroke contains a series of points on its skeleton, and each point has a corresponding covering ellipse. We thus have a 2D curve, $\mathbf{K}$, for the skeleton of the stroke and another two 1D curves, $\mathbf{Ma}$ and $\mathbf{Mi}$, for the major and minor radii of the covering ellipse. We compute a minimum distance from a point on the skeleton to a point on the stroke contour for each pixel on the skeleton, which results in an offset distance curve $\mathbf{D}$. Before extracting any features for these curves, we apply a Fourier transformation to discard the low frequency components from the curve of $\mathbf{K}$ to get the curve $\mathbf{S}$ which only indicates the local shape of the skeleton. This can discard much of the semantics of the particular stroke and thus yield a more compact representation for shape feature comparison across different strokes; this effectively increases the generality of our training samples. For Chinese characters in particular, there can be easily hundreds of distinct shapes for a single stroke. Without such a content invariant representation, we would have labored intensively to collect a large number of training samples.

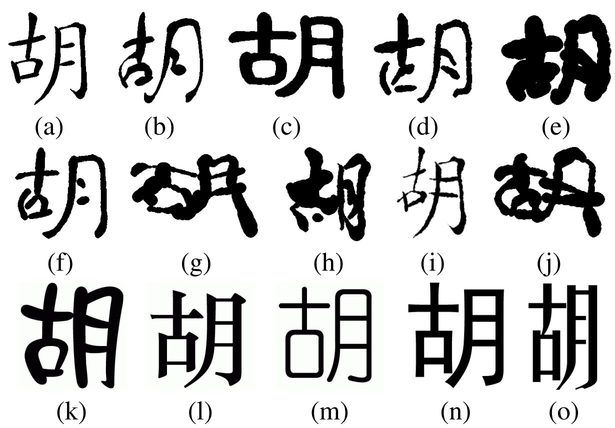


Figure 3: Stroke spatial relationship grading experiment. (a)–(e) are five training characters used in this experiment. (a)–(c) are positive examples and (d) and (e) are negative examples. (f)–(j) are the characters graded as aesthetically unacceptable by the spatial relationship analysis. Their respective scores are: 44.7, 12.7, 7.3, 64.7, 5.4. (k)–(o) are aesthetically acceptable results as judged by our grading algorithm. Their respective scores are: 80.1, 99.0, 99.6, 88.5, 99.8.

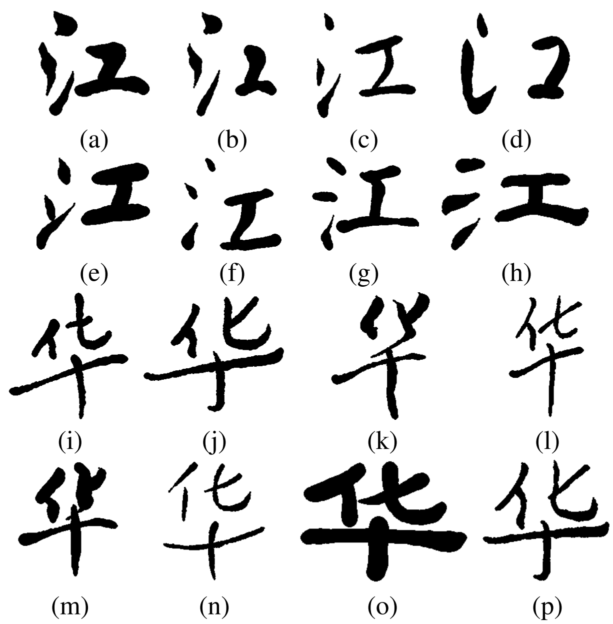


Figure 4: Automatically generated calligraphy using our system with the proposed grading method serving as visual quality control.

Now we have a set of curves $\omega = (\mathbf{S}_x, \mathbf{S}_y, \mathbf{Ma}, \mathbf{Mi}, \mathbf{D})$, where $\mathbf{S}_x$ and $\mathbf{S}_y$ are the x, y components of the 2D curve $\mathbf{S}$. We then compute the associated derivative curves for each of them, getting another set of curves $\omega' = (\mathbf{S}'_x, \mathbf{S}'_y, \mathbf{Ma}', \mathbf{Mi}', \mathbf{D}')$. With both ω and ω', we can compute the shape features of the curves for use in both the learning and the grading process. For each curve $\mathbf{C}$ above, which is a 1D signal, we obtain its largest element $\mathbf{C}_{max}$, the average value $\mathbf{C}_{ave}$ and its median value $\mathbf{C}_{med}$. The set of features for input into the neural network is derived as: $\mathbf{F} \triangleq \{\mathbf{C}_{max} | \mathbf{C} \in \Theta\} \bigcup \{\mathbf{C}_{ave} | \mathbf{C} \in \Theta\} \bigcup \{\mathbf{C}_{med} | \mathbf{C} \in \Theta\} \bigcup \vartheta$, where $\Theta \triangleq \omega \bigcup \omega'$ and ϑ are defined below. In our experiments, we find those feature terms in ϑ_{ave} work most discriminatively.

