Automatic Generation of Artistic Chinese Calligraphy

Songhua Xu, *Zhejiang University*

Francis C.M. Lau, *University of Hong Kong*

William K. Cheung, *Hong Kong Baptist University*

Yunhe Pan, *Zhejiang University*

A novel intelligent system uses a constraint-based analogous-reasoning process to automatically generate original Chinese calligraphy that meets visually aesthetic requirements.

Chinese calligraphy is among the finest and most important of all Chinese art forms and an inseparable part of Chinese history. Its delicate aesthetic effects are generally considered to be unique among all calligraphic arts. Its subtle power is integral to traditional Chinese painting, where—as figure 1a shows—calligraphy is not just an annotation but also a stylized visual component affecting the viewer's emotional response to a painting. This emotional effect also explains why, as figure 1b shows, calligraphy is often preferred to printed type in Asian banners, signage, newspaper mastheads, and other promotional contexts. The 2008 Beijing Olympics Games logo (http://en.beijing-2008.org) is a recent example.

Chinese calligraphers predominantly use a soft hair brush. Generating artistically appealing calligraphy with the brush can be highly challenging. The brush-stroke shapes as well as the topology over multiple strokes are often very complex. The Chinese language's large character set—more than 3,000 commonly used characters—presents a problem all its own. Being able to master some of the characters doesn't mean that you can also write the other characters as satisfactorily. Similarly, mastery of one or more styles doesn't necessarily indicate mastery in other styles, let alone creativity in generating new styles. This is where computers can help.

In the digital world, calligraphic art is most often applied to creating typographic or artistic fonts for printing or display; Donald Knuth has done pioneering work in this area.[1] We propose an intelligent system that can automatically create novel, aesthetically appealing Chinese calligraphy from a few training examples of existing calligraphic styles. To demonstrate the proposed methodology's feasibility, we have implemented a prototype system that automatically generates new Chinese calligraphic art from a small training set—typically, fewer than 10 samples for each character. To the best of our knowledge, no other published work uses our approach. One remotely related project uses analogous reasoning to simulate the creativity in jazz performance and to model other artistic activities from the simulation.[2]

Overall approach and system architecture

Let $\mathcal{P}$ denote a model with a parameterization E that is flexible enough to represent a class of highly deformable shapes—different Chinese character styles, in our case. Normally, constructing a flexible model requires significant effort. At the same time, arbitrary instantiations from such models can easily

1541-1672/05/$20.00 © 2005 IEEE
Published by the IEEE Computer Society

generate unacceptable results. Thus, a model-based approach to generating novel and yet aesthetically appealing calligraphy is by no means straightforward. Our approach uses a constraint-based analogous-reasoning process (ARP), which we apply to a given set of training examples. Analogous reasoning basically fuses knowledge from multiple sources to support a restricted form of reasoning.[3] In our case, the knowledge sources are *training examples*, which are in the form of images.

In our experiments using the prototype system, the training examples come from printed "copybooks" that present multiple calligraphic styles. Because Chinese characters derived from pictographs, which evolved over time into symbols, many basic features recur in different Chinese characters. To take advantage of this redundancy, we devised a hierarchical representation for Chinese characters as the basis for our process. The proposed ARP consists of three major phases:

- *Shape decomposition*. Decomposing (or recovering) the calligraphic shapes of a given training example is equivalent to the problem of extracting structural features for constructing a reference model P. The reference model is an instance of the model $\mathcal{P}$ that best represents the input example. The underlying mechanism in our approach is character stroke segmentation and extraction.
- *Calligraphy model creation from examples*. Given n reference models $\{P_i\}$, where i is the index of the reference model constructed from a set of training examples, we can define a family of novel shapes $P(\omega)$ by blending the reference models: first, we identify the corresponding structural features among the reference models; then, we combine the aligned models by interpolating or extrapolating the parameterizations $\{E\}$. The newly derived shape family is essentially a "reparameterization" via the blending parameters, ω, which control the contribution of each training example.
- *Artistic calligraphy generation*. Given $P(\omega)$ and a set of aesthetics-related geometrical constraints, we identify some ω that satisfies the given constraints.

Figure 1. Artistic Chinese calligraphy in Asian societies: (a) Chinese painting with calligraphy; (b) top—ceiling of a Kong Zi (Confucius) temple; bottom—masthead of the *China Daily* newspaper.

Figure 2 shows the overall architecture of the proposed intelligent calligraphy generation system.

Character representation

Our proposed system decomposes Chinese calligraphy into the six levels shown in figure 3a: constructive ellipse, primitive stroke, compound stroke, radical, single character, and complete artwork. We adopted parametric representations at all levels. Taken

