Weight Assignment in Dissimilarity Function for Chinese Cursive Script Character Image Retrieval Using Genetic Algorithm

David Yuk-Ming Chan and Irwin King
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
E-mail: {ymchan,king}@cse.cuhk.edu.hk

Abstract When solving the problem of content based-image retrieval, using a single image attribute may not have enough discriminative information for retrieval. On the other hand, when multiple features are used, it is hard to determine the suitable weighting factors to various shape features. Therefore, we proposed to use a Genetic Algorithm (GA) to determine the weighting factors in the dissimilarity function for trademark image retrieval in [3]. In this paper, we use the same technique to find the weights in the dissimilarity function for image retrieval in a Chinese cursive script character image database. Several shape features are chosen to represent a Chinese calligraphy character. They are edge direction histogram of the original image, eccentricity of the original, thinned, and normalized image, and the first three invariant moments of the original, thinned, and normalized image. A database of 1400 monochromatic images was tested. From the results, the learned dissimilarity function increased the accuracy of retrievals. Besides, our system was robust to retrieve deformed images. We also compared our approach with other integration methods. Experimental results are presented to show the feasibility and practicability of the proposed system.

1 Introduction

Chinese calligraphy is invaluable in the history of Chinese civilization. Graphically, it is an abstract art and has a rich variety of form and design. Practically, it is a kind of written language. There are five styles of Chinese calligraphy. Among them, the cursive script style most closely approaches abstract art. In this research, we try to solve the problem of cursive script style character image retrieval. The features shared by all cursive styles are a simplified structure, running together of strokes, rapidly written and a low level of legibility. Figure 1 shows a page of Chinese cursive style calligraphy [15].

Cursive script characters usually have a low level of legibility; therefore, it is challenging to develop a Chinese cursive script character image retrieval system. We use several shape features including edge direction histogram of the original characters, invariant moments of the original, thinned and normalized characters, and eccentricities of the original, thinned and normalized characters. Our approach is to integrate the shape features using the weights found by the GA, so as to increase the robustness and accuracy of retrieval.

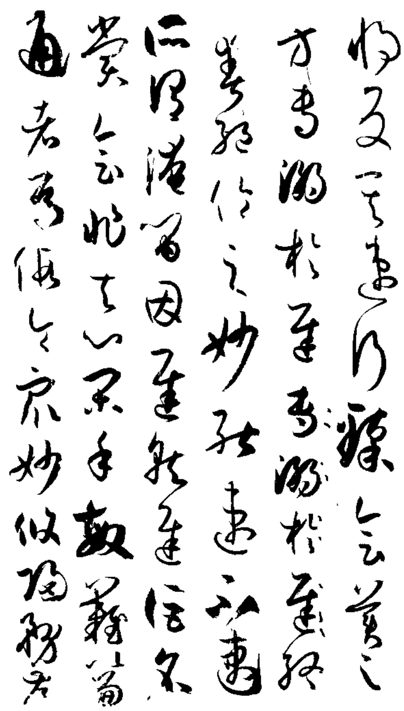


Figure 1: A page of Chinese cursive style calligraphy.

Experiments have been conducted on a database of 1400 cursive script character images. From the results, the proposed technique can handle several different kinds of deformations. The retrieval time for each query was less than 12 seconds on average.

The paper is organized as follows. In Section 2, previous work on content-based image retrieval is reviewed. In Section 3, the differences between image retrieval in trademark database and cursive script character database are discussed. The supervised learning method using GA [3] is reviewed in Section 4. Section 5 describes the cursive script character image retrieval model. Experimental results are presented in Section 6 and Section 7 concludes our report.