$$\begin{cases} \vartheta & \triangleq \quad \vartheta_{ave} \bigcup \vartheta_{max} \bigcup \vartheta_{med} \\ \vartheta_{ave} & \triangleq \quad \{\frac{\mathbf{C}'_{max}}{\mathbf{C}_{ave}}, \frac{\mathbf{C}'_{ave}}{\mathbf{C}_{ave}}, \frac{\mathbf{C}'_{med}}{\mathbf{C}_{ave}} | \mathbf{C} \in \omega\} \\ \vartheta_{max} & \triangleq \quad \{\frac{\mathbf{C}'_{max}}{\mathbf{C}_{max}}, \frac{\mathbf{C}'_{ave}}{\mathbf{C}_{max}}, \frac{\mathbf{C}'_{med}}{\mathbf{C}_{max}} | \mathbf{C} \in \omega\} \\ \vartheta_{med} & \triangleq \quad \{\frac{\mathbf{C}'_{max}}{\mathbf{C}_{med}}, \frac{\mathbf{C}'_{ave}}{\mathbf{C}_{med}}, \frac{\mathbf{C}'_{med}}{\mathbf{C}_{med}} | \mathbf{C} \in \omega\} \end{cases} . \tag{1}$$

To label the training examples, we hired a calligraphy expert who manually gave a score to each stroke. For his convenience, only three values, 10, 60, 95 out of a full score of 100, are actually assigned even though the expert could surely give more elaborate grades. We feed a training set containing a total of 2500 such labeled single stroke examples, coming from around 500 characters, into the back-propagation neural network and train it over 10000 iterations. The output score evaluating the appearance of a single stroke falls in the range [0, 100].

Grading the Spatial Layout of Strokes

As important as the appearance of single strokes is the way the strokes are spatially arranged to compose a character. The visual qualities of these individual strokes interact with one another to form the overall visual impression of the whole character, in much the same way as the looks of our facial parts mutually interact to form the overall appearance of a human face. For Chinese characters, these spatial arrangements not only affect the aesthetic appearance, but can also lead to different readings as to what the characters are. Sometimes a minute change of the spatial relationship between strokes can result in an entirely different character being perceived, and not merely the same character written in a different style. This poses much challenge to our algorithmic design.

For every pair of strokes, x, y, we compute the maximum, minimum and mean distances, $l_{max}(x, y), l_{min}(x, y), l_{mean}(x, y)$ from a point on one curve to a point on the other curve. These values can describe both the topological and the spatial relationship between the strokes. For example, we can easily determine by these values whether the two strokes intersect or how much they overlap if intersecting. However, these three values may not tell us the strokes' relative position precisely, which is important for the overall visual appearance or determining the identity of the character. To capture that, we draw a bounding box for each stroke[2], and then compute the horizontal, vertical, and planar overlap of

[2]A bounding box is the minimum rectangle that is parallel to the X-Y axis and which includes all the parts of the stroke.

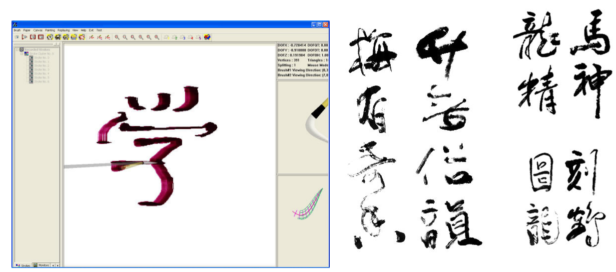
the pair of bounding boxes. We denote the computed values as $B_h(x, y), B_v(x, y), B_p(x, y)$. This is similar to the measurement of the degree of overlap between shapes in (Xu *et al.* 2005). Assume a character has n strokes, doing the above gives us six $n \times n$ matrices $M_{max}, M_{min}, M_{mean}, M_h, M_v, M_p$ where the element on the i-th row and j-th column of M is taken from a corresponding value computed for the stroke pair of the i-th stroke and the j-th stroke.

The next step is to compute the features of these matrices. For a character α, we can find its corresponding character $\widetilde{\alpha}$ written in Kai style. We then compute the above six matrices for each of them, and then do a matrix deduction between the corresponding pair of matrices. The results are denoted as $Q_{max}, Q_{min}, Q_{mean}, Q_h, Q_v, Q_p$. Such a deduction operation attempts to derive a more or less content invariant version of the spatial relationships between the constituent strokes of a character; the benefit is analogous to the gain from removing the low frequency components when grading individual strokes (see the discussion in the previous subsection). For each of the resultant matrices, we further compute its inverse matrix, thus obtaining a total of 12 matrices. For simplicity, we name them as Q_i $(i = 1, \cdots, 12)$. Lastly, for each such matrix, we compute its maximum element value φ_{max}, minimum element value φ_{min}, maximum absolute value φ_{maxa}, mean element value φ_{mean}, median element value φ_{med}, as well as its first three eigenvalues $\lambda_1, \lambda_2, \lambda_3$. This gives a set of eight feature values for a matrix, and thus a total of 96 features for a character. The acquisition of the training labels is done in the same way as single-stroke appearance grading. We use a back-propagation neural network and train it through 10000 iterations. Because of computational overheads, for demonstration in this paper, we only train the network using 100 most frequently used characters. The training set contains six different styles of these characters as provided by a Chinese calligraphy font system, which are all good looking, and another six styles of naive or ugly looking samples created by novices or untrained writers. Our immediate future work will include applying more training samples and see how our method will scale up.

Finally, the overall score of a character is obtained through yet another neural network. The training examples are labeled by a human expert while the input to the network is just the topological score of the character and all the shape scores for its composite strokes. We use a dedicated neural network for producing the overall scores of characters having a certain number of strokes. The examples used for training each of these neural networks are 50 characters, of which each character is written in six different styles.

Generating Chinese Calligraphy

(Xu *et al.* 2005) proposed an intelligent system which is able to generate a great variety of stylistic calligraphic characters. However, only a subset of the generated results are aesthetically pleasing. This is because of the lack of a built-in judging mechanism. With our proposed calligraphy aesthetics grading method, an elaborate and practical visual quality control module can be added in the system. We did, and the result represents a significant performance improvement in terms of the quality of the artistic calligraphy being generated.



(a) (b)

Figure 5: Intelligent calligraphy writing tutoring system: a screenshot (a), and some calligraphy created interactively using the system (b).

Intelligent Calligraphy Tutoring System

We have also developed a tutoring system to support interactive calligraphy learning and writing. The major functionality of our tutoring system is to alert a novice of visually unpleasing strokes and to suggest improvements. Although our tutoring system cannot at this stage create calligraphy in the conventional sense (just like a human calligrapher), it provides prompt feedbacks and useful exemplifications to assist the human learning process. A screen shot of the running system is shown in Fig. 5(a). With the system, the end user manipulates a tablet pen corresponding to a hairy brush to do calligraphy writing. Signals from the tablet pen carrying 5 degrees of freedom are processed in the system in real time to control the position, orientation and elevation of the virtual brush. Fig. 5(b) shows some "e-calligraphy" written using our system. The intelligence of this tutoring system can be easily seen when during the online interactive user writing process, the score for the current writing stroke is fed back to the user in real time. This suggests an opportunity for them to make corrections to the strokes on the spot. The feedback includes suggestions on the appearance of both the individual strokes and the spatial relationships between the strokes. The system would identify strokes that are potentially unpleasing, and supply a candidate set of automatic correction plans for the user to choose from or for the user's reference. Such an interactive feedback is enormously helpful for the calligraphy beginners.

Experiment Results

Our implementation uses Microsoft Visual C++ and runs on a PC with an Intel Pentium4 3.0 GHz CPU and 1 G memory. It achieves "interactive" performance. Ten graduate students have tried the system, and some loved using the system for generating calligraphy for greeting cards. Fig. 2 shows our single stroke aesthetics grading results alongside some of the training samples. To test the learnability of our method, for this example, we only use 50 single strokes, including both negative and positive samples, which were collected from 10 characters in a training set. If a stroke is graded below the score of 50, we would reject it. The experiment results agree surprisingly well with the ground-truth data given by a calligraphy expert. Fig. 3 reports a stroke spatial relationship grading experiment. Again, to show the effectiveness of learning of our method, we restrict to using only five training samples in this controlled experiment. Fig. 4 shows some automatically generated calligraphic characters using our proposed grading method for visual quality control. More results and supplementary materials are available on our project website **http://www.cs.hku.hk/∼songhua/calligraphy**.

Conclusion

This paper describes an integrated intelligent system covering the functions of aesthetic calligraphy decomposition, visual quality evaluation, automatic calligraphy generation, and calligraphy writing tutoring and correction. These functions are achieved through a collaborative design based upon existent AI techniques and machine learning algorithms. We also propose an intelligent user interface for the intervention of the user in order to circumvent the extremely difficult problem of correctly segmenting cursive characters. Even though each of the employed AI techniques has limits and shortcomings, when organized together in the way as reported here, they achieve satisfactory overall intelligent behaviors—to learn, create and tutor aesthetic Chinese calligraphy.

Acknowledgements

This work is supported by a Hong Kong RGC CERG grant (7145/05E) and the National Natural Science Foundation of China (No. 60533090).

References

Avizzano, C. A.; Solis, J.; and Bergamasco, M. 2002. Teaching to write Japanese characters using a haptic interface. In *Proc. of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 255–262.

Bates, J. 1994. The role of emotion in believable agents. *Commun. ACM* 37(7):122–125.

Hertzmann, A. 2003. Tutorial: A survey of stroke-based rendering. *IEEE Comput. Graph. Appl.* 23(4):70–81.

Knuth, D. E. 1986. *The METAFONTbook*. Reading, Massachusetts: Addison-Wesley.

Liu, K.; Huang, Y. S.; and Suen, C. Y. 1999. Identification of fork points on the skeletons of handwritten chinese characters. *IEEE Trans. Pattern Anal. Mach. Intell.* 21(10):1095–1100.

Liu, C.-L.; Kim, I.-J.; and Kim, J. H. 2001. Model-based stroke extraction and matching for handwritten Chinese character recognition. *Pattern Recognition* 34(12):2339–2352.

McCorduck, P. 1990. *Aaron's Code: Meta-Art, Artificial Intelligence and the Work of Harold Cohen*. W. H. Freeman & Co.

Neusius, C., and Olszewski, J. 1994. A noniterative thinning algorithm. *ACM Trans. Mathematical Software* 20(1):5–20.

Picard, R. W. 2000. *Affective Computing*. The MIT Press.

Roads, C. 1985. Research in music and artificial intelligence. *ACM Comput. Surv.* 17(2):163–190.

Rocha, J., and Pavlidis, T. 1994. A shape analysis model with applications to a character recognition system. *IEEE Trans. Pattern Anal. Mach. Intell.* 16(4):393–404.

Shin, J. 2004. On-line cursive hangul recognition that uses DP matching to detect key segmentation points. *Pattern Recognition* 37(11):2101–2112.

Sundar, H.; Silver, D.; Gagvani, N.; and Dickinson, S. 2003. Skeleton based shape matching and retrieval. In *Proc. of Shape Modeling Intel. (SMI)*, 130–139.

Turing, A. M. 1950. Computing machinery and intelligence. *Mind* 49:433–460.

van Eede, M.; Macrini, D.; Telea, A.; Sminchisescu, C.; and Dickinson, S. S. 2006. Canonical skeletons for shape matching. In *Proc. of the 18th Intel. Conf. on Pattern Recognition (ICPR)*, 64–69.

Xu, S.; Lau, F. C.; Cheung, K.-W.; and Pan, Y. 2004. Automatic generation of artistic Chinese calligraphy. In *Proc. of the 19th National Conf. on Art. Intell. and 16th Conf. on Innovative Appl. of Art. Intell. (AAAI/IAAI)*, 937–943.

Xu, S.; Lau, F. C.; Cheung, W. K.; and Pan, Y. 2005. Automatic generation of artistic Chinese calligraphy. *IEEE Intell. Systems* 20(3):32–39.

Xu, S.; Xu, Y.; Kang, S. B.; Salesin, D. H.; Pan, Y.; and Shum, H.-Y. 2006. Animating Chinese paintings through stroke-based decomposition. *ACM Trans. Graph.* 25(2):239–267.

Xu, S.; Lau, F. C.; and Pan, Y. 2003. Automatic artistic calligraphy generation. Technical Report HKU-CS-TR-2003-02, Department of Computer Science, The University of Hong Kong.