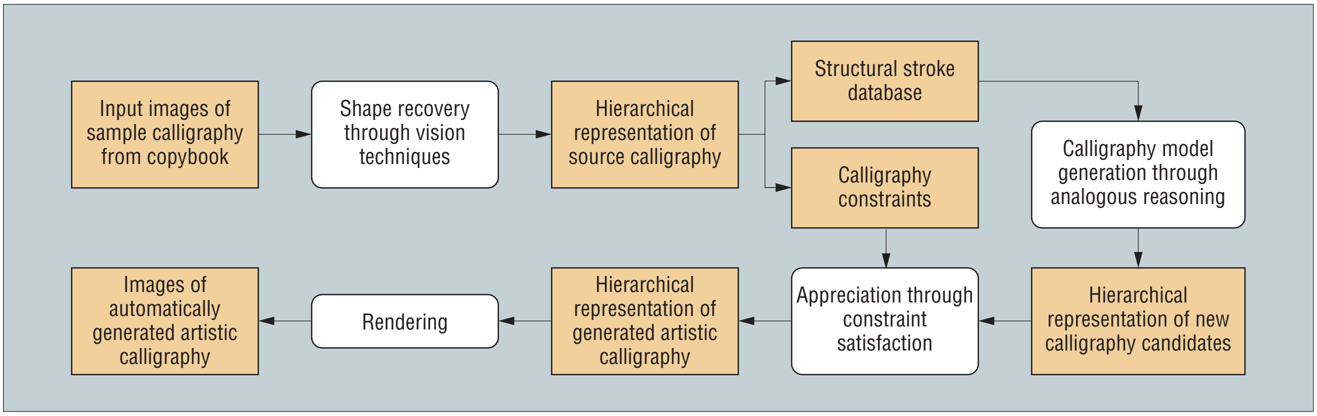

Input images of sample calligraphy from copybook
Shape recovery through vision techniques
Hierarchical representation of source calligraphy
Structural stroke database
Calligraphy constraints
Calligraphy model generation through analogous reasoning
Hierarchical representation of new calligraphy candidates
Appreciation through constraint satisfaction
Hierarchical representation of generated artistic calligraphy
Rendering
Images of automatically generated artistic calligraphy

Figure 2. System architecture for intelligent calligraphy generation.

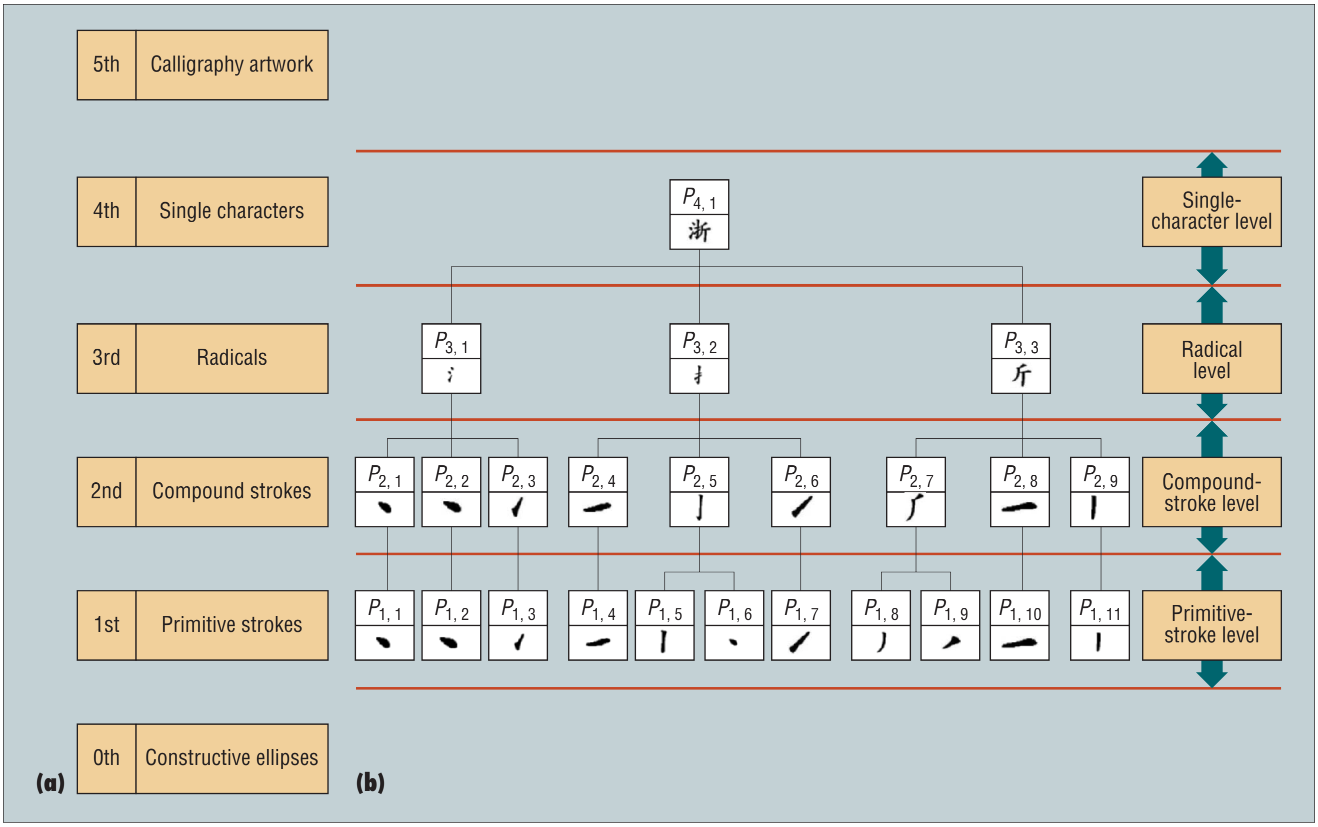


Figure 3. Chinese calligraphy representation: (a) six-level representation hierarchy; (b) four levels in the representation for the Chinese character "zhe."

together, these representations form the parameter space *E* for modeling Chinese calligraphy artwork.

Prototype implementation

In the prototype system, we've implemented five typical, frequently occurring primitive strokes: points, horizontals, verticals, left slants, and right slants. Figure 4 shows these strokes as well as 24 typical, frequently occurring compound strokes and 36 radicals. Figure 3b shows the hierarchical representation of the Chinese character "zhe," as in "Zhejiang," a scenic coastal province and the home of Zhejiang University.

Level 0 of the hierarchical representation views an artwork as a collection of *constructive ellipses* (see figure 3a). The system will render the artwork's "image" as the image space regions that the constructive ellipses cover. This representation is inspired by the Blum model,[4] which defines a zonary area by an ellipse moving along a predefined curve. A 4 × 1 matrix parameterizes each constructive ellipse: two rows store the coordinates of the constructive ellipse's center, and the other two rows store the lengths of its major and minor axes.

Traversing the hierarchy from the bottom up, the system first "lines up" the constructive ellipses to form *primitive strokes* (level 1). Then, using shape grammar rules, it combines primitive strokes to form *compound strokes* (level 2), which are subsequently combined to form *radicals* (level 3). By grouping radicals according to their spatial proximity, the system forms *single characters* (level 4). It blends learned examples of the same character in different styles into a flexible character model.

Finally, the top-level constructive element is *calligraphy artwork* (level 5), which might combine more than one character.

Advantages of the hierarchical parametric approach

Because our approach generates new calligraphic styles by reasoning from a set of existing styles, it belongs to the hard domain of qualitative reasoning. Our parametric representation offers a tool to attack the challenging qualitative-reasoning problem through quantitative means—analogous reasoning together with aesthetic constraint satisfaction.

Hierarchical representation enables efficient local learning of constructive elements and reduces the huge global-knowledge representation space to only local shape-variation characterizations. It also supports efficient retrieval (and thus reuse) of past calligraphy artwork reasoning results.

The hierarchical parametric approach can represent all calligraphy styles—including cursive styles that are heavily deformed and distorted—in a uniform six-level hierarchy, and it can process the characters using the same reasoning pipeline. This increases our system's capability to learn and generate cursive calligraphy, which is an important aspect of calligraphic aesthetics.

Shape decomposition

In the first of its three phases, our system extracts hierarchical and parametric representations from the knowledge sources, which are static images of calligraphy artwork.

Extracting levels 0–1 elements

To extract primitive strokes and thus the corresponding constructive ellipses from a training example, we first compute the input image's skeleton—that is, a close approximation to the brush's actual trajectory when the calligrapher created the artwork.

Various approaches exist for skeletonizing binary character images. We employed the algorithm proposed by Rong He and Hong Yan,[5] which composes the extracted skeleton from segmented *primitive strokes*. Figure 5 gives a step-by-step illustration. Such a stroke decomposition is by no means optimal, and our system can benefit from any improved decomposition algorithm.

To further enhance the robustness of the stroke identification step, the system uses several structural variants of the five primitive strokes in figure 4. Once it identifies the primitive stroke skeletons, the system uses Bresenham's ellipse rasterization algorithm[6] to compute all the constructive ellipses.

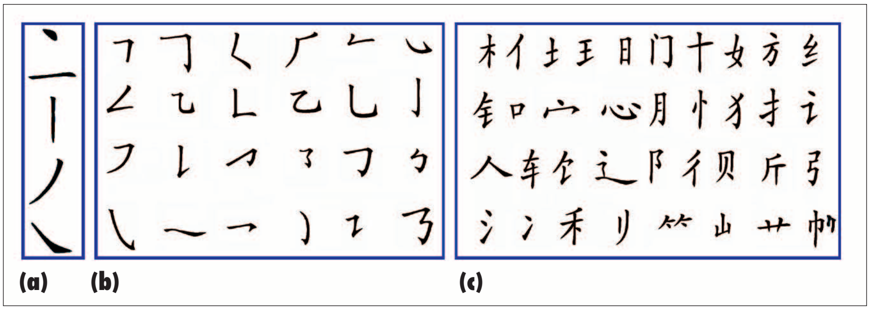


Figure 4. (a) Five primitive strokes, (b) 24 compound strokes, and (c) 36 radicals.

Extracting levels 2–3 elements

We identify compound strokes and radicals by analyzing the spatial relation between the primitive and compound strokes, respectively. The analysis uses carefully designed shape grammar production rules. The syntactic description of any constructive element is represented using the production system's syntax and generated using rule deduction.

As an example, the shape production rule for the compound stroke in the upper leftmost corner of figure 4b, denoted as CS1, is as follows:

```
IF horizontal(a) AND
vertical(b) AND ontop(a,b) AND onleft(a,b) AND
touch(a,b)
THEN CS1 := {a,b}
```

where horizontal(a), vertical(b), ontop(a,b), onleft(a,b), and touch(a,b) are the predicates indicating the relationships of horizontal primitive stroke, vertical primitive stroke, a on top of b, a on left side of b, and {a,b} touching each other, respectively.

We use fuzzy set theory to increase the extraction process's reliability.[7] This approach lets us associate a confidence value with each shape production rule via the deduction process. The system derives an overall confidence value for the shape grammar production from the confidence values of all its statements. Then it applies the rule that yields the highest overall confidence for the corresponding stroke composition.

Extracting level 4 elements

To extract the level 4 constructive elements, we must determine which radicals can be grouped to form a character. This is equivalent to the well-known "character segmentation" problem in pattern recognition. In our system, we use standard projection analysis[8] to segment the individual characters in a calligraphy artwork.

Calligraphy model creation from examples

To generate new calligraphic art, we apply analogous reasoning to a training set of different calligraphic styles. Our approach has roots in Herbert A. Simon's notion of artistic design generation and synthesis, discussed in 1975,[3] and in Mark Keane's later application of analogical mechanisms to problem solving.[9] In general, we can view ARP as a process that synthesizes new knowledge—shapes, in our case—by fusing or blending with existing, independent knowledge sources—that is, the training examples. To support the fusion, the process must establish feature correspondence between the knowledge sources.

In principle, we can apply ARP at different hierarchical levels, resulting in different artistic effects. Assume that we apply it to $P_{k,m}$, the *m*th constructive element at the *k*th

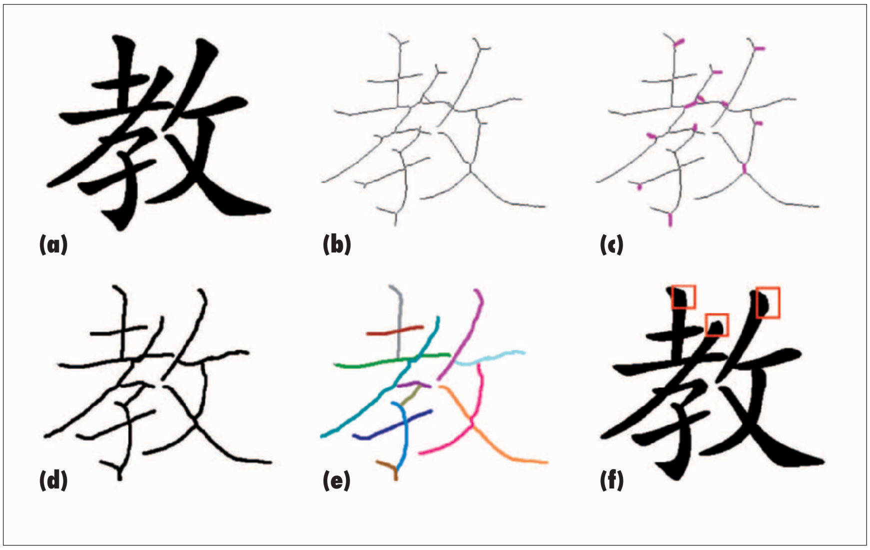


Figure 5. Extracting levels 0–1 elements of a character from its image: (a) the input image, (b) the raw skeleton computed using a thinning algorithm, (c) the plausible short branches detected and color marked, (d) the short branches identified and removed, (e) the skeleton segmented into different strokes in the character and color coded, and (f) the reconstructed character using the extracted levels 0–1 elements. Note that the reconstructed image (f) and the original image (a) have slight differences at the ends of some strokes, as shown in the red rectangles.

level of the hierarchical representation, and that n different versions of $P_{k,m}$ exist, $P^1_{k,m}$, …, $P^n_{k,m}$, derived from n training examples—the independent ARP knowledge sources. The ARP result is denoted as $P^r_{k,m}$. We can then state the general mathematical principle:

$$P^r_{k,m} = \sum_{i=1}^{n} \omega^i P^i_{k,m}$$

where $\omega^i (i = 1, \ldots, n)$ is defined as the analogous reasoning intensity for $P^i_{k,m}$ with the constraint that $\sum_{i=1}^{n} \omega^i = 1$.

Obviously, a $P^i_{k,m}$ with a higher ω^i value would contribute more to the overall reasoning result.

We can interpret the proposed ARP as either an interpolation or an extrapolation process. We're assuming a one-to-one correspondence among different versions of $P_{k,m}$. In reality, a constructive element, such as a constructive ellipse or a primitive stroke, derived from a training example can correspond to one element in another training example in more than one way. The process therefore requires a feature correspondence step before it can blend together features extracted from the different examples. In our intelligent calligraphy generation system, the user can either adjust all the analogous reasoning intensities manually through a graphical interface or generate them randomly. The intensities are fed to a subsequent phase that automatically filters out those that violate aesthetic constraints.

Fusing knowledge sources in ARP

To establish the feature correspondence between training examples, we first derive a discrete planar curve for each constructive element $P^i_{k,m}$ using the centers of all the constructive ellipses associated with it. The curve forms the element's skeleton, and we use Pengfei Zhu and Paul Chirlian's algorithm[10] for detecting the critical points on the planar curves as the key points for setting up the correspondence. Details of this process are available online at http://pantheon.yale.edu/~sx25/ca.

In our application, we first assume the shape of a constructive element in the font style "Kai" (the GB2312 character set used in the recent version of Microsoft Word) to be the standard reference, which we denote as $P^{\text{std}}_{k,m}$. Because the shape decomposition phase has already extracted the $P_{k,m}$'s shape, we can easily compute the deviation $F^i_{k,m}$ by which the shape of the ith source $P^i_{k,m}$ differs from that of $P^{\text{std}}_{k,m}$, as follows:

$$F^i_{k,m} \triangleq P^i_{k,m} \ominus P^{\text{std}}_{k,m}$$

After computing all the deviations of the reasoning sources $F^1_{k,m}, \cdots, F^n_{k,m}$, we can then derive the overall deviation $F^r_{k,m}$ as follows:

$$F^r_{k,m} = \varnothing(F^1_{k,m}, \cdots, F^n_{k,m}, \varpi)$$

where $\varnothing$ is defined as the analogous reasoning mechanism simulation operator, which our prototype system implements as an interpolation or extrapolation process, and ϖ is the *aesthetic viewpoint sequence* dictating the weights and contribution order for different sources. The ordered set of intensities is ARP's *viewpoint sequence*. So not only different reasoning intensities affect the final output, but also different orders of the training examples.

Finally, by adding back the $P^{\text{std}}_{k,m}$ shape (that is, the standard constructive element associated with ARP's reasoning result $P^r_{k,m}$), we obtain

$$P^r_{k,m} = F^r_{k,m} \oplus P^{\text{std}}_{k,m}$$

where the operator $\oplus$ is a reverse function of the operator $\ominus$.

Note that the system can apply ARP not only to the constructive elements from all the reasoning sources but also to some topological constructors in the form of geometric transformation matrices for the corresponding constructive elements. The topological constructors further increase the reasoning power. Simple ARP simulation operators for the topological constructors include arithmetic mean, geometric mean, and harmonic mean.

A computational-psychology perspective

If all the knowledge sources' intensities fall within (0, 1), ARP is simulated using an interpolation process; otherwise, it's simulated using an extrapolation process. From a psychological point of view, the negative values for the reasoning intensities reflect the brain's inverse reasoning activities, which treat certain input source knowledge as negative examples.

By contrast, positive values correspond to exaggeration of brain activities, in which the larger an input example's reasoning intensity, the more heavily the generated result will follow the input source knowledge's style. When the number of knowledge sources is greater than two, ARP mimics combined thinking activity, which will make use of several knowledge reference cases during the reasoning process.

Artistic calligraphy generation

With the mechanisms of calligraphic shape decomposition and model creation from examples in place, we can easily generate candidates for original calligraphy artwork by randomly perturbing the reasoning intensities. The analogous reasoning steps apply equally as well to a single parameter of a constructive ellipse as they do to all the parameters of a character. This provides a highly flexible system for varying shapes to generate many possible candidates.

A filtering step ensures that only candidates meeting aesthetic requirements are output.

Extracting aesthetic constraints from training examples

Aesthetic constraints are criteria that quantify the aesthetic quality of a candidate or its parts. There are two categories:

- constraints for the visual appearance of a constructive element, and
- constraints for the spatial relationship between neighboring constructive elements.

Fortunately, the proposed ARP can automatically guarantee satisfaction of the first constraint—at least under most circumstances—because the constructive elements are parametric. Therefore, we need focus only on deriving and applying constraints of the second type to guarantee the generation of visually pleasing calligraphy.

An important consideration for a quantifiable constraint on aesthetics is the degree of overlap between two constructive elements. Our system uses three types of overlapping between a pair of elements, a and b: the x-dimensional overlapping $\vartheta_x(a, b)$, the y-dimensional overlapping $\vartheta_y(a, b)$, and the area overlapping $\vartheta_s(a, b)$. All three measures are based on the constructive element's bounding boxes. After computing the overlapping measures for all element pairs of the training examples, the system records their upper and lower bounds. The upper bound keeps two neighboring elements in a new calligraphy artwork from being squeezed too closely together while the lower bound keeps them from being too far apart. These overlapping measures then direct the process of generating the upper-level constructive elements from the lower-level ones.

The overall effect is to constrain ARP from perturbing the spatial relationships of each constructive element's subcomponents too much beyond what they are in the training examples. Thus, for example, if ARP generates a calligraphy candidate that contains subconstructive elements whose *x*-dimensional overlapping is smaller than the derived lower bound, the prototype system will reject the candidate.

If needed, end users can relax the upper- and lower-bound constraints to allow for new styles that cannot be easily imagined. In our system, users can adjust the two bounds interactively according to their preferences. Thus, choosing the best values for the two bounds becomes a matter of personal aesthetic taste. According to our experience with the prototype system's behaviors, relaxing or ignoring the ARP constraints seems to create a more cursive and running-style writing.

Using past results for efficient reasoning

You could choose a random process to simulate ARP, but this might be computationally intensive. Reusing past results can make the reasoning process more efficient. In addition, the hierarchical representation lets users reuse whole or partial results. Our proposed system therefore offers a high degree of reusability of past reasoning results.

Experimental results

Figure 6 shows the results our prototype system obtained using six training examples as the input knowledge sources and linear interpolation to simulate the ARP generation step. Figure 7 shows the results of using five training examples and a nonlinear interpolation process. In the latter case, you can easily see the consistency in style among characters within the same newly generated calligraphic piece. Space limitations restrict the results shown to a small set; more results are available at http://pantheon.yale.edu/~sx25/ca.

The results show that our approach can yield novel calligraphy styles given some existing ones. To verify that the system was generating quality outputs, we asked practicing artists, art school professors, and calligraphy enthusiasts to examine the outputs; most of them considered the generated calligraphy to be novel with regard to their writing style.