2 Literature Review

There are various techniques that deal with image retrieval based on shape similarity. The QBIC (Query By Image Content) system [17, 5, 1] serves as an image database filter which allows queries of large image databases based on visual image content such as color percentages, color layout, and textures occurring in the images. STAR (System for Trademark Archival and Retrieval) system [13, 20] uses features based on R, G, and B color components, invariant moments, and Fourier descriptors extracted from manually isolated objects. Kim *et. al.* [12] developed a trademark retrieval system, which uses Zernike or pseudo-Zernike moments of an image as a feature set. The retrieval scheme is based on visually salient feature that dominantly affects the global shape of the trademark. Jain *et. al.* [11] proposed a two-stage hierarchical image retrieval system and the system was tested on a trademark database. In the first stage, easily computable features like histogram of edge angles and moment invariants are used while in the second stage, the plausible retrievals from the first stage are screened using a deformable template matching process. Logo similarity matching based on positive and negative shape features was suggested by Soffer *et. al.* [19]. Mokhtarian *et. al.* [16] proposed the boundary-based similar shape retrieval method using the maxima of curvature scale space.

3 Comparison to Trademark Retrieval Problem

There are several differences between the trademark image retrieval problem and the cursive script character image retrieval problem based on shape.

3.1 Feature Selection

The goals of the two systems are different; therefore, the shape features to be used are different. The goal of the trademark retrieval problem is to detect the infringement cases based on visual similarity. The shape features to be used in such a system should be insensitive to translation, rotation, and size variations. Moreover, mirror resemblances existing in trademarks must be considered. On the other hand, for a Chinese cursive script character retrieval system, the similarity measure must be insensitive to translation and size variations. Rotational invariance is not important because the character is usually directed upward.

3.2 Speed of System

For a trademark retrieval system, the speed can be slow because the tasks can be done offline. However, for a cursive script character retrieval system, speed is an important factor. The speed of the system should be fast.

3.3 Noises and Variations

A trademark usually has less noise and styles. However, in handwritten characters, there exist many types of variations, such as position, size, inclination, and shape distortions due to various writing styles of human beings. For cursive style, the situation is much worse because there are many variations.

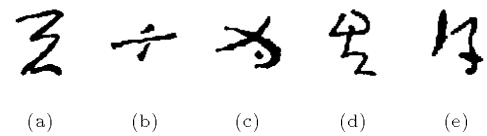


Figure 2: Characters with simplified structures.

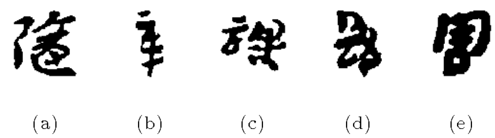


Figure 3: Characters with strokes jammed together.

1. **Simplified Structures:** The characters often have simplified structures. Some examples are given in Fig. 2.
2. **Jammed Strokes:** Some of the strokes in a character are usually jammed together as shown in Fig. 3.
3. **Variation of Aspect Ratio:** The aspect ratio and the width of strokes of the same character may have a large variation. Several examples are shown in Fig. 4.
4. **Variation of Style:** Figure 5 shows 27 different cursive styles of the same character. Some of them cannot even be recognized by human beings easily.

Due to the reasons stated above, it is challenging to develop a Chinese cursive style calligraphy image retrieval system based on shape. It is not our target to retrieve all the other styles of the character given a particular style of a character because it is quite impossible to achieve; on the other hand, it is our aim to find the similar images in the database given a query image.

4 Finding Weight in Dissimilarity Function

The problem of finding weights in the dissimilarity function to integrate different shape features for image retrieval, and the supervised learning method using GA [3] is reviewed in this Section.

4.1 Problem Definition

Definition 1 (Feature Extraction Function) *Given an image I and a set of feature parameters $\boldsymbol{\theta} = \{\theta_i\}_{i=1}^{n}$, a feature extraction function f is defined as*

$$f : I \times \boldsymbol{\theta} \to \mathcal{R}^d \ , \qquad (1)$$

which extracts a real-valued d-dimensional feature vector.

Definition 2 (Integrated Dissimilarity) *Let x_f^I be a feature vector of an image I on the basis of a feature*

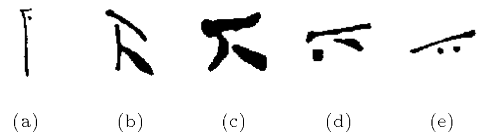


Figure 4: The aspect ratio and the width of strokes for the same character varies.

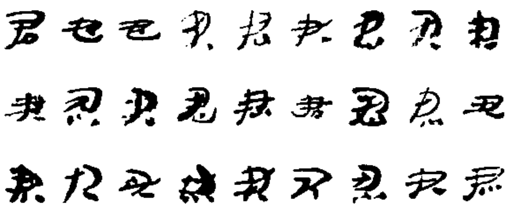


Figure 5: There can be a large variety of styles for the same character.

extraction function f. Then, the integrated dissimilarity function, D_t, between two images I_1 and I_2 is defined as

$$D_t(I_1, I_2) = \sum_{i=1}^{n} D_{f_i} \frac{w_i}{\sum_{j=1}^{n} w_j} \ , \tag{2}$$

where f_i is a feature extraction function, D_{f_i} is the Euclidean distance between the feature vector $x_{f_i}^{I_1}$ and feature vector $x_{f_i}^{I_2}$, w_i is the weight assigned to feature vector set i and n is the number of the feature vector sets.

Definition 3 (Training Pair) *A training pair TP is defined as*

$$TP = (I_T, I_S) \ , \tag{3}$$

where $I_T \in DB$ is the target image for a query and $I_S \in DB$ is the user defined best matched image.

Definition 4 (Ranking Score Function) *Given a training pair $TP = (I_T, I_S)$, r is the ranking of I_S for the query I_T. The ranking score function s is defined as*

$$s : r \to \mathcal{R} \ . \tag{4}$$

The higher the ranking is, the higher is the score. In our implementation, the ranking score function s was defined as

$$s(r) = \begin{cases} \frac{k+1-r}{k}, & \text{if } r \leq k \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where r is the ranking of the target and k is the lowest ranking that has a score.

Definition 5 (Total Count) *Given an integrated dissimilarity function D_t, m training pairs, the total count $TC(\boldsymbol{w})$ is defined as*

$$TC(\boldsymbol{w}) = \sum_{i=1}^{m} s(r_i) \ , \tag{6}$$

where s is the ranking score function, r_i is the ranking for the target i using D_t, and $\boldsymbol{w}$ is the set of weights in D_t.

Definition 6 (Optimal Weights) *The set of weights $\boldsymbol{w}$ in a dissimilarity function D_t is defined as $\boldsymbol{w} = \{w_i\}_{i=1}^{n}$, where w_i is the weight assigned to feature vector set i. The problem is to find $\boldsymbol{w}$ such that $TC(\boldsymbol{w})$ is maximized, i.e.,*

$$\arg\max_{\boldsymbol{w}} \ TC(\boldsymbol{w}) \ . \tag{7}$$

4.2 Finding Weights using Genetic Algorithm

A genetic algorithm (GA) is an optimization method based on the evolutionary metaphor. A short introduction of GA can be found in [18]. The details of our GA are presented in this section.

Chromosome Representation

A chromosome representation is used to describe an individual in the population of interest. A chromosome in our GA is defined as

$$c = (w_1, w_2, \ldots, w_i, \ldots, w_n), \tag{8}$$

where w_i is the weight assigned to feature vector set i and n is the number of feature vector sets.

Selection Function

A selection function plays a vital role in a GA because it selects individuals to reproduce successive generations. A probabilistic selection is performed based on the individual's fitness such that the better individuals have an increased chance of being selected.

The selection method, Roulette wheel, proposed by Holland [7] is used in our implementation. The probability, P_i, for each individual is defined by

$$P[Individual\ i\ is\ chosen] = \frac{F_i}{\sum_{j=1}^{PopSize} F_j} \ , \tag{9}$$

where F_i is equal to the fitness of individual i. A series of N random numbers is generated and compared against the cumulative probability, $C_i = \sum_{j=1}^{i} P_j$, of the population. If $C_{i-1} < U(0,1) \leq C_i$, the individual is selected.

Genetic Operators

Genetic operators provide the basic searching mechanism of the GA. The operators are used to create new solutions based on existing solutions in the population. The operators including simple crossover and uniform mutation are used in our implementation. For more details, please refer to [7, 6, 4].

Initialization, Termination and Evaluation Function

The initial population is randomly generated. The stopping criterion is a predefined maximum number of generations and the evaluation function is the total count $TC(\boldsymbol{w})$ defined in Definition 5.

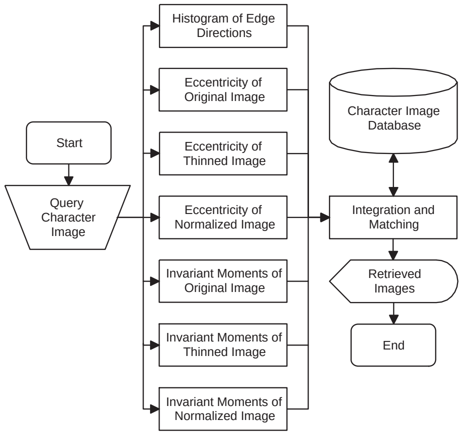
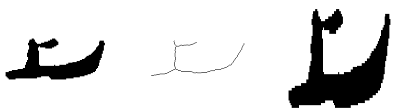


Figure 6: The proposed system.



(a) (b) (c)

Figure 7: An example of the (a) original image, (b) thinned image, and (c) normalized image.

5 Image Retrieval Model

In the proposed system, several shape features are chosen to represent a cursive script character (see Fig. 6). They are edge direction histogram [11] of the original image, eccentricity [9] of the original, thinned and linearly normalized image, and the invariant moments [8] of the original, thinned and linearly normalized image. Note that for the invariant moments, we use the first 3 terms to represent an image.

5.1 Feature Selection

For each image, we find its thinned image and its linearly normalized image. A thinned character has a one-pixel width and a normalized character has an aspect ratio of one. Figure 7 shows an example of the original, thinned and linearly normalized image.

Then, we store the edge directions [11] of the original image, eccentricity [9] of the original, thinned and linearly normalized image, and the invariant moments [8] of the original, thinned and linearly normalized image as a feature set for image retrieval.

Invariant Moments

The shape of an object can be represented by invariant moments [8]. These features are invariant under rotation, scale, translation, and reflection of images. For a two-dimensional image, $f(x, y)$, the central moment of order $(p+q)$ is given by

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y). \qquad (10)$$

From the second-order moments and third-order moments a set of seven invariants moments, which is invariant to translation, scale change and rotation, has been derived [8]:

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} && (11)\\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 && (12)\\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 && (13)\\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} - \eta_{03})^2 && (14)\\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - && (15)\\
&\quad 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) && (16)\\
&\quad [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] && (17)\\
\phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] && (18)\\
&\quad +4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) && (19)\\
\phi_7 &= (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - && (20)\\
&\quad 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) && (21)\\
&\quad [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] && (22)
\end{aligned}$$

where η_{pq} are normalized central moments defined by

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{(p+q+2)/2}} \qquad (23)$$

for $p+q = 2, 3, \ldots$, and μ_{pq} is the central moment of order $(p+q)$ of an image. The invariance properties have been proven for the case of continuous functions by [8]. In our image retrieval system, we use the first 3 terms to represent a shape.

Eccentricity

The ratio of the major axis to the minor axes is called eccentricity [9] and is defined as

$$\epsilon = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}{(\mu_{20} + \mu_{02})^2}\ , \qquad (24)$$

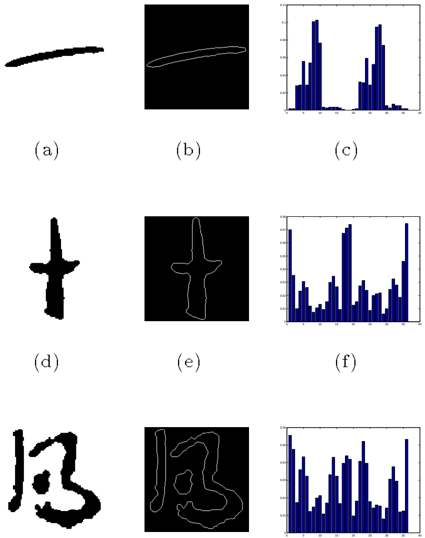
where μ_{pq} is the central moment of order $(p+q)$ of an image. The value of eccentricity ranges from 0 to 1. It is zero for a circular object and one for a line-shaped object. Eccentricity is invariant to translation, rotation and scaling.

Edge Directions

A histogram of edge directions is used to represent a shape. The edge information is extracted using Canny edge operator [2] with $\sigma = 1$ and Gaussian mask of size $= 9$. The histogram edge directions is represented by 36 bins, each spanning 10°. Euclidean distance metric is used to compute the dissimilarity value between two edge directions histograms. Figure 8 demonstrates the shape representation using edge directions.

In order to achieve invariance to scale, the histograms are normalized to the number of edges points in the image. The edge directions representation method is not rotation invariant. To reduce the effect of rotation, we smooth the histograms using the method proposed in [10]. A histogram is treated as a 1-D discrete signal. Smoothing is defined as:

$$I_s[i] = \frac{\sum_{j=i-k}^{i+k} I[j]}{2k+1}\ , \qquad (25)$$



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 8: An example of shape representation using edge directions. (a), (d) and (g) show three database images, (b), (e) and (h) show the corresponding edge images, and (c), (f) and (i) show the corresponding histogram of edge directions.

where I_s is the smoothed histogram. I is the original histogram and the parameter k determines the degree of smoothing. In our experiments, we used $k = 1$.

5.2 Integration of Shape Attributes

The extracted feature vectors of the images are stored in the database. The weighting factors of the dissimilarity function are found beforehand using the GA. When a user raises a query, the features of the query image are first extracted. Then the extracted features are matched linearly with the features in the database and then integrated by the dissimilarity function (see Definition 2). After that, the images are sorted and displayed in the order of similarity.

6 Experimental Results and Discussion

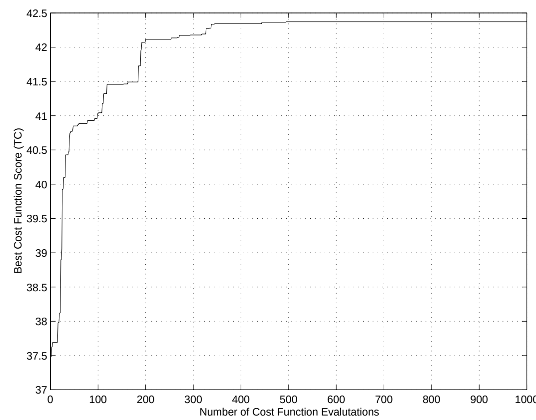
The cursive script character image retrieval system is implemented under UNIX using Matlab 5.3. The prototype system has 1400 binarized Chinese cursive script character images. Each of them was normalized to the size of 200 by 200 pixels. These images were scanned from the book [14] with the resolution of 150dpi. In this section, we present the results of the GA training and the results of image retrieval.

6.1 Experiment 1: Weight Assignment using Genetic Algorithm

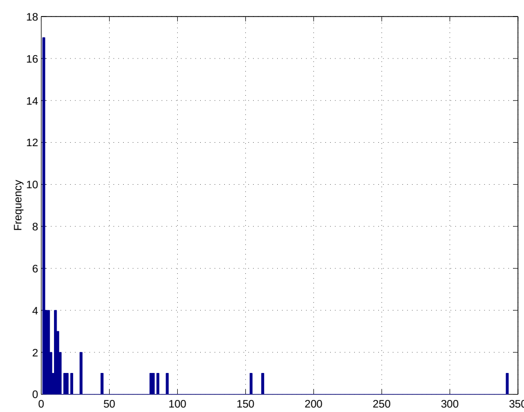
The proposed GA was tested with the following setup. The population size $PopSize$ was 30 and the maximum number of iterations was 1000. In addition, there were 50 training pairs TP. The size of the database DB was 1400. The values of the weights (genes) were bounded by 0 and 1. The probability of application of crossover was 0.6 and the probability of application of mutation was 0.05. The experiments were repeated 10 times with different random seeds.

The value of k in the ranking score function was set to 140. Therefore, if the target is ranked in the first position, $TC(\boldsymbol{w})$ is increased by 1. If the target is ranked in the second position, $TC(\boldsymbol{w})$ is increased by $\frac{139}{140}$ and so on. We set k to be 140 because it is 10% of the total number of images in the database.

Figure 9 shows the best result obtained among the 10 trials. From Fig. 9(a), the GA improved the accuracy of the retrievals of the training examples by changing the weights in the dissimilarity function. It converged in 500 iterations with a total count of 42.37, which means on average, the distance function ranks the target of a query in a training example to the 22nd position. The best weights found are presented in Table 1 and were used as the final weights for retrieval afterward.



(a)



(b)

Figure 9: Results of Experiment 1. (a) Best cost function value (TC) versus the number of cost function evaluations. (b) The ranking histogram.

The performance of the final distance function can be further analyzed by the ranking histogram in Fig. 9(b). From the histogram, 74% of the targets were ranked at

Table 1: Results of Experiment 1. Weights found for different shape features. Note that I_o, I_n, and I_t are the original image, normalized image and thinned image, respectively.

Feature	Weight	%
Edge directions histogram of I_o	0.0224	0.40
Eccentricity of I_o	0.0433	0.77
Eccentricity of I_n	0.1497	2.66
Eccentricity of I_t	0.0494	0.88
First invariant moment of I_o	0.3756	6.67
Second invariant moment of I_o	0.9073	16.11
Third invariant moment of I_o	0.9854	17.50
First invariant moment of I_n	0.1156	2.05
Second invariant moment of I_n	0.1051	1.87
Third invariant moment of I_n	0.9962	17.69
First invariant moment of I_t	0.9755	17.32
Second invariant moment of I_t	0.0136	0.24
Third invariant moment of I_t	0.8930	15.86

the top 14 positions, 80% were ranked at the top 28 positions, 86% were ranked at the top 70 positions, and 94% were ranked at the top 140 positions. However, there were several targets were ranked very low. This is because images that appear to be similar need not be similar in their shape. Another reason is that the features that we used cannot model the similarity of those training examples. However, we believe that by introducing more effective features, the accuracy can be improved.

6.2 Experiment 2: Speed on Feature Extraction and Retrieval

In this experiment, we tested the speed on feature extraction and image retrieval. When a query was submitted to the system, its features would be computed first. Next, the similarity measure between the query and all images in the database were computed. The average elapse time of 100 trials for feature extraction of the query image and database query are listed in Table 2.

Table 2: Results of Experiment 2. Average elapse time for retrieving an image from a database of 1400 images on a Sun Ultra 5 Machine.

Feature Extraction	Database query	Total
11.61s	0.04s	11.65s

6.3 Experiment 3: Evaluation by Recall for Deformed Images

Our aim is to develop a Chinese cursive script image retrieval system that is insensitive to variations on image deformation. In this experiment, we tested the behavior of our image retrieval system in the presence of the deformation transformation as shown in Fig. 10. The cursive character images in Fig. 11 were used to generate a set of 100 deformed images as shown in Fig. 12. The deformed images were submitted as query images to our retrieval system to examine whether the deformed images can retrieve their original images or not.

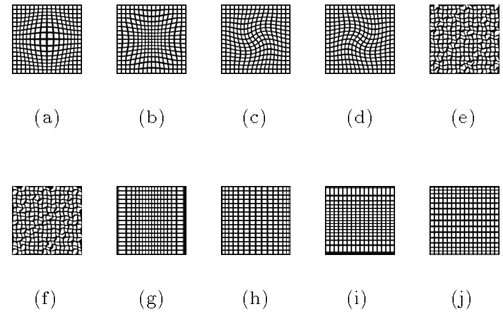


Figure 10: Distortions including Pinch: (a) and (b); Twirl: (c) and (d), Ripple: (e) and (f); Horizontal Extension (HE): (g) and (h); and Vertical Extension (VE): (i) and (j).

Figure 11: Original images.

Table 3 presents the results of recall rate in the top 70 candidates queried by the deformed images, where n corresponds to the position of the correct retrieval. From the results, 90% of the deformed images can recall their original ones in the top 28 candidates which is equal to 2% of the database size. The average recall rate of the original ones among the top 70 candidates queried by deformed images was 94%.

6.4 Experiment 4: Evaluation by Recall for Rotated and Scaled Images

In order to test the recall rate for rotated and scaled images, we conducted the following experiment.

1. **Rotated Query:** Every image in Fig. 11 was rotated arbitrarily with the range of $\pm 10°$ and then presented as the query image. We only tested with the range stated above because our target is not to retrieve arbitrarily rotated images and we assume the query image to our system is directed upward.

2. **Scaled Query:** Every image in Fig. 11 was scaled arbitrarily and then presented as the query image. The scaling factors were bounded by 0.5 and 50.

The experiments were repeated with 100 rotated queries and 100 scaled queries. From the results presented in Table 4, we notice that the shape features were very

Figure 12: 100 queries deformed from 10 character images for performance evaluation.

Table 3: Results of Experiment 3. Recall for deformed images. n refers to the position of the correct retrieval. For the query nature, please refer to Fig. 10.

Query Nature	$n = 1$ (%)	$n \leq 5$ (%)	$n \leq 14$ (%)	$n \leq 28$ (%)	$n \leq 70$ (%)
(a)	50	80	90	90	90
(b)	40	80	90	90	100
(c)	10	50	80	80	90
(d)	10	40	80	80	80
(e)	20	60	80	80	90
(f)	20	40	70	80	90
(g)	70	100	100	100	100
(h)	60	90	100	100	100
(i)	80	100	100	100	100
(j)	70	100	100	100	100
Avg.	43	74	89	90	94

effective in retrieving scaled images. However, the system only recalled 98% of the rotated images in the top 70 positions. This is because we used an edge direction histogram to represent an image, which is rotation sensitive. We consider not to use an edge direction histogram for representing a character image in our future development.

Table 4: Results of Experiment 4. Recall for rotated and scaled images. n refers to the position of the correct retrieval.

Query Nature	$n = 1$ (%)	$n \leq 5$ (%)	$n \leq 14$ (%)	$n \leq 28$ (%)	$n \leq 70$ (%)
Rotated	64	83	91	96	98
Scaled	100	100	100	100	100

6.5 Experiment 5: Comparison of Different Integration Methods

In this experiment, we compare the recall rate of different integration methods. The deformed images in Fig. 12 were used as queries to recall their original images. The integration methods include:

1. **Equal weights.** All the weights were set to 1 for all the features in the dissimilarity function. This method has been used in [11] for integrating two shape features.

2. **Weights determined by individual recall rate.** A weighted Euclidean distance was used as the distance function. The weights are proportional to the accuracy of retrievals using individual features. This method was proposed by [11] but they did not tested its performance.

3. **Integration with the method used in the QBIC system [17, 5, 1].** The distance between two images is computed using a weighted Euclidean distance with the inverse of feature variances used for normalization. For example, the distance between image i and j is defined as

$$d_{ij} = \sum_{k} \frac{(k_i - k_j)^2}{\sigma_k^2} , \qquad (26)$$

where k is a feature.

4. **Weights found by GA.** The distance between two images is computed using a weighted Euclidean distance with the weights found by our proposed GA.

Table 5: Results of Experiment 5. Retrieval results on the basis of different integration methods. n refers to the position of the correct retrieval. (a) Equal weights. (b) Weights determined by individual recall rate. (c) QBIC. (d) Our system.

Query Nature	$n = 1$ (%)	$n \leq 5$ (%)	$n \leq 14$ (%)	$n \leq 28$ (%)	$n \leq 70$ (%)
(a)	43	62	73	80	88
(b)	41	54	69	76	83
(c)	51	65	80	86	93
(d)	43	74	89	90	94

The results of the testing are presented in Table 5. From the results, the dissimilarity function found by our GA has a better recall rate than that of the other methods. This is expected because the weights were optimized by the GA based on the training examples given by the users. Hence, its recall rate should be better than that of the other integration methods.

7 Conclusion and Future Work

In this research, a prototype system for Chinese cursive script character image retrieval has been developed. Several shape features including histogram of edