HKU CSIS Tech Report TR-2003-02

Automatic Artistic Calligraphy Generation

Songhua Xu[1], Francis C.M. Lau[1], and Yunhe Pan[2]

[1] Department of Computer Science and Information Systems,
The University of Hong Kong, Pokfulam Road, Hong Kong
{songhua, fcmlau}@csis.hku.hk
[2] CAD & CG State Key Lab of China,
Zhejiang University, Hangzhou, P.R.C.
panyh@sun.zju.edu.cn
6 May 2003

Abstract. We introduce a novel intelligent system which can generate new Chinese calligraphic artwork that meets certain aesthetic requirements automatically. In the machine learning phase, parametric representations of the existent calligraphic artwork are derived from input images of calligraphy. Using a six-layer hierarchical representation, the acquired knowledge is organized as a small structural stroke database, which is then exploited by a constraint-based analogous reasoning component to create artwork in new styles. The simulated analogous reasoning can generate new "e-calligraphy", and constraint satisfaction is used to reject the unacceptable results. The combination of knowledge from various input sources creates a huge space for the intelligent system to explore and produce new styles of calligraphy. (http://www.csis.hku.hk/∼songhua/ca/ provides supplementary materials on this paper.)

Category:
H.4 Information Systems Applications: Miscellaneous
I.2.6 Artificial Intelligence: Learning [*Analogies, Knowledge acquisition*]
I.3.5 Computer Graphics: Computational Geometry and Object Modeling [*Geometric algorithms and systems, Hierarchy and geometric transformations*]
I.4.9 Image Processing and Computer Vision: Applications
J.5 Computer Applications: Art and Humanities [*Arts, fine and performing, Fine art*]
Keywords: analogous reasoning, simulated analogous reasoning process, character skeletonization, radical extraction, character segmentation, constructive element, topological constructor, hierarchical representation of calligraphic artwork, constraint satisfaction, degree of interference, non-photorealistic rendering

1 Introduction

Chinese calligraphy is among the finest and most important of all Chinese art forms, and an inseparable part of Chinese history. Other than artists, it has also caught the attention of scientists who are interested in computer-assisted art.

Fig. 1. Chinese painting with calligraphy

Chinese calligraphy is predominantly done using a brush. Computerizing Chinese calligraphy is challenging as the shapes of brush strokes as well as the topology over multiple strokes could be very complex. In comparison, Western calligraphy which is based on Latin alphabets is much simpler and easier to computerize. Chinese calligraphy can convey not just what is in the written message but also the emotions of the writer. The very delicate aesthetic effects achievable by Chinese calligraphy are unique among all calligraphic arts because the normal shape and topological structure of the font in aesthetic Chinese calligraphy can be largely distorted for its better perceptual impression. Chinese calligraphy is also an integral part of traditional Chinese painting. The use is not just for annotation, but also because calligraphy embedded in a painting can affect the overall visual and perhaps also the emotional perception of the viewer. Figure 1 shows an example. As such, many people see calligraphy as a kind of painting. This opinion is particularly widely accepted as to the specific case of Chinese calligraphy.

Calligraphic art is based on font, which is a set of printer's type of the same size and face. Cubic Bezier curves and straight lines can be used to describe font shapes [33, 30]. For artistic rendering, researchers have tried to model the brush

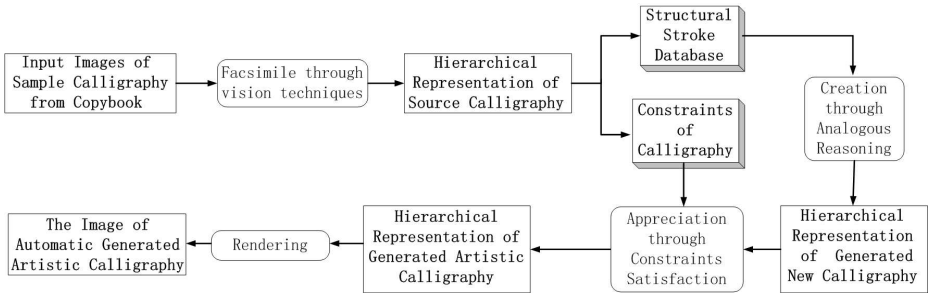
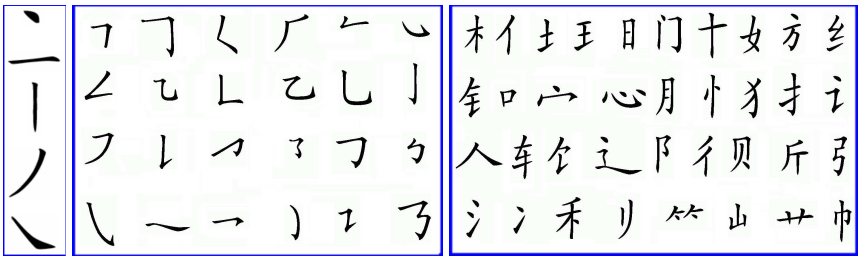


Fig. 2. Architecture of our intelligent calligraphy generation system



(a) (b) (c)

Fig. 3. (a) 5 implemented primitive strokes. (b) 24 implemented compound strokes. (c) 36 implemented radicals

used in calligraphy, such as [28] where the brush is modeled as a collection of bristles which evolve over the course of the stroke. In [29], a virtual brush based on solid modeling was demonstrated as a feasible interactive tool for creating realistic Chinese calligraphic writings. In [7], the authors gave a detailed analysis on the writing effects that hairy brushes can produce. There have also been attempts on automatic generation of new fonts, such as [35] where the authors employed an algebra of geometric shapes to generate fonts by mixing existing fonts. But calligraphy can go beyond the boundaries of fonts; for example, it is possible to mix different styles and sizes of characters in a calligraphic artwork.

There has not been any published work on automatic creation (not just imitation) of beautiful calligraphic artwork using existing calligraphy as learning samples. This paper proposes and describes such an intelligent system. We discuss the underlying principles and theories, and present the calligraphic results generated by a prototype we implemented. Our prototype system is able to generate brand new Chinese calligraphic artwork fully automatically. The number of input training samples used is very small.

Figure 2 shows the overall architecture of our intelligent calligraphy system. At the center of the system is an analogous reasoning component that creates new calligraphic artwork based on the learned samples. The automatically generated calligraphic artwork would satisfy certain aesthetic constraints. In our

experiments, these learning samples come from printed "copybooks". Our simulated analogous reasoning process is essentially data prediction (either interpolation or extrapolation) subject to the aesthetical constraints. For convenience, we abbreviate "analogous reasoning process" to **ARP** and "simulated analogous reasoning process" to **SARP**.

The system has three main components. The first component learns and produces facsimiles of the existent calligraphic artwork in a hierarchical and parametric form; these facsimiles form a calligraphic knowledge base serving as the knowledge source for the **SARP**. The second component generates new calligraphic artwork automatically through the **SARP**. The third component applies constraint satisfaction to admit only those generated results that are aesthetically acceptable. The three components are referred to as the *facsimile component*, the *creation component*, and the *appreciation component*, respectively. To verify that the system was indeed able to generate quality outputs, we devised an experiment and asked practicing artists, art school professors, and amateurs to examine the outputs from our system. The examination results led us to conclude that our approach is practical and the system is capable of generating acceptable outputs.

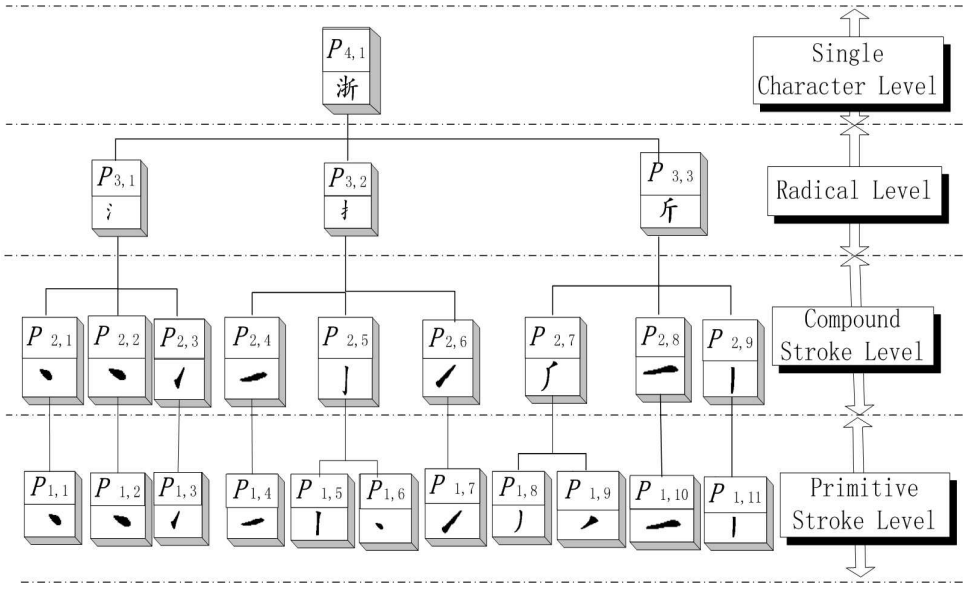
The structure of the paper is as follows. Sections 2 addresses the representation of a calligraphic artwork in a parametric and hierarchical form. Section 3 discusses how input images of sample calligraphic artwork are analyzed and parameterized by machine intelligence techniques. Sections 4 explains the generation of new calligraphic artwork automatically through the **SARP**. Section 5 introduces the appreciation component and how it can filter out unacceptable generated samples. Section 6 shows some actual results generated by our prototype implementation. Section 7 discusses some advanced applications using our method. Section 8 concludes the paper with a discussion on future work. Note that because of space limitation, our discussions stop at the 4-th level, where single characters as a whole are objects for the reasoning process.

2 Hierarchical and Parametric Representation

The earliest Chinese characters are pictographs, which project meanings through shapes and images in an intuitive manner. Over time, these characters gradually became symbols. In our system, we treat Chinese characters and calligraphic artwork as images that are in a parametric form. This facilitates automatic processing of knowledge. We adopt also a knowledge representation that is hierarchical, which helps to increase the speed of the **SARP** and thus the system's performance.

2.1 Hierarchical Representation

It can be easily observed that many local features recur in many different Chinese characters frequently. To capitalize on this image information redundancy, we introduce a hierarchical representation of Chinese calligraphy. A piece of



(a)

Layer Number	Layer Name
the #5th layer	the layer of calligraphic artwork
the #4th layer	the layer of single characters
the #3rd layer	the layer of radicals
the #2nd layer	the layer of compound strokes
the #1st layer	the layer of primitive strokes
the #0th layer	the layer of constructive ellipses

(b)

Fig. 4. (a) Hierarchical representation of a character (only four levels are shown). (b) Six-layer hierarchical representation of calligraphy

Chinese calligraphy as an image is decomposed into six layers (or levels): the constructive ellipse layer, the primitive stroke layer, the compound stroke layer, the radical layer, the single-character layer, and the complete artwork layer (see Figure 4(b)). This hierarchical representation can avoid much redundancy when storing the characters, and its various granularity makes **SARP** more effective and the input reasoning source as well as the output reasoning results more reusable. These six layers represent the calligraphic artwork parametrically. All the input parametric representations of the calligraphic artwork together form a reasoning space for the **SARP** to generate new aesthetic calligraphic artwork automatically.