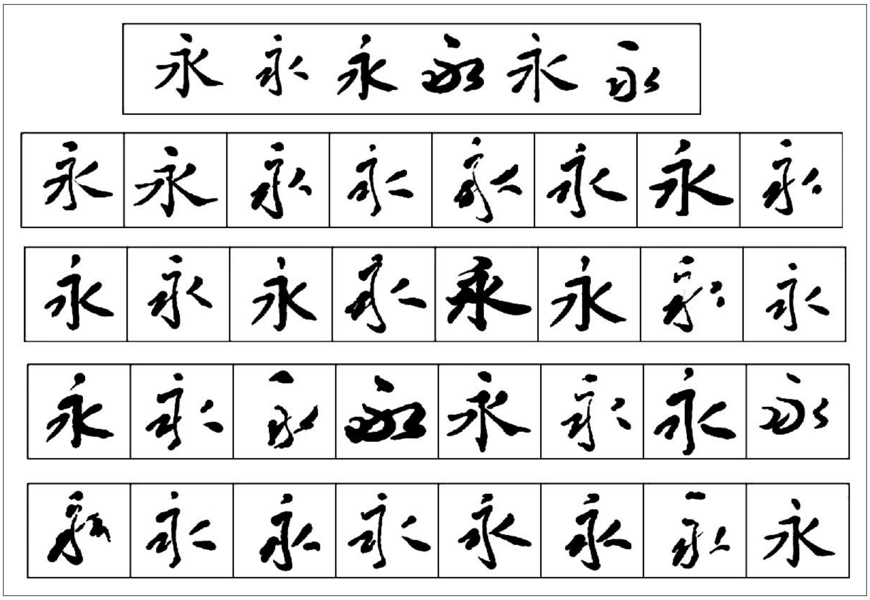


Figure 6. A single character in many styles; the first row is the training examples, and the other rows are automatically generated by our system.

In addition, we analyzed the system's sensitivity in terms of its "creativity" by varying the number of training examples. In this experiment, we used training examples in training examples of different calligraphy styles, including Kai, Li, Xingshu, Weibei, Xingkai, Xingchao, and Kuangchao. We invited six calligraphic fans with at least two years' writing experience and four professional calligraphists to form a review committee. They cast votes on the system-generated calligraphic art. If a result received more

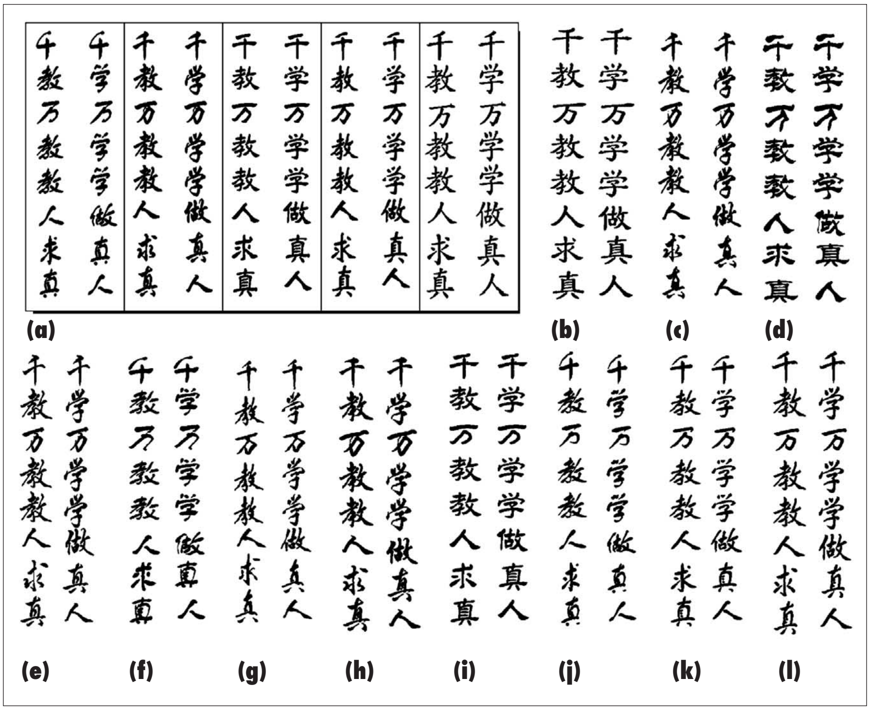


Figure 7. A calligraphic couplet, or epigram, in many styles: (a) the training examples; (b)–(l) selected computer-generated results.

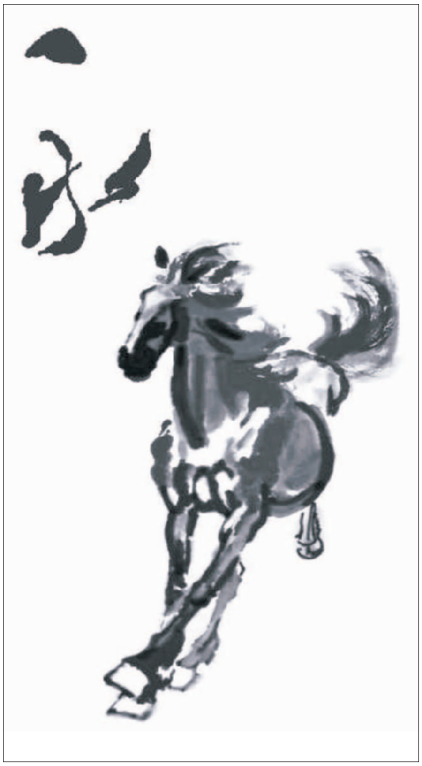


Figure 8. "Forever running." The calligraphy is styled to match a horse created with paint-brush software.

than seven votes, it was considered a new calligraphic work. With more training examples of different styles, the chance of generating a creative and yet aesthetically acceptable calligraphic art increased. Also, nonlinear ARP generated more creative calligraphic art than linear APR. Using from six to seven training examples, the linear process generated about 30 acceptable calligraphic works and the nonlinear process generated more than 50.

Figure 8 shows an interesting example. An artist generated the horse using paint-brush software.[11] Our prototype system automatically generated the calligraphy (the character "forever" in Chinese).

Applications

This system for computer-generated original Chinese calligraphy has several potential applications.

For example, users could generate personalized fonts with it. First, they would import computer-installed fonts. Then our system could compute a flexible model of each character by aligning all corresponding characters of the different fonts. Next, the system would ask for a small set of characters in the user's handwriting as additional knowledge. On the basis of these knowledge sources, the system could simulate the proposed ARP but now with the additional criterion to best match the user inputs. The uniformity of the hierarchical representation means that the resulting set of intensities applies directly to the full character set of all the existing fonts, generating a font customized to the user's handwriting.

Figure 9 shows the results from a simple test. The top row shows handwritten input characters from different users, and the rows below show the characters generated by mimicking the input style. The results are impressive. Of course, if the user's handwriting is so peculiar that it lies outside the feasible space composed from all the existing fonts, the system might fail.

An application along the same lines could make an individual's handwriting style more beautiful, which could improve readers' impressions of the writer. With our system, users can input their handwriting as one ARP source and then specify an appealing existing style as another source. By manually setting the reasoning intensities, users could choose the extent to which they rectified their personal handwriting while still preserving a degree of its unique qualities.

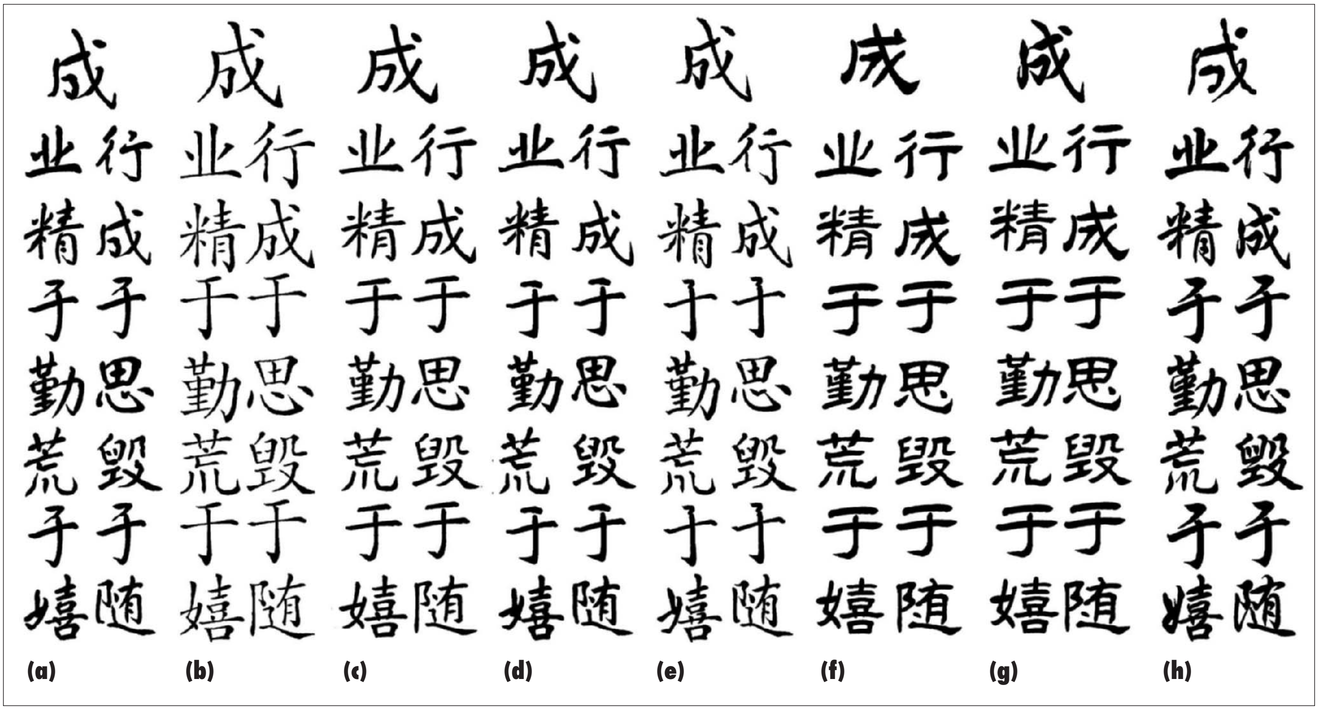


Figure 9. Personalized font generation from a single character. The first row shows a single character written by different users in their respective handwriting styles, and the second row shows automatically generated characters that mimic the captured handwriting style.

The proposed system is essentially a specific version of a deformable model for modeling Chinese characters of different calligraphic styles. The model's representation power makes the system suitable for deformable model-based handwriting recognition. In addition, the unified ARP modeling approach is an excellent candidate for writer adaptation in existing or emerging handwriting recognition systems.

Carnegie Mellon University's CAPTCHA project (www.captcha.net) presents another possible application of our approach. CAPTCHA stands for "Completely Automated Public Turing Test to Tell Computers and Humans Apart." A CAPTCHA is any program that can generate and grade tests that computers cannot pass, and one kind—popular for protecting Web sites from massive automated registrations—uses words written in distorted letters that humans can read but computers cannot. Our system could be used to develop a Chinese CAPTCHA agent that would generate similarly distorted Chinese writing that humans but no machine-computable algorithms could read.

Parametric hierarchical knowledge representation enables computer-generated Chinese calligraphy artwork in a variety of styles—fully automated in real time from a compact set of printed calligraphic inputs. The creative capability of the proposed intelligent system lies mainly in the huge feasible space available for the simulated analogous reasoning process. Our prototype system has generated artwork that can stand among existing instances, whether realistic or completely inventive in appearance.

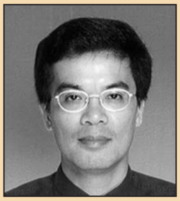
We are considering several extensions to our proposed system. We could extend the approach to other languages. We might also add a feedback component to fine-tune the aesthetic constraints through reinforcement learning. Finally, we could capture and translate the "psychological" states of other media and then link them to the corresponding states in calligraphy—for example, allowing music to direct the generation of calligraphy.

Acknowledgments

We are grateful to the reviewers whose comments have been most useful and encouraging.

The Authors

Songhua Xu is a member of the CAD and CG State Key Lab of China at Zhejiang University and an honorary researcher in the Department of Computer Science at the University of Hong Kong. He is also pursuing computer graphics research with the Computer Science Department at Yale University. His research interests include artificial intelligence, computer graphics, and human-computer interaction. He received his bachelor of engineering summa cum laude in computer science from Zhejiang University. He is a member of the AAAI and Eurographics. Contact him at the CAD and CG State Key Lab of China, Zhejiang Univ., Hangzhou, China, 310027; songhua.xu@yale.edu.



Francis C.M. Lau is head of the Department of Computer Science at the University of Hong Kong. His research interests are in parallel and distributed computing, object-oriented programming, operating systems, Web and Internet computing, computer graphics, and AI. He received his PhD in computer science from University of Waterloo. He is a senior member of the IEEE. Contact him at the Dept. of Computer Science, University of Hong Kong, Hong Kong; fcmlau@cs.hku.hk.

William K. Cheung is an assistant professor in the Department of Computer Science at Hong Kong Baptist University. His research interests include model-based pattern recognition, machine learning, and artificial intelligence with applications to data mining, information extraction, recommender systems, and Web and Grid service management. He received his PhD in computer science from the Hong Kong University of Science and Technology. Contact him at the Dept. of Computer Science, Hong Kong Baptist Univ., Hong Kong; william@comp.hkbu.edu.hk.

Yunhe Pan is a full professor in the Department of Computer Science and president of Zhejiang University, as well as an honorary professor of the University of Hong Kong. His research interests include artificial intelligence, computer graphics, and multimedia computing. He received his MS in computer science and engineering from Zhejiang University. Contact him at the President's Office, Zhejiang Univ., Hangzhou, China, 310027; panyh@zju.edu.cn.

References

1. D.E. Knuth, *Tex and Metafont: New Directions in Typesetting*, Am. Mathematical Soc., 1979.
2. G. Ramalho and J.-G. Ganascia, "Simulating Creativity in Jazz Performance," *Proc.12th Nat'l Conf. Artificial Intelligence* (AAAI 94), 1994, pp. 108–113.
3. H.A. Simon, "Style in Design," *Spatial Synthesis in Computer Aided Building Design*, C.M. Eastman, ed., Applied Science, 1975.
4. H. Blum, "A Transformation for Extracting New Descriptors of Shape," *Models for the Perception of Speech and Visual Form*, W. Wathen-Dunn, ed., MIT Press, 1967, pp. 362–380.
5. R. He and H. Yan, "Stroke Extraction as Preprocessing Step to Improve Thinning Results of Chinese Characters," *Pattern Recognition Letters*, vol. 21, no. 9, 2000, pp. 817–825.
6. J.E. Bresenham, "A Linear Algorithm for Incremental Digital Display of Circular Arcs," *Comm. ACM*, vol. 20, no. 2, 1977, pp. 100–106.
7. H.M. Lee, C.W. Huang, and C.C. Sheu, "A Fuzzy Rule-Based System for Handwritten Chinese Characters Recognition Based on Radical Extraction," *Fuzzy Sets and Systems*, vol. 100, nos. 1–3, 1998, pp. 59–70.
8. B.A. Yanikoglu and P.A. Sandon, "Recognizing Offline Cursive Handwriting," *Proc. Computer Vision and Pattern Recognition*, IEEE CS Press, 1994, pp. 397–403.
9. M. Keane, "Analogical Mechanisms," *Artificial Intelligence Rev.*, vol. 2, no. 4, 1988, pp. 229–250.
10. P. Zhu and P.M. Chirlian, "On Critical Point Detection of Digital Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, 1995, pp. 737–748.
11. S. Xu et al., "Virtual Hairy Brush for Painterly Rendering," *Graphical Models*, vol. 66, no. 5, 2004, pp. 263–302.