We can give a definition over the hierarchic representation of calligraphic artwork based on the concept of equivalent relationship. If $\mathbf{R}$ is the equivalent relationship defined over the field of $\mathbf{P} = \{p_1, p_2, \cdots, p_n\}$, i.e. $\mathbf{R}$ is (1) self-reflective (2) symmetrical (3) transitive, field $\mathbf{P}$ can be divided into a collection

of sub-sets $\mathbf{P}_1, \mathbf{P}_2, \cdots, \mathbf{P}_m$ under $\mathbf{R}$. We call p_i is equivalent to p_j if $(p_i, p_j) \in \mathbf{R}$, $1 \leq i, j \leq n$. Using the concept of equivalent relationships, we can now introduce the formal definition for the multi-layer calligraphic artwork representation.

In an image of a piece of calligraphic artwork, we adopt the following five kinds of equivalent relationships to establish our six-level hierarchic representation for one piece of calligraphy: $\mathbf{R}_1$: all the constructive ellipses that compose the same primitive strokes are equivalent to each other; $\mathbf{R}_2$: all the primitive strokes that compose the same compound strokes are equivalent to each other; $\mathbf{R}_3$: all the compound strokes that compose the same radical are equivalent to each other; $\mathbf{R}_4$: all the radicals that compose the same Chinese character are equivalent to each other; $\mathbf{R}_5$: all the characters in the same piece of calligraphic artwork are equivalent to each other.

Suppose in the parametric representation of a calligraphic artwork $\mathbf{C}$, there are num_0 constructive ellipses, denoted as $\mathbf{F}_0 \triangleq \{\mathbf{P}_{0,1}, \mathbf{P}_{0,2}, \cdots, \mathbf{P}_{0,num_0}\}$, where each $\mathbf{P}_{0,i}$ is a constructive ellipse. And $\mathbf{F}_0$ is divided into num_1 equivalent classes (of primitive strokes) under the equivalent relationship of $\mathbf{R}_1$, and denote these classes as $\mathbf{F}_1 \triangleq \{\mathbf{P}_{1,1}, \mathbf{P}_{1,2}, \cdots, \mathbf{P}_{1,num_1}\}$, where each $\mathbf{P}_{1,i}$ is a primitive stroke. These num_1 primitive strokes are further divided into num_2 equivalent classes (of compound strokes) under the equivalent relationship of $\mathbf{R}_2$, and denoted as $\mathbf{F}_2 \triangleq \{\mathbf{P}_{2,1}, \mathbf{P}_{2,2}, \cdots, \mathbf{P}_{2,num_2}\}$, where each $\mathbf{P}_{2,i}$ is a compound stroke. All the compound strokes are divided into num_3 equivalent classes (of radicals) under the equivalent relationship of $\mathbf{R}_3$, and denoted as $\mathbf{F}_3 \triangleq \{\mathbf{P}_{3,1}, \mathbf{P}_{3,2}, \cdots, \mathbf{P}_{3,num_3}\}$, where each $\mathbf{P}_{3,i}$ is a radical. Finally, all the radicals are divided into num_4 equivalent classes (of single Chinese characters) under the equivalent relationship of $\mathbf{R}_4$, and denoted as $\mathbf{F}_4 \triangleq \{\mathbf{P}_{4,1}, \mathbf{P}_{4,2}, \cdots, \mathbf{P}_{4,num_4}\}$, where each $\mathbf{P}_{4,i}$ is a single Chinese character. That is, in a certain calligraphic artwork of Chinese handwriting $\mathbf{C}$, there are num_0 constructive ellipses $\mathbf{P}_{0,i}$, $i \in \{1, 2, \cdots, num_0\}$. Or we can view $\mathbf{C}$ as being composed by num_1 primitive strokes $\mathbf{P}_{1,i}$, $i \in \{1, 2, \cdots, num_1\}$. Namely, $\mathbf{C}$ contains num_2 compound strokes $\mathbf{P}_{2,i}$, $i \in \{1, 2, \cdots, num_2\}$. Or we can say that $\mathbf{C}$ contains num_3 radicals $\mathbf{P}_{3,i}$, $i \in \{1, 2, \cdots, num_3\}$. We can also say that $\mathbf{C}$ contains num_4 single Chinese characters $\mathbf{P}_{4,i}$, $i \in \{1, 2, \cdots, num_4\}$. From the point of view of level 5 in the hierarchical representation, $\mathbf{C}$ is actually $\mathbf{P}_{5,1}$ with $num_5 = 1$.

The hierarchical structural knowledge representation of calligraphic artwork can be stated formally as (1). In (1.2), $q_{k,l} = min\{t | \mathbf{P}_{k-1,t} \overline{\in} \bigcup_{s=1}^{l-1} \mathbf{P}_{k,s}, \mathbf{P}_{k-1,t} \in \mathbf{F}_{k-1}\}$. And in (1), $\mathbf{P}_{k-1,i} \in \mathbf{F}_{k-1}$, $k = 1, \cdots, 5$. Denote the number of elements in set $\mathbf{M}$ as $|\mathbf{M}|$, the relationship (2) holds within the hierarchy of Chinese calligraphic artwork representation.

$$\mathbf{P}_{k,l} \triangleq \begin{cases} \{\mathbf{P}_{k-1,i} | (\mathbf{P}_{k-1,i}, \mathbf{P}_{k-1,1}) \in \mathbf{R}_k\} \ (l = 1) & (1.1) \\ \{\mathbf{P}_{k-1,i} | (\mathbf{P}_{k-1,i}, \mathbf{P}_{k-1,q_{k,l}}) \in \mathbf{R}_k\} \ (l \neq 1) & (1.2) \end{cases} \tag{1}$$

$$|\mathbf{F}_{k-1}| = \sum_{i=1}^{|\mathbf{F}_k|} |\mathbf{P}_{k,i}| = num_{k-1} \ (k = 1, \cdots, 5) \tag{2}$$

The hierarchical representation describes how an artwork is composed from constructive ellipses at the lowest level. Each higher level describes how to gen-

Operators	Definition	Operators	Definition
$\ominus$	(18)	$\oslash$	(21)
$\oplus$	(25)	$\odot$	(27)
$\vartheta_x(), \vartheta_y(), \vartheta_s()$	(28)	$\theta_x(), \theta_y(), \theta_s()$	(29)
$\nabla^b_{m,n}()$	(7)	$\nabla^c_n()$	(9)
$\nabla^o_n()$	(10)	$\nabla^d_n()$	(11)
$\nabla^e_{n,m}()$	(12)	∇^g_t	(16), (17)
$\nabla^f_{n,m}()$	(19)	$\otimes$	(24)

Fig. 5. Operators Defined

erate one level of representation from the information at one level down. It is essentially a tree-like knowledge representation. For the prototype implementation, we selected 5 typical and most frequently occurring primitive strokes: horizontal strokes, vertical strokes, left slanting strokes, right slanting strokes, and point strokes (Figure 3(a)), 24 typical and most frequently occurring compound strokes (Figure 3(b)), and 36 radicals (Figure 3(c)). Figure 4(a) shows an example of hierarchical representation, which is that of the Chinese character "zhe". Note that since the number of compound strokes and radicals implemented in the system is limited because of resource limitations, it is possible that some lower-layer element cannot be combined with other elements in the same level. In this case, that lower-layer element promotes itself to the next level. An example is the primitive stroke $\mathbf{P}_{1,1}$ in Figure 4(a). which becomes the compound stroke $\mathbf{P}_{2,1}$ in the next level. Similarly, it is possible for a radical to degrade to a compound stroke, and then to a primitive stroke.

2.2 Six Levels of Parametric Representation

We denote the i-th constructive element in the k-th level as $\mathbf{P}_{k,i}$, and its matrix form parametric representation as $\mathbf{E}_{k,i}$. If $k \geq 1$, $\mathbf{P}_{k,i}$ must be composed of one or more constructive elements in one level down; we call the latter sub-constructive elements. All the information needed for the composition of $\mathbf{P}_{k,i}$ is stored in $\mathbf{T}_{k,i}$, the *topological constructor* of $\mathbf{P}_{k,i}$.

To derive $\mathbf{E}_{k,i}$, we need first to define several operators for calligraphic knowledge representation and operations to simulate the **ARP**. A quick index of these operators is in Figure 5, which is a list over all the operators defined for knowledge representation and simulating **ARP** with constraint satisfying in our system to generate artistic calligraphy automatically.

We use $\overline{\mathbf{P}}_{k,i}$ to represent the bounding box of the image space that the element $\mathbf{P}_{k,i}$ occupies, that is, $\overline{\mathbf{P}}_{k,i} \triangleq \{\overline{\mathbf{P}}_{k,i}.h, \overline{\mathbf{P}}_{k,i}.w, \overline{\mathbf{P}}_{k,i}.x, \overline{\mathbf{P}}_{k,i}.y\}$, where $\overline{\mathbf{P}}_{k,i}.h$ is the box's height, $\overline{\mathbf{P}}_{k,i}.w$ the box's width, and $(\overline{\mathbf{P}}_{k,i}.x, \overline{\mathbf{P}}_{k,i}.y)$ the box's center. All the coordinates are in the world coordinate system.

In the 0-th level, the calligraphy is viewed as a set of ellipses, denoted as $\mathbf{F}_0$. These ellipses are called the "constructive ellipses" of the calligraphic artwork (Figure 4(b)). For each constructive ellipse, $\mathbf{P}_{0,i}$, let (x_i, y_i) be the center and

a_i and b_i the lengths of its major and minor axis respectively. Then the "image" of the calligraphic artwork $\mathbf{C}$ can be represented as the image area covered by all its constructive ellipses, defined as (3). This representation is inspired by the Blum model [7], in which a zonary area is defined through an ellipse moving along a predefined curve.

$$Img(\mathbf{C}) \triangleq \{(x,y) \in \mathbf{R}^2 | \exists \mathbf{P}_{0,i} \in \mathbf{F}_0, \frac{(x-x_i)^2}{{a_i}^2} + \frac{(y-y_i)^2}{{b_i}^2} \leq 1\} \tag{3}$$

The ranges for the horizontal and vertical coordinates x_i, y_i and the horizontal and vertical distances a_i, b_i are normalized with respect to the bounding box of the constructive ellipse $\overline{\mathbf{P}}_{0,i}$, as defined in (4). The resultant respective values for x_i, y_i, a_i, b_i are denoted as x'_i, y'_i, a'_i, b'_i and recorded in the matrix form representation of $\mathbf{P}_{0,i}$, such that $\mathbf{E}_{0,i} \triangleq (x'_i, y'_i, a'_i, b'_i)^T$.

Suppose that the element $\mathbf{P}_{k+1,1}$ is composed of n elements in the next lower level, $\mathbf{P}_{k,l_1}, \cdots, \mathbf{P}_{k,l_n}$. Then $\mathbf{E}_{k+1,1}$ can be derived by concatenating the matrices $\mathbf{E}_{k,l_1}, \cdots, \mathbf{E}_{k,l_n}$ column by column in sequence. Since the parametric representation of a constructive ellipse is a 4×1 matrix, concatenation at the higher levels will produce matrices having exactly four rows. Each row of the matrix form parametric representation of a constructive element is called a *field* of the element's parametric representation. Different fields of an element can be separately reasoned on.

The parametric representation of each constructive element only records the relative coordinates. The use of relative coordinates makes the representation independent from other elements' representations, and hence reusable in different **SARP**s. Also because of the use of relative coordinates, coordinate transformation is necessary to convert the relative coordinates between different relative coordinate systems. We include the coordinate transformation associated with $\mathbf{E}_{k,l_i}$ in $\mathbf{P}_{k,l_i}$'s topological constructor, $\mathbf{T}_{k,l_i}$.

2.3 Deriving Parametric Representations for Constructive Elements

Level 0 of the Parametric Representation For each constructive element in level 0, i.e. constructive ellipses $\mathbf{P}_{0,i}$ in our hierarchy, we use procedure introduced at Section 3.1 to compute the ellipse's four parameters (x_i, y_i, a_i, b_i). We then employ (4) to convert the absolute coordinates (x'_i, y'_i, a'_i, b'_i) into relative coordinates, which are actually recorded by $\mathbf{E}_{0,i}$.

$$\begin{pmatrix} x'_i \\ y'_i \\ a'_i \\ b'_i \end{pmatrix} = \begin{pmatrix} \frac{x_i - \overline{\mathbf{P}}_{0,j}.x}{\overline{\mathbf{P}}_{0,j}.w} + \frac{1}{2} \\ \frac{y_i - \overline{\mathbf{P}}_{0,j}.y}{\overline{\mathbf{P}}_{0,j}.h} + \frac{1}{2} \\ \frac{a_i}{\overline{\mathbf{P}}_{0,j}.w} \\ \frac{b_i}{\overline{\mathbf{P}}_{0,j}.h} \end{pmatrix} \tag{4}$$

Levels 1–5 of the Parametric Representation To depict the order by which several constructive elements compose one piece of calligraphic artwork in a level-by-level method, we also introduce the topological constructor in our hierarchic representation. Each of the five levels (1-st level to 5-th level) in the representation has its individual topological constructor. All the topological constructors in one complete piece of calligraphic artwork are also managed in five levels, namely the primitive stroke level, the compound stroke level, the radical level, the single character level and the whole calligraphy level. All these form a topological tree.

Recall at Section 2.1, the i-th element on the k-th level in the hierarchy is denoted as $\mathbf{P}_{k,i}$, and $\mathbf{T}_{k,i}$ is the topological constructor associated with $\mathbf{P}_{k,i}$. $\mathbf{T}_{k,i}$ carries the topological constructive relationship to compose element $\mathbf{P}_{k,i}$ based on $\mathbf{P}_{k-1,1+l_{k,i}}$, $\mathbf{P}_{k-1,2+l_{k,i}}$, $\cdots$, $\mathbf{P}_{k-1,|\mathbf{P}_{k,i}|+l_{k,i}}$, where $l_{k,i} = \sum_{s=1}^{i-1} |\mathbf{P}_{k,s}|$. And we can derive the topological constructor $\mathbf{T}_{k,i}$ using matrix (5).

$$\begin{cases} \mathbf{T}_{k,i} \triangleq (\mathbf{TCR}_{k,i}, \mathbf{TCS}_{k,i}) \\ \mathbf{TCR}_{k,i} \triangleq \begin{pmatrix} \mathbf{TR}_{k,1+l_{k,i}} \\ \mathbf{TR}_{k,2+l_{k,i}} \\ \vdots \\ \mathbf{TR}_{k,|\mathbf{P}_{k,i}|+l_{k,i}} \end{pmatrix} \\ \mathbf{TCS}_{k,i} \triangleq \left(\mathbf{TS}_{k,1+l_{k,i}}, \mathbf{TS}_{k,2+l_{k,i}}, \ldots, \mathbf{TS}_{k,|\mathbf{P}_{k,i}|+l_{k,i}}\right) \end{cases} \tag{5}$$

In (5), $l_{k,i} \triangleq \sum_{s=1}^{i-1} |\mathbf{P}_{k,s}|$; $\mathbf{P}_{k,s} \in \mathbf{F}_k; k = 1, \cdots, 5$ and $\mathbf{TCR}_{k,i}, \mathbf{TCS}_{k,i}$ are the scale and transition transformation components of the topological constructor $\mathbf{T}_{k,i}$. The definitions for matrices $\mathbf{TR}_{k,z}, \mathbf{TS}_{k,z}$, which are the elements of $\mathbf{TCR}_{k,i}, \mathbf{TCS}_{k,i}$ are as (6), where $z = 1 + l_{k,i}, 2 + l_{k,i}, \cdots, |\mathbf{P}_{k,i}| + l_{k,i}$, $\mathbf{I}$ is a unit matrix and $\mathbf{0}$ is a full zero matrix.

$$\begin{cases} \mathbf{TR}_{k,z} \triangleq \begin{cases} \begin{pmatrix} \frac{\overline{\mathbf{P}}_{k,i}.w}{\overline{\mathbf{P}}_{k-1,z}.w} & 0 \\ 0 & \frac{\overline{\mathbf{P}}_{k,i}.h}{\overline{\mathbf{P}}_{k-1,z}.h} \end{pmatrix} & (k = 2, 3, 4) \\ \mathbf{I}_{2\times 2} & (k = 1) \end{cases} \\ \mathbf{TS}_{k,z} \triangleq \begin{cases} \begin{pmatrix} \frac{\overline{\mathbf{P}}_{k-1,z}.x - \frac{\overline{\mathbf{P}}_{k-1,z}.w}{2} - \overline{\mathbf{P}}_{k,i}.x}{\overline{\mathbf{P}}_{k,i}.w} + \frac{1}{2} \\ \frac{\overline{\mathbf{P}}_{k-1,z}.y - \frac{\overline{\mathbf{P}}_{k-1,z}.h}{2} - \overline{\mathbf{P}}_{k,i}.y}{\overline{\mathbf{P}}_{k,i}.h} + \frac{1}{2} \end{pmatrix} & (k = 2, 3, 4) \\ \mathbf{0}_{2\times 1} & (k = 1) \end{cases} \end{cases} \tag{6}$$

With topological constructors of the calligraphic artwork, a one-to-one mapping between points at different levels in its hierarchical representation can be established. That is, any point $[x_{k,i}, y_{k,i}]$ in the image space taken up by $\mathbf{P}_{k,i}$ is uniquely mapped to the point $[x_{l,t}, y_{l,t}]$ in the image space taken up by $\mathbf{P}_{l,t}$. Without lost of generality, we assume $l > k$. According to our hierarchical representation, for any $[x_{k,i}, y_{k,i}]$ there must exist such a chain: $\mathbf{P}_{k,i} \in \mathbf{P}_{k+1,m_1} \in \cdots \in \mathbf{P}_{k+(l-k-1),m_{l-k-1}} \in \mathbf{P}_{l,t}$, where $\mathbf{P}_{k+j,m_j} \in \mathbf{F}_{k+j}$ $(j = 1, 2, \cdots, l-k-1)$.

For any point $[x_{m,n}, y_{m,n}]$ on the m-th level in the hierarchy, which falls within the image space taken up by $\mathbf{P}_{m,n}$, we can use the matrix operator $\nabla^b_{m,n}$ to find its correspondent point $\nabla^b_{m,n}([x_{m,n}, y_{m,n}])$ on the $(m+1)$-th level in the hierarchy:

$$\nabla^b_{m,n}([x_{m,n}, y_{m,n}]) \triangleq (\mathbf{TR}_{m+1,n}[x_{m,n}, y_{m,n}]^T + \mathbf{TS}_{m+1,n})^T. \tag{7}$$

We can also find the correspondent point $[x_{l,t}, y_{l,t}]$ on the l-th level in the hierarchical representation for any point $[x_{k,s}, y_{k,s}]$, which is on the k-th level in the hierarchy and falls within the image space taken up by $\mathbf{P}_{k,s}$ by applying the above relationship iteratively as (8).

$$[x_{l,t}, y_{l,t}] = \nabla^b_{l-1,m_{l-k-1}}\Big(\cdots\Big(\nabla^b_{k+1,m_1}\big(\nabla^b_{k,s}([x_{k,s}, y_{k,s}])\big)\Big)\Big) \tag{8}$$

We introduce the matrix operator ∇^c_n which can generate a $f \times \sum_{l=1}^{n} d_l$ dimensional matrix $\mathbf{M} = (m_{i,j})_{f \times \sum_{l=1}^{n} d_l}$ by concatenating n input matrices $\mathbf{M}_l = (m_{l,i,j})_{f \times d_l}$, which is individually a $m \times d_l$ $(l = 1, 2, \cdots, n)$ dimensional matrix. That is, we can denote $\mathbf{M} \triangleq \nabla^c_n(\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_n)$ iff (9) holds. In (9.1), $\sum_{l=1}^{z} d_l < j \leq \sum_{l=1}^{z+1} d_l, z = 1, 2, \cdots, n-1$. In (9.2), $j \leq d_1$.

$$m_{i,j} = \begin{cases} m_{z+1,i,j-\sum_{l=1}^{z} d_l} & (9.1) \\ m_{1,i,j} & (9.2) \end{cases} \tag{9}$$

A slight variation of ∇^c_n leads to a new operator ∇^o_n, defined in (10). ∇^o_n concatenates some matrices and transposes the resultant matrix. Based on the definition of ∇^c_n, we further define the matrix operator ∇^d_n as (11), which concatenates n copies of the input matrices.

$$\nabla^o_n(\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_n) \triangleq \Big(\nabla^c_n(\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_n)\Big)^T \tag{10}$$

$$\nabla^d_n(\mathbf{A}) \triangleq \nabla^c_n(\underbrace{\mathbf{A}, \mathbf{A}, \cdots, \mathbf{A}}_{n \text{ matrices } \mathbf{A}\text{s}}) \tag{11}$$

Once again, based on ∇^d_n a new matrix operator $\nabla^e_{n,m}$ is defined as (12), where $col(\mathbf{E}_{n,m})$ is the number of columns in matrix $\mathbf{E}_{n,m}$ and $\mathbf{0}_{2\times 2}$ is a 2×2 dimensional full zero matrix. $\nabla^e_{n,m}$ converts the matrix form parametric representation $\mathbf{E}_{n,m}$ for the constructive element $\mathbf{P}_{n,m}$ into its correspondent part in the matrix form parametric representation $\mathbf{E}_{n+1,l}$ for the constructive element $\mathbf{P}_{n+1,l}$, in which $\mathbf{P}_{n,m} \in \mathbf{P}_{n+1,l}$.

$$\begin{aligned} &\nabla^e_{n,m}(\mathbf{E}_{n,m}) \triangleq \\ &\nabla^c_2\Big(\nabla^o_2(\mathbf{TR}_{n,m}, \mathbf{0}_{2\times 2}), \nabla^o_2(\mathbf{0}_{2\times 2}, \mathbf{TR}_{n,m})\Big)\mathbf{E}_{n,m} + \\ &\nabla^d_{col(\mathbf{E}_{n,m})}\Big(\nabla^o_2(\mathbf{TS}^T_{n,m}, \mathbf{0}_{1\times 2})\Big) \end{aligned} \tag{12}$$

Now, we can derive the formal definition for the hierarchical and parametric representation for calligraphic artwork as (13), in which $l_{k,i} = \sum_{s=1}^{i-1} |\mathbf{P}_{k,s}|$.

$$\begin{cases} \mathbf{E}_{k,i} = (x'_i, y'_i, a'_i, b'_i)^T \ (\mathbf{P}_{0,i} \in \mathbf{F}_0) \ (k = 0) \\ \mathbf{E}_{k,i} = \nabla^c_{|\mathbf{P}_{k,i}|}\Big(\nabla^e_{k-1,1+l_{k,i}}(\mathbf{E}_{k-1,1+l_{k,i}}), \\ \qquad \nabla^e_{k-1,2+l_{k,i}}(\mathbf{E}_{k-1,2+l_{k,i}}), \ldots, \\ \qquad \nabla^e_{k-1,|\mathbf{P}_{k,i}|+l_{k,i}}(\mathbf{E}_{k-1,|\mathbf{P}_{k,i}|+l_{k,i}})\Big) \ (k = 1, \cdots, 5) \end{cases} \tag{13}$$

3 Facsimiling Existent Calligraphy

This is the process in which the hierarchical parametric representations are extracted from input images of existent calligraphic artwork. The reason why we choose to process this kind of input rather than using tablet input devices is that many famous calligraphists in history only left their handwriting as static images. Obviously, it is very easy to tailor our system to process parametric calligraphy directly sampled by tablet pen as input data.

3.1 Extracting Levels 0–1 Elements

To extract the ellipses from the input image, we first compute the skeleton of the calligraphy. This is the "character skeletonization" problem. The target is to extract a skeleton that is a close approximation to the actual trajectory of the brush when the calligraphy was created. Several existing papers discuss various approaches to automatic skeletonizing binary images of characters [2, 22, 34]. For the specific problem of handwritten Chinese character skeletonization, many approaches have also been proposed [3, 4, 8–10, 13, 14, 32].

In our approach, we employed the algorithm in [26] to extract skeletons of strokes from input images of Chinese characters. The strokes of a character are extracted first and then the isolated strokes are skeletonized. The algorithm can work effectively on characters written in most styles. However, for those largely distorted calligraphic styles, it will tend to commit mistakes. These mistakes occur during stroke segmentation, where multiple strokes could be mistaken to be of the same stroke or a single stroke segmented into multiple strokes. This is a difficult problem to tackle, and an important area for future research.

Once the skeleton of a primitive stroke is identified, each pixel on this discrete curve is taken as the center of an ellipse, and the maximum ellipse within the stroke area is computed. It is easy to compute all the constructive ellipses using the Bresenham ellipse rasterization algorithm. In our approach, all the constructive ellipses do not have rotational freedom, i.e. their major axes must be either horizontal or vertical.

We then determine the syntax of the identified stroke. This is through comparing the shape of the stroke to the shapes of the five standard primitive strokes (Figure 3(a)). The identified primitive stroke's syntax is recognized as one of the

five standard primitive strokes with which the mutual shape similarity is maximum. The similarity between two 2D shapes **a** and **b** is defined to be the maximum overlapping between **a** and **b**, i.e, $Similarity(\mathsf{a}, \mathbf{b}) \triangleq max\{Over(\mathsf{a}, \mathbf{b})\}$ with the condition that shape **a** can be arbitrarily rotated and scaled. The overlapping between **a** and **b** is defined to be $Over(\mathsf{a}, \mathbf{b}) \triangleq (\mathbf{a} \cap \mathbf{b})/(\mathbf{a} \cup \mathbf{b})$, where $\mathbf{a} \cap \mathbf{b}$ and $\mathbf{a} \cup \mathbf{b}$ are respectively the intersection and union of the image spaces taken up by **a** and **b** separately.

3.2 Extracting Levels 2–3 Elements

Based on the identified primitive strokes, we can use the spatial relation between them to compose constructive elements at higher levels through shape grammar productions. We can therefore extract constructive elements on levels 2–3. The syntax of any constructive element produced this way can be easily determined since each shape grammar production is associated with a certain syntax. Inspired by [11], the idea of fuzzy set is used to increase the reliability of the extraction process.

The shape grammar production for the compound stroke $\mathbf{CS}_1$ in the first column and first row of Figure 3(b) is:

– IF {**a** is a horizontal primitive stroke} AND {**b** is a vertical primitive stroke} AND {**a** is on top of **b**} AND {**a** is on the left side of **b**} AND {**a** touches **b**} THEN {**a**,**b** should be combined to form the compound stroke $\mathbf{CS}_1$.}

And the shape grammar production for the radical $\mathbf{R}_{36}$ in the last column and last row of Figure 3(c) is:

– IF {**a** is a degraded compound stroke, which resembles a vertical primitive stroke} AND {**b** is a degraded compound stroke, which resembles a vertical primitive stroke} AND {**c** is a compound stroke of the kind $\mathbf{S}_2$} AND {**a** is on the left of **c**} AND {**a** is on the left of **b**} AND {**b** crosses **c**} AND {**a** touches **c**}
THEN {**a**,**b**,**c**, should be combined to form the radical $\mathbf{R}_{36}$.}

$\mathbf{S}_2$ refers to the compound stroke in the first row and the second column of Figure 3(b). The speaking of "a degraded compound stroke resembling a vertical typed primitive stroke" is introduced at Section 2.1. Following [11], during shape grammar production deduction, each statement in the production is associated a confidence value. The overall confidence of the shape grammar production can be derived by the confidence of all its statements. Only the shape grammar production that yields the highest confidence will be applied.

The above processes of extracting compound strokes and radicals through shape grammar productions are not always correct when the calligraphy is cursive. Thus, during extraction of constructive elements on levels 2–3, direct user interaction is allowed through a friendly GUI. Due to the space limitation, we omit the discussion about this GUI.

3.3 Extracting Level 4 Elements

To extract constructive elements on level 4, we need to determine which radicals belong to the same character, and whether the radicals are degraded or not. This is the well-known problem of "character segmentation" in pattern recognition research. In our system, we use projection analysis to account for possible slanting of characters in order to segment the characters in a calligraphy piece, like what is done in [1]. More accurate and sophisticated character segmentation methods are introduced in [5], which could be incorporated into future versions of our system.

4 Generating New Calligraphy

4.1 Principle of New Calligraphy Generation

As early as in 1968, Evan [31] proposed a paradigm for solving geometric analogy intelligence test questions. In 1975, Simon pointed out that design and creation is a class of problems featured by their synthesis nature [6]. In early 1980s, Winston published his pioneering results on the relationship between learning, reasoning and analogy [24, 25]. Other fundamental work on learning by analogy includes [12] and [19]. Holyoak concluded that analogical thinking is an important feature of human intelligence [16]. Keane [20, 21] applied analogical mechanisms to problem solving. Our approach is also based on analogical reasoning. We devised a *calligraphy creation component* by simulating the **ARP** using a computational approach.

Suppose that the **SARP** is applied to the k-th level in the hierarchical representation of calligraphic artwork. In the reasoning, there are n constructive elements $\mathbf{P}_{k,l_1}, \cdots, \mathbf{P}_{k,l_n}$ already learned by the computer, which are organized and stored in a small structural stroke database and activated as source knowledge for the **SARP**. Recall each element has four fields (Section 2.2). We denote the analogous reasoning intensity used against the s-th field of the i-th source knowledge ($\mathbf{P}_{k,l_i}$) during the **SARP** as $\omega_{l_i}^s$, where s's range is $1, \cdots, 4$. All the analogous reasoning intensities, $\omega_{l_i}^s$ $(s = 1, \cdots, 4)$, together form the "viewpoint sequence" of the **SARP**: $\overline{\omega} = \{\omega_{l_i}^s | i = 1, \cdots, n; s = 1, \cdots, 4\}$. We denote the result of the **SARP** as $\mathbf{P}_{k,r}$ with its matrix form parametric representation being $\mathbf{E}_{k,r}$. Then the general mathematic principle we adopted in the **SARP** can be stated as (14), where $\mathbf{E}_{k,l_i}$ is the matrix form parametric representation of the constructive element $\mathbf{P}_{k,l_i}$. With (14), we can generate a new constructive element $\mathbf{P}_{k,r}$ based on all the machine-learned samples, $\mathbf{P}_{k,l_i}$ $(i = 1, \cdots, n)$, and the viewpoint sequence $\overline{\omega}$ through our **SARP**. Note that (14) is not a strict mathematic equation. It is only a sketch showing the principle we adopted to generate new calligraphy through the **SARP**. Section 4.2 discusses in more details the principle.

$$\mathbf{E}_{k,r} = \sum_{i=1}^{n} \sum_{s=1}^{4} \omega_{l_i}^s \mathbf{E}_{k,l_i} \tag{14}$$

Our **SARP** is essentially either an interpolation or an extrapolation process. That is, $\sum_{i=1}^{n} \omega_{l_i}^{s} = 1$ $(i = 1, \cdots, n; s = 1, \cdots, 4)$. In our intelligent calligraphy generation system, all the analogous reasoning intensities can be input and adjusted by the user manually through a graphical interface; the computer would perform auto-normalization to scale the sum of all the input reasoning intensities to 1. Our system is also equipped with a component to generate random numbers to be used as reasoning intensities, and to filter out those "ugly looking" calligraphic outputs via a constraint satisfying procedure. Section 5 has more details on this component.

4.2 New Calligraphy Generation System

Generating new Constructive Elements To carry out the **SARP**, we need to equalize the dimensions of the reasoning sources. That is, if $\mathbf{P}_{k,s}$ and $\mathbf{P}_{k,t}$ are reasoning sources of **SARP**, their matrix representations $(\mathbf{E}_{k,s})_{4\times n_1}$ and $(\mathbf{E}_{k,t})_{4\times n_2}$ must be such that $n_1 = n_2$. To make the **SARP** also possible even when the dimensions of $\mathbf{P}_{k,s}$ and $\mathbf{P}_{k,t}$ are different, we introduce an equalization operator, ∇_t^g, to convert a matrix with any number of columns into a new matrix with t columns based on "key columns" in the original matrix.

Assume that $\mathbf{P}_{k,l_i}$ is a reasoning source in the **SARP**. We first derive a discrete planar curve composed of the centers of all the constructive ellipses that $\mathbf{P}_{k,l_i}$ contains by (15), denoted $\mathbf{C}_{k,l_i}$. In (15), $(\mathbf{e}_{n,i})_{n\times 1} \triangleq (\sigma(i,1), \cdots, \sigma(i,n))^T$, where $\sigma(i,j) = 1$ if $i = j$, otherwise $\sigma(i,j) = 0$ and $co = col(\mathbf{E}_{k,l_s})$.

$$\begin{aligned}\mathbf{C}_{k,l_s} &= \nabla_2^o\Big((\mathbf{e}_{4,1}^T \times \mathbf{E}_{k,l_s})^T, (\mathbf{e}_{4,2}^T \times \mathbf{E}_{k,l_s})^T\Big) \\ &= \begin{pmatrix} x_1 & x_2 & \cdots & x_{co} \\ y_1 & y_2 & \cdots & y_{co} \end{pmatrix}\end{aligned} \tag{15}$$

If the curve has $v+1$ key points, with their occurrences in the curve being the sequence u_0, u_1, $\cdots$, u_v, the "key columns" in the matrix $\mathbf{E}_{k,l_i}$ are selected as the u_0, u_1, $\cdots$, u_v-th columns. That is, if the curve $\mathbf{C}_{k,l_s}$ has $v+1$ key points, with their individual coordinates as $\mathbf{C}_{k,l_s}\mathbf{e}_{co,u_0}$, $\mathbf{C}_{k,l_s}\mathbf{e}_{co,u_1}$, $\cdots$, $\mathbf{C}_{k,l_s}\mathbf{e}_{co,u_v}$, the key columns in the matrix $\mathbf{E}_{k,l_s}$ are selected as $\mathbf{E}_{k,l_s}\mathbf{e}_{co,u_0}$, $\mathbf{E}_{k,l_s}\mathbf{e}_{co,u_1}$, $\cdots$, $\mathbf{E}_{k,l_s}\mathbf{e}_{co,u_v}$. We use the algorithm in [23] to extract key points on the planar curve. With the key columns of the matrices for all the reasoning sources, a correspondence between related pieces of knowledge can be set up.

Suppose $\mathbf{E}_{k,l_s}$ is the matrix representation of a certain analogy source $\mathbf{P}_{k,l_s}$ with $v+1$ key columns extracted. These key columns are the u_0th, u_1th, $\cdots$, u_vth columns in the matrix $(1 = u_0 < u_1 < \cdots < u_v = col(\mathbf{E}_{k,l_s}), s = 1, 2, \cdots, n)$. Then, a matrix operator ∇_t^g can be defined as (16), which converts one matrix into a matrix having t columns. In (16), $\theta = \lceil u_j + \frac{u_{j+1}-u_j}{\lceil\frac{t\times(j+1)}{v}\rceil - \lceil\frac{t\times j}{v}\rceil} \times (i - \lceil\frac{t\times j}{v}\rceil)\rceil$; $\lceil\frac{t\times j}{v}\rceil < i \leq \lceil\frac{t\times(j+1)}{v}\rceil$; $j \in \{0, 1, \cdots, v-1\}$; $s = 1, 2, \cdots, n$; $\lceil\cdot\rceil$ is a floor function. In particular, if each column in matrix $\mathbf{E}_{k,l_s}$ is selected as the key column, operator ∇_t^g can be simplified into (17).

$$\big(\nabla_t^g(\mathbf{E}_{k,l_s})\big)\mathbf{e}_{t,i} \triangleq \mathbf{E}_{k,l_s}\mathbf{e}_{col(\mathbf{E}_{k,l_s}),\theta} \ (i = 1, 2, \cdots, t) \tag{16}$$

$$\big(\nabla_t^g(\mathbf{E}_{k,l_s})\big)\mathbf{e}_{t,i} \triangleq \mathbf{E}_{k,l_s}\mathbf{e}_{col(\mathbf{E}_{k,l_s}),\lceil\frac{i\times col(\mathbf{E}_{k,l_s})}{t}\rceil}(i=1,2,\cdots,t) \tag{17}$$

In the **SARP**, we assume the shape of a constructive element written in the font style "Kai" (GB2312) as used in recent versions of Microsoft Word to be the standard shape of the element. For each reasoning source $\mathbf{P}_{k,l_i}$ in the **SARP**, we denote its associated standard constructive element as $\mathbf{P}_{k,l_i}^{std}$ and its matrix form parametric representation as $\mathbf{E}_{k,l_i}^{std}$. We then compute the distance $\mathbf{E}_{k,l_i}^{f}$ by which the shape of $\mathbf{P}_{k,l_i}$ differs from that of $\mathbf{P}_{k,l_i}^{std}$, as expressed in (18). $\mathbf{E}_{k,l_i}^{f}$ is used as the feature of $\mathbf{P}_{k,l_i}$ in the **SARP**.

$$\mathbf{E}_{k,l_i}^{f} \triangleq \mathbf{E}_{k,l_i} \ominus \mathbf{E}_{k,l_i}^{std} \tag{18}$$

Based on the operator of ∇_t^g, we can define an active analogy source reaction operator ∇_n^f as (19). In (19), $h = max\{col(\mathbf{M}_i)|i=1,2,\cdots,n\}$.

$$\nabla_n^f(\mathbf{M}_1,\mathbf{M}_2,\cdots,\mathbf{M}_n) \triangleq \nabla_n^c\big(\nabla_h^g(\mathbf{M}_1),\nabla_h^g(\mathbf{M}_2),\cdots,\nabla_h^g(\mathbf{M}_n)\big) \tag{19}$$

Applying the operator ∇_n^f can derive a feature matrix $\mathbf{E}_{k,src}^{f}$ relating to all the activated analogous reasoning sources as (20), where $h = max\{col(\mathbf{E}_{k,l_i})|i = 1,2,\cdots,n\}$. (20) is a detailed version of (18), which gives the implementation of the operator $\ominus$.

$$\mathbf{E}_{k,src}^{f} = \mathbf{E}_{k,src} \ominus \mathbf{E}_{k,std} \triangleq \mathbf{E}_{k,src} - \nabla_n^f(\mathbf{E}_{k,l_1}^{std},\cdots,\mathbf{E}_{k,l_n}^{std}) \tag{20}$$

Recall that all the reasoning intensities in **SARP** are organized into the viewpoint sequence $\overline{\omega}$ (Section 4.1). We simulate **ARP** as an interpolation/extrapolation process. To derive $\mathbf{P}_{k,r}^{f}$, the feature of the reasoning result from the **SARP**, we take the reasoning intensity $\omega_{l_i}^s$ against the s-th field of the i-th reasoning source $\mathbf{P}_{k,l_i}$ as the weight for the s-th row of the feature matrix $\mathbf{E}_{k,l_i}^{f}$ of $\mathbf{P}_{k,l_i}$ in an interpolation/extrapolation process ($s=1,\cdots,4;\ i=1,\cdots,n$). This means that $\mathbf{E}_{k,l_i}^{f}$s are the entities that are actually interpolated/extrapolated. The interpolation/extrapolation process we employ to simulate **ARP** is in the form of (21).

$$\mathbf{E}_{k,r}^{f} = \oslash(\mathbf{E}_{k,l_1}^{f},\cdots,\mathbf{E}_{k,l_n}^{f},\overline{\omega}) \tag{21}$$

where $\mathbf{E}_{k,r}^{f}$ is the matrix form parametric representation of $\mathbf{P}_{k,r}^{f}$; $\oslash$ is the analogous reasoning mechanism simulation operator, which is currently implemented as an interpolation/extrapolation process in our prototype system. (24) depicts the specific interpolation/extrapolation strategy we employed, to be explained later.

According to a user specified reasoning intensities $\omega_{l_i}^s$ for each reasoning source in the hierarchy, we can derive an analogous reasoning viewpoint matrix $\mathbf{W}^s$ acting on the s-th fields of all the activated analogous reasoning sources in **SARP** as (22), where $h = max\{col(\mathbf{E}_{k,l_i})|i=1,2,\cdots,n\}$ and $\mathbf{I}_{h\times h}$ is a $h\times h$ dimensional unit matrix, $s=1,\cdots,4$.

$$\mathbf{W}^s \triangleq \nabla_n^o(\omega_{l_1}^s\times\mathbf{I}_{h\times h},\omega_{l_2}^s\times\mathbf{I}_{h\times h},\cdots,\omega_{l_n}^s\times\mathbf{I}_{h\times h}) \tag{22}$$

Now, we can get the reasoning feature result $\mathbf{E}_{k,r}^{f}$ from **SARP** as (23). (23) is a detailed version of (21), which gives the implementation of the operator $\oslash$.

$$\begin{aligned}\mathbf{E}_{k,r}^{f} = \oslash(\mathbf{E}_{k,l_1}^{f}, \cdots, \mathbf{E}_{k,l_n}^{f}, \overline{\omega}) \triangleq \\ \left(\nabla_4^c\Big((\mathbf{E}_{k,src}^{f} \otimes \mathbf{W}^1)^T\mathbf{e}_{4,1}, \cdots, (\mathbf{E}_{k,src}^{f} \otimes \mathbf{W}^4)^T\mathbf{e}_{4,4}\Big)\right)^T\end{aligned} \tag{23}$$

In (23), $\otimes$ is the analogous reasoning mechanism simulation operator, defined at (24). In (24), $c_{i,j}$ is the element in the i-th row and j-th column of the matrix $\mathbf{C}_{p\times r}$. If the **SARP** is linear, $\otimes$ is defined as (24.1). In (24.1), $\bullet$ is the ordinary matrix multiplication operator. If the reasoning process is z-degree polynomial, $\otimes$ is defined as (24.2). If the process is non-polynomial, $\otimes$ can be defined as (24.3).

$$\mathbf{C}_{p\times r} = \mathbf{A}_{p\times q} \otimes \mathbf{B}_{q,r} \triangleq \begin{cases} \mathbf{A}_{p\times q} \bullet \mathbf{B}_{q,r} & (24.1) \\ c_{i,j} = \sqrt[z]{\sum_{k=1}^{q}(a_{i,k} \times b_{k,j})^z} & (24.2) \\ c_{i,j} = \sqrt[q]{\prod_{k=1}^{q}(a_{i,k} \times b_{k,j})} & (24.3) \end{cases} \tag{24}$$

If all the intensities of reasoning sources fall within $(0, 1)$, namely $0 \leq \omega_{l_i}^{s} \leq 1$ $(s = 1, \cdots, 4;\ i = 1, \cdots, n)$, the **ARP** is simulated using an interpolation process; otherwise it is simulated using an extrapolation process. From a psychological point of view, if $\exists\omega_{l_i}^{s} < 0$, the **SARP** reflects the inverse reasoning of brain activity; if $\exists\omega_{l_i}^{s} > 1$, the **SARP** represents positive exaggeration of brain activity; and if $n \geq 3$, **SARP** mimics combined thinking activity.

Finally by adding back the shape of $\mathbf{P}_{k,r}^{std}$, the standard constructive element associated with the feature $\mathbf{P}_{k,r}^{f}$ of the reasoning result $\mathbf{P}_{k,r}$ in the **SARP**, we obtain the parametric representation $\mathbf{E}_{k,r}$ of $\mathbf{P}_{k,r}$ as indicated by (25), where $\mathbf{E}_{k,r}^{std}$ is the matrix form parametric representation of the shape of $\mathbf{P}_{k,r}^{std}$.

$$\mathbf{E}_{k,r} = \mathbf{E}_{k,r}^{f} \oplus \mathbf{E}_{k,r}^{std} \tag{25}$$

(26) is a detailed version of (25), which gives the implementation of the operator $\oplus$. With $\oplus$, the resultant knowledge (constructive elements in calligraphy) from the **SARP** can be derived. In (26), $h = max\{col(\mathbf{E}_{k,l_i})|i = 1, 2, \cdots, n\}$.

$$\mathbf{E}_{k,r} = \mathbf{E}_{k,r}^{f} \oplus \mathbf{E}_{k,r}^{std} \triangleq \mathbf{E}_{k,r}^{f} + \nabla_h^g\mathbf{E}_{k,r}^{std} \tag{26}$$

Generating new Topological Constructor Note that the **SARP** can be applied not only on the matrix representations, $\mathbf{E}_{k,l_1}, \cdots, \mathbf{E}_{k,l_n}$, of all the reasoning sources, $\mathbf{P}_{k,l_1}, \cdots, \mathbf{P}_{k,l_n}$, by evaluating a series of matrix operations simulating the **ARP**, but can also be applied to the topological constructors of all the reasoning sources, $\mathbf{T}_{k,l_1}, \cdots, \mathbf{T}_{k,l_n}$. If the corresponding intensities of $\mathbf{T}_{k,l_1}, \cdots, \mathbf{T}_{k,l_n}$ are $\omega_{l_1}, \cdots, \omega_{l_n}$, where $\sum_{i=1}^{n}\omega_{l_i} = 1$, the newly generated topological constructor $\mathbf{T}_{k,r}$ can be given as: $\mathbf{T}_{k,r} \triangleq \odot(\mathbf{T}_{k,l_1}, \cdots, \mathbf{T}_{k,l_n}, \omega_{l_1}, \cdots, \omega_{l_n})$. Here $\odot$ is the **ARP** simulation operator for topological constructors. Similarly, we can overload the definition of the operator $\odot$ to simulate different types of creative

thinking activities; some simple ones are: arithmetic mean, geometric mean and harmonic mean. The strict definition about the analogous reasoning simulation operator for topological constructors $\odot$ are as (27).

$$\mathbf{T}_{k,r} \triangleq \odot(\mathbf{T}_{k,l_1}, \cdots, \mathbf{T}_{k,l_n}, \omega_{l_1}, \cdots, \omega_{l_n}) \triangleq \begin{cases} \sum_{i=1}^{n}(\mathbf{T}_{k,l_i} \times \omega_{l_i}) & \textbf{Arithmetic Mean} \\ \prod_{i=1}^{n}(\mathbf{T}_{k,l_i}^{\omega_{l_i}}) & \textbf{Geometric Mean} \\ \left(\sum_{i=1}^{n}(\frac{\omega_{l_i}}{\mathbf{T}_{k,l_i}})\right)^{-1} & \textbf{Harmonic Mean} \end{cases} \tag{27}$$

5 Generating Artistic Calligraphy

5.1 Constraints on the Process

There are four constraints that are useful: $\mathbf{Con}_1$ is a rigid constraint, and $\mathbf{Con}_2$, $\mathbf{Con}_3$, and $\mathbf{Con}_4$ are soft constraints.

$\mathbf{Con}_1$ says that the source knowledge that is being reasoned must be homogeneous in terms of its compositive constructive elements; that is, they must be composed of the same number of sub-constructive elements from one level down. $\mathbf{Con}_2$ and $\mathbf{Con}_3$ suggest that all the reasoning sources in the **SARP**, namely parameterized constructive elements from existent calligraphy, must have similar syntax. $\mathbf{Con}_2$ requires all the constructive elements used in the reasoning process must be in the same level; specifically, when we apply reasoning on $\mathbf{P}_{m,s}$ and $\mathbf{P}_{n,t}$, we must guarantee $m = n$. $\mathbf{Con}_3$ dictates that the constructive elements being reasoned on should have similar properties. For instance, if we are reasoning at the level of "primitive strokes", the elements involved must be of the same type which is one of the five possible types in level 1: a horizontal stroke, a vertical stroke, a left slanting stroke, a right slanting stroke or a point stroke, which are illustrated at Figure 3(a). $\mathbf{Con}_4$ demands that the structure of the newly generated calligraphy should not go beyond the maximum and minimum tolerable constraint extracted from all the samples learned. Section 5.3 discusses the details of maximum and minimum tolerable constraints.

5.2 Extracting Aesthetic Constraints from Existent Artwork

Interference between Constructive Elements Needed for a quantifiable constraint on aesthetics is the concept of the *degree of interference* between two constructive elements, which indicates the spatial inter-relationship between the elements. These degrees of interference supervise the process of generating an upper-level constructive element from several lower-level ones.

We denote the bounding boxes of two constructive elements $\mathbf{a}$ and $\mathbf{b}$ as $\overline{\mathbf{a}}$ and $\overline{\mathbf{b}}$. There are three kinds of degrees of interference possible between $\overline{\mathbf{a}}$ and $\overline{\mathbf{b}}$, as given in (28). $\vartheta_x(\overline{\mathbf{a}}, \overline{\mathbf{b}})$ is the x dimensional degree of interference; $\vartheta_y(\overline{\mathbf{a}}, \overline{\mathbf{b}})$ the y dimensional degree of interference; and $\vartheta_s(\overline{\mathbf{a}}, \overline{\mathbf{b}})$ the shaping degree of interference. In (28.3), $I(\mathbf{a})$ and $I(\mathbf{b})$ are the image spaces that the constructive elements $\mathbf{a}$ and $\mathbf{b}$ take up; $I(\mathbf{a}) \cap I(\mathbf{b})$ and $I(\mathbf{a}) \cup I(\mathbf{b})$ are the intersection

and union of these two image spaces, respectively. With (28), the x, y directional spatial relativity between $\mathbf{a}$ and $\mathbf{b}$ can be compactly represented, and the shaping degree of interference $\vartheta_s(\overline{\mathbf{a}}, \overline{\mathbf{b}})$ can depict the degree of overlapping of the two constructive elements $\mathbf{a}$ and $\mathbf{b}$.

$$\begin{cases} \vartheta_x(\overline{\mathbf{a}}, \overline{\mathbf{b}}) \triangleq (\overline{\mathbf{a}}.x - \overline{\mathbf{b}}.x)/(\overline{\mathbf{a}}.w + \overline{\mathbf{b}}.w) & (28.1) \\ \vartheta_y(\overline{\mathbf{a}}, \overline{\mathbf{b}}) \triangleq (\overline{\mathbf{a}}.y - \overline{\mathbf{b}}.y)/(\overline{\mathbf{a}}.h + \overline{\mathbf{b}}.h) & (28.2) \\ \vartheta_s(\overline{\mathbf{a}}, \overline{\mathbf{b}}) \triangleq (I(\mathbf{a}) \cap I(\mathbf{b}))/(I(\mathbf{a}) \cup I(\mathbf{b})) & (28.3) \end{cases} \tag{28}$$

Introducing these three kinds of the degrees of interference not only gives much convenience in describing the spatial relativity between the two constructive elements concerned quantitatively, but also helps express the spatial relativity qualitatively. Take $\vartheta_x(\overline{\mathbf{a}}, \overline{\mathbf{b}})$ for example, if $\vartheta_x(\overline{\mathbf{a}}, \overline{\mathbf{b}}) < -\frac{1}{2}$, $\overline{\mathbf{a}}$ is on the left side of $\overline{\mathbf{b}}$, not overlapping; if $\vartheta_x(\overline{\mathbf{a}}, \overline{\mathbf{b}}) = -\frac{1}{2}$, $\overline{\mathbf{a}}$ is on the left side of $\overline{\mathbf{b}}$, just overlapping; if $\vartheta_x(\overline{\mathbf{a}}, \overline{\mathbf{b}}) \in [-\frac{1}{2}, -\frac{1}{2}]$, $\overline{\mathbf{a}}$ overlaps with $\overline{\mathbf{b}}$; if $\vartheta_x(\overline{\mathbf{a}}, \overline{\mathbf{b}}) = \frac{1}{2}$, $\overline{\mathbf{a}}$ is on the right side of $\overline{\mathbf{b}}$, just overlapping; if $\vartheta_x(\overline{\mathbf{a}}, \overline{\mathbf{b}}) > \frac{1}{2}$, $\overline{\mathbf{a}}$ is on the right side of $\overline{\mathbf{b}}$, not overlapping. In the same manner, with $\vartheta_y(\overline{\mathbf{a}}, \overline{\mathbf{b}})$, the spatial relativity between $\overline{\mathbf{a}}$ and $\overline{\mathbf{b}}$ along the y dimension can be conveniently derived. $\vartheta_s(\overline{\mathbf{a}}, \overline{\mathbf{b}})$ also reveals whether the two constructive elements overlap: if $\vartheta_s(\overline{\mathbf{a}}, \overline{\mathbf{b}}) > 0$, the two constructive elements overlap.

Structure Matrix of a Constructive Element Based on the degrees of interference between two constructive elements just defined, we introduce the *structure matrix* of a constructive element. Let $\overline{\mathbf{P}}_{k+1,1}$ be the bounding box of a constructive element $\mathbf{P}_{k+1,1}$, which is composed of m constructive elements $\mathbf{P}_{k,l_1}$, $\cdots$, $\mathbf{P}_{k,l_m}$ in the next lower level. We use three matrices, $\theta_x(\mathbf{P}_{k+1,1})$, $\theta_y(\mathbf{P}_{k+1,1})$, and $\theta_s(\mathbf{P}_{k+1,1})$, to represent the structure of $\mathbf{P}_{k+1,1}$, as given in (29). The three matrices are essentially made up of the x, y dimensional and shaping degrees of interference between every pair of $\overline{\mathbf{P}}_{k,l_i}$ and $\overline{\mathbf{P}}_{k,l_j}$.

$$\begin{cases} \theta_x(\mathbf{P}_{k+1,1}) \triangleq (\theta^{i,j}_{k,x})_{m\times m};\ \theta^{i,j}_{k,x} = \vartheta_x(\overline{\mathbf{P}}_{k,l_i}, \overline{\mathbf{P}}_{k,l_j}) \\ \theta_y(\mathbf{P}_{k+1,1}) \triangleq (\theta^{i,j}_{k,y})_{m\times m};\ \theta^{i,j}_{k,y} = \vartheta_y(\overline{\mathbf{P}}_{k,l_i}, \overline{\mathbf{P}}_{k,l_j}) \\ \theta_s(\mathbf{P}_{k+1,1}) \triangleq (\theta^{i,j}_{k,s})_{m\times m};\ \theta^{i,j}_{k,s} = \vartheta_s(\overline{\mathbf{P}}_{k,l_i}, \overline{\mathbf{P}}_{k,l_j}) \\ (i = 1, \cdots, m;\ j = 1, \cdots, m) \end{cases} \tag{29}$$

5.3 Constraint Satisfaction for Calligraphy Generation

In Section 4.1, we assumed there are n knowledge sources, $\mathbf{P}_{k,l_1}$, $\cdots$, $\mathbf{P}_{k,l_n}$, in the **SARP**. With the structure matrices of these n samples computed according to (29), we can derive constraint matrices for the **SARP** needed for the generation of artistic calligraphy. Without lost of generality, we discuss how to derive two x-dimensional constraint matrices based on $\theta_x(\mathbf{P}_{k,l_1})$, $\cdots$, $\theta_x(\mathbf{P}_{k,l_n})$. Among the two constraint matrices, one is the matrix of the maximum tolerable structure θ^{max}_x and another one is the matrix of the minimum tolerable structure θ^{min}_x. Assume each of $\mathbf{P}_{k,l_1}$, $\cdots$, $\mathbf{P}_{k,l_n}$ is composed of m constructive elements of the next

lower level; then θ_x^{max} and θ_x^{min} are both $m \times m$ dimensional matrices. In θ_x^{max}, the element in the i-th row and the j-th column ($\theta_x^{max}(i,j)$) is the maximum value of all the n elements in the i-th row and the j-th column of $\theta_x(\mathbf{P}_{k,l_1})$, $\cdots$, $\theta_x(\mathbf{P}_{k,l_n})$. Similarly, $\theta_x^{min}(i,j)$, the element in the i-th row and the j-th column of θ_x^{min}, is the minimum value of all the elements in the i-th row and the j-th column of $\theta_x(\mathbf{P}_{k,l_1})$, $\cdots$, $\theta_x(\mathbf{P}_{k,l_n})$. The use of the two constraint matrices in the **SARP** is simple: during the **SARP**, each time when the system automatically generates a new constructive element $\mathbf{P}_{k,r}$, the x-dimensional structure matrix of this element is computed as $\theta_x(\mathbf{P}_{k,r})$. The system will output this newly generated constructive element only if $\theta_x(\mathbf{P}_{k,r})$ is no smaller than θ_x^{min} under the tolerance τ^{min} and no larger than θ_x^{max} under the tolerance τ^{max}. We say a matrix is larger (resp. smaller) than another matrix under a certain tolerance τ only if all of its elements are at least (resp. at most) τ times that of the corresponding elements in another matrix and these other elements are non-zero. In our experiments, we set $\tau^{max} = 0.8$ and $\tau^{min} = 1.2$. Similarly, we also derive θ_y^{min}, θ_y^{max}, θ_s^{min}, θ_s^{max} to constrain the randomly generated intensities of the **SARP** to forbid the system to output a calligraphy that violates the aesthetic constraints extracted from existent calligraphy.

5.4 Relaxing the Aesthetic Constraints

The constraints of the **SARP** can be relaxed in order to allow for results with new styles that are not so imaginable. To relax $\mathbf{Con}_1$, constructive elements that are heterogeneous can be turned into homogeneous ones by combining some sub-constructive elements together. [17] gives an optimized strategy to do the combining using fuzzy-attribute graphs. To relax $\mathbf{Con}_2$, we apply the **SARP** simultaneously to constructive elements belonging to different layers. To relax $\mathbf{Con}_3$, we apply the **SARP** to constructive elements with different syntaxes, such as reasoning between a point and a vertical stroke. To relax $\mathbf{Con}_4$, we can adjust the thresholds τ^{min} and τ^{max} when comparing the structure matrix of the newly generated constructive element against the maximum and minimum tolerance matrices.

From a computational psychology's perspective, relaxing or ignoring the constraints in our analogous reasoning process corresponds to creative brain activity of the calligraphists such as when performing cursive and running style writing. Such a loose **SARP** could well be the reflection of the thinking process of a calligraphist while creating an artwork of running style, a style which is considered to be the freest of all forms. Without the constraints or with them relaxed, there is a huge space in which reasoning could lead to plenty of fancy results. Going to the extreme with the relaxation, however, might give rise to ugly handwriting results. When such a situation arises, some human intervention to filter out the unacceptable might be necessary.

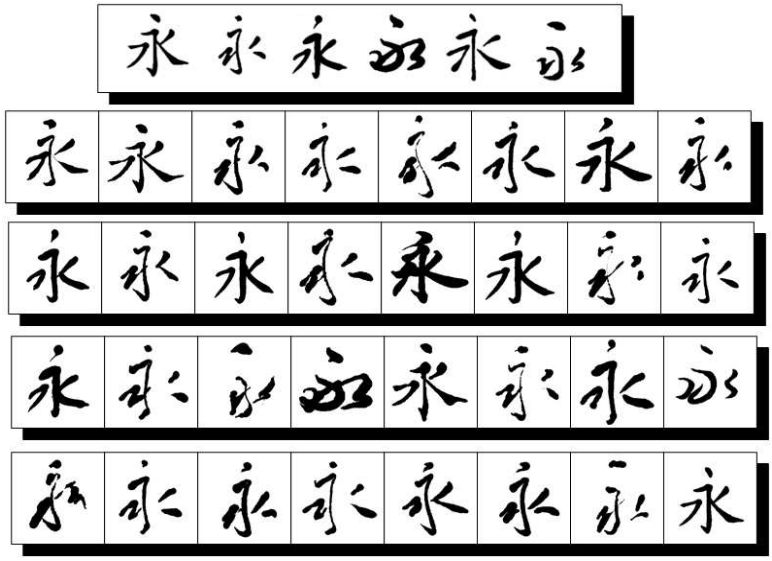


Fig. 6. A single character in many styles; the first row is the learned samples, and the other four rows are the computer generated results

6 Experimental Results

Because of space limitation, we can show a small sample of the results. Figure 6 shows the results from the **SARP** being applied to a single character using six learned samples as the reasoning source, with the analogous reasoning simulated as a linear interpolation process. Figure 7 shows the results at the single-character level using five learned samples, with the analogous reasoning simulated as a non-linear interpolation process. Figure 8 is the result of another experiment of analogous reasoning up to the single-character level. The results verify that our approach can yield different new styles, as well as consistently styled characters within the same new style.

We also analyzed the sensitivity in terms of the increase of creativity of the system when the number of learned samples was varied. Figure 9 reports the results, where the sample styles learned are the shapes of the "Kai", "Li", "Xingshu", "Weibei", "Xingkai", "Xingchao", and "Kuangchao" styles. We invited six calligraphic fans with at least more than 2 years' writing experience and four professional calligraphists to form a review committee, including a professor major at calligraphy in an art school as the reviewing committee. They cast votes on the calligraphic artwork generated by the system. If an artwork received more than seven votes, it was considered a new calligraphic work. Figure 9 clearly shows that with more learned samples, the chance of generating an acceptable calligraphy piece increases.

We would like also to show Figure 10. The calligraphy (the character "forever" in Chinese) at the top of the picture was generated fully automatically

Fig. 7. A "couple" in many styles. (a) The learned samples. (b)–(l) Some selected computer generated results

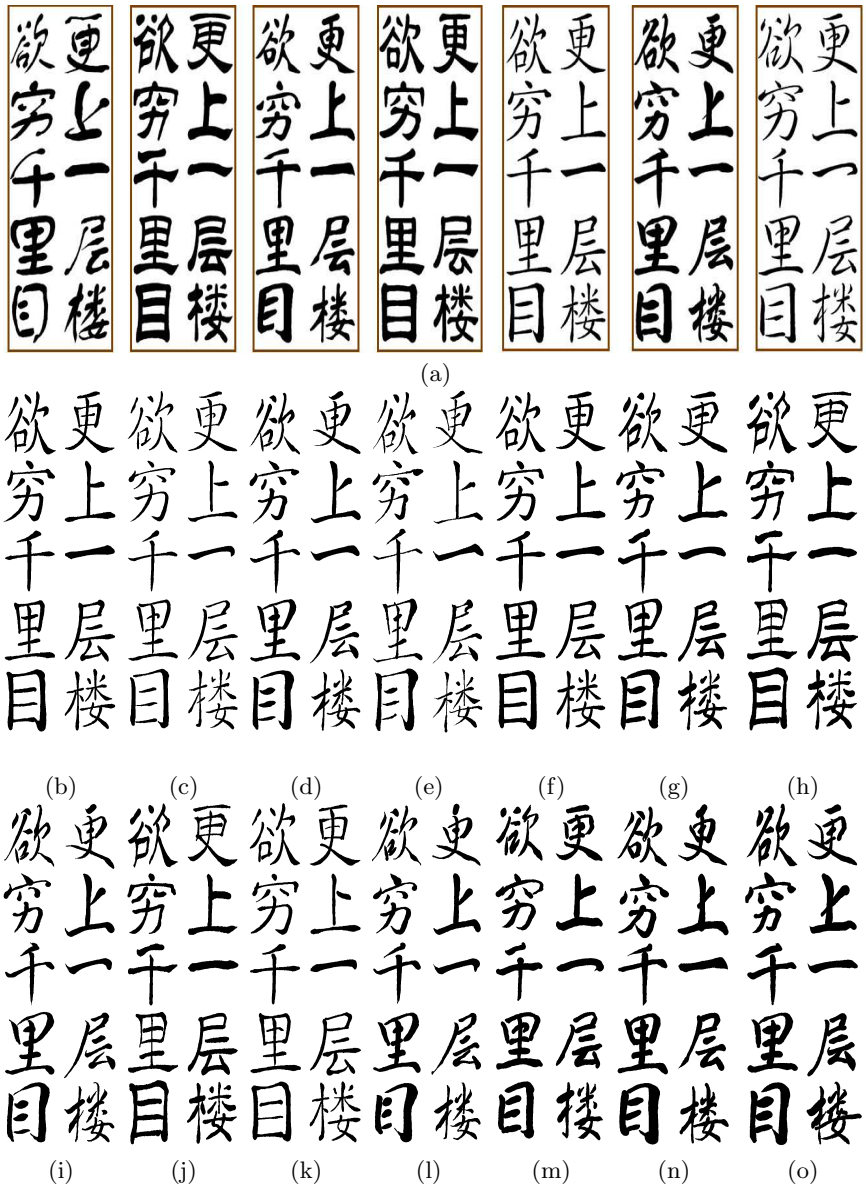
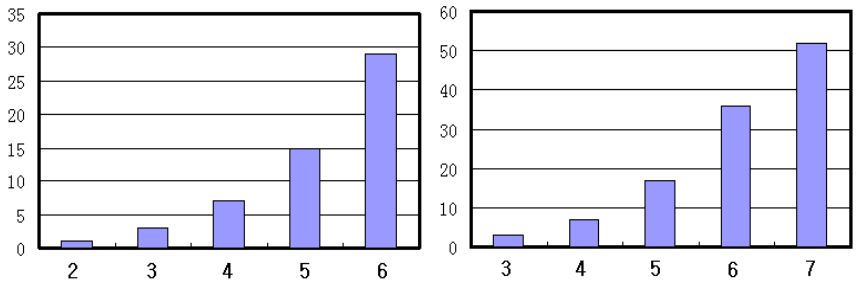


Fig. 8. (a) Learned samples in seven styles. (b)–(o) Some selected samples of newly generated calligraphy



(a) (b)

Fig. 9. System's creativity (vertical axis—number of acceptable results) against number of learned samples (horizontal axis). (a) Single-character level using linear reasoning. (b) Single-character level using non-linear reasoning

Fig. 10. "Forever running"

using the our prototype system, while the horse was generated through human manipulation using a paint-brush software [29]. The character is in a rapid-running style that projects the same spirit as that of the running horse.

7 Advanced Applications

The system described above is an interesting and practical system by itself. There are other, more advanced applications that are possible using our current framework as a base.

7.1 Recognizing Artistic Calligraphy

Artistic calligraphy tends to contain many distorted characters. Normal optical character recognition techniques (OCR) cannot effectively deal with them because they are mostly based on templates of styles that are commonly used.

However, each person is likely to have his own personal handwriting style. Thus there are countless different styles in the world. With our system, a relationship between the shapes of the same character written in different styles can be established. Such a relationship between even a small number of typical handwriting styles would allow the shapes of the character in a wide range of styles to be predicted by the system. This mechanism can be used in a calligraphy recognition system.

7.2 Beautifying Personal Handwriting

In many places, especially in Asia, handwriting is looked on as something that reflects the quality of a person. In situations where personal handwriting is preferred to writing with regular font types, our system can help beautify a person's handwriting. He first writes his own handwriting. And then the handwriting is input to our system as one source for the simulated analogous reasoning process. He may pick a beautiful style as another source for the reasoning process, or several of them. By manually specifying the reasoning intensities, he can choose to what extent his personal handwriting should be rectified. The computer can remember the setting, and so in the future, he can always generate his beautiful handwriting with the same consistent style.

8 Conclusion and Future Work

With the parametric hierarchical knowledge representation of Chinese calligraphy, the computer is able to create new Chinese calligraphic artwork in a variety of styles fully automatically and in real time after learning a limited number of printed calligraphic inputs. The creative ability of the proposed intelligent calligraphy generation system is very powerful because of the huge feasible space available for the simulated analogous reasoning process. The experimental results show that our approach can indeed generate calligraphic artwork that can

stand among existing artwork, whether they appear to be realistic or completely inventive.

There is a tradeoff between the creativity and the practical acceptability of interpolation results. Too strict a set of constraints can limit the creativity, while too loose a set of constraints could harm the overall acceptability of the results. How to find the best tradeoff point should be a worthwhile future pursuit.

The input calligraphic artwork samples we used are in different well-disciplined styles in the Chinese calligraphy world, based on which our system can produce a large number of new writing styles. How to create a new writing style with a pre-specified constraint on its style (as opposed to a character) is a much harder and more challenging research issue. In order to solve the problem, we need to extract the relationships between the parameters driving the analogy; these parameters include the analogous reasoning's source intensities and quantitative definitions for the visual features of different writing styles. The degree of automation to which our system can learn the sample source knowledge from copybooks can be further improved, which is a nontrivial pattern recognition problem for characters in cursive writing style.

Every piece of newly created calligraphic artwork has a set of source intensities associated with the reasoning. Some generated results are more beautiful than others. An interesting task would be to find out the relationships among these source intensities and how they translate into aesthetic values. This can be called "quantitative aesthetics", which should be a challenging topic for future research.

References

1. B. A. Yanikoglu, P. A. Sandon, Recognizing Off-Line Cursive Handwriting, *Proc. Computer Vision and Pattern Recognition*, 1994.
2. B. Kegl, A. Krzyzak, Piecewise Linear Skeletonization Using Principal Curves, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **24**(1): 59-74, JANUARY, 2002.
3. C.W. Liao, J.S. Huang, Stroke segmentation by Bernstein-Bezier curve fitting, *Pattern Recognition*, **23**(5): 475-484, 1990.
4. E.L. Homer, Extraction of strokes in handwritten characters, *Pattern Recognition*, **33**: 1147-1160, 2000.
5. G. C. Richard, L. Eric, A Survey of Methods and Strategies in Character Segmentation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **18**(7): 690 - 706, July, 1996.
6. H.A. Simon, Style in Design, *Spatial Synthesis in computer Aided Building Design*, Edited by C.M. Eastman. London: Applied Science Pub, 1975.
7. H. Blum, W. D. Wed, *Model for the Perception of Speech and Visual*, **10**(2):119–122, Cambridge, Massachusetts: MIT Press, 1967.
8. H.P. Chiui, D.C. Tseng, A novel stroke-based feature extraction for handwritten Chinese character recognition, *Pattern Recognition*, **32**: 1947-1959, 1999.
9. H.P. Chiu, D.C. Tseng, A feature-preserved thinning algorithm for handwritten Chinese characters, *Signal Processing*, **58**: 203-214, 1997.

10. H. Ogawa, K. Taniguchi, Thinning and stroke segmentation for handwritten Chinese character recognition, *Pattern Recognition*, **15**(4): 299-308, 1982.
11. H.M. Lee, C.W. Huang, C.C. Sheu, A fuzzy rule-based system for handwritten Chinese characters recognition based on radical extraction, *Fuzzy Sets and Systems*, **100**: 59-70, 1998.
12. J. G. Carbonell, Learning by analogy: formalizing and generalizing plans from past experience, *Machine Learning II. Calif: Kaufmann, os Altos*, 415-435, 1983.
13. J.Y. Lin, Z. Chen, A Chinese-character thinning algorithm based on global features and contour information, *Pattern Recognition*, **28**(4): 493-512, 1995.
14. J.J. Zou, H. Yan, Extracting strokes from static line images based on selective searching, *Pattern Recognition*, **32**: 935-946, 1999.
15. K. P. Hall, Computational Approaches to Analogical Reasoning – A comparative Analysis, *Artificial Intelligence*, **39**. 1989
16. K. J. Holyoak, Analogical thinking and human intelligence, *Advances in the Psychology of Human Intelligence, (ed. Sternberg, R.H.) Vol. 2, Erlbauum: Hillsdate. N.J.*, 1984.
17. K. P. Chan, Y. S. Cheung, Fuzzy-attribute graph with application to Chinese character recognition, *IEEE Trans. on Systems, Man and Cybernetics*, **22**(1):153 -160, 1992.
18. L.Y. Tseng, C.T. Chuang, An efficient knowledge-based stroke extraction method for multi-font Chinese characters, *Pattern Recognition*, **25**: 1445-1458, 1992.
19. M. Gick, K. J. Holyoak, Analogical problem solving, *Cognitive Psychology*, **12**:306, 1980.
20. M. Keane, On drawing analogies when solving problems: a theory and test of solution generation in an analogical problem solving task, *British Journal of Psychology*, **76**:449, 1985.
21. M. Keane, Analogical mechanisms, *Artificial Intelligence Review*, **2**:229, 1988.
22. M. P. Deseilligny, G. Stamon, C. Y. Suen, Veinerization: A New Shape Description for Flexible Skeletonization, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **20**(5): 505-521, MAY, 1998.
23. P. Zhu, P. M. Chirlian, On Critical Point Detection of Digital Shapes, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **17**(8):737–748, 1995.
24. P. H. Winston, Learning and reasoning and reasoning by analogy, *Communications of The ACM*, **23**:689, 1980.
25. P. H. Winston, Learning new principles from precedents and exercises, *Artificial Intelligence*, **19**:321, 1982.
26. R. He, H. Yan, Stroke extraction as pre-processing step to improve thinning results of Chinese characters, *Pattern Recognition Letters*, **21**: 817-825, 2000.
27. S. Chen, F. Y. Shih, Skeletonization for fuzzy degraded character images, *IEEE Trans. on Image Processing*, **5**(10): 1481 -1485, Oct 1996.
28. S. Strassmann, Hairy brushes, *Proc. of Siggraph*, ACM Press, 225-232, 1986.
29. S. Xu, M. Tang, F.C.M. Lau, Y.H. Pan, A solid model based virtual hairy brush, *Proc. of Eurographics 2002*, Saarbrucken, Germany, September 2002.
30. T. Nishita, S. Takita, E. Nakamae, A display algorithm of brush strokes using Bezier functions, *Proc. of CGI '93*, 244–257, 1993.
31. T. G. Evan, A paradigm for the solution of a class of geometric analogy intelligence test questions, *Semantic Information Processing*, ed. Minsk, M., Cambridge, MIT Press, 1968.
32. X. Sun, L. Yang, Y.Y. Tang, Y. Hu, A new stroke extraction method of Chinese characters, *International Journal of Pattern Recognition and Artificial Intelligence*, **15**(4): 707-721, 2001.

33. Y. Chua, Bezier brush strokes, *Comput. Aided Design*, **22**(9):550–555, 1990.
34. Y. Xia, Skeletonization Via the Realization of the Fire Front's Propagation and Extinction in Digital Binary Shapes, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **11**(10): 1076-1086, October, 1989.
35. Z. Pan, X. Ma, M. Zhang, J. Shi, Chinese font composition method based on algebraic system of geometric shapes, *Comput. & Graphics* , **Vol.18, 21**(3): 321-328, 1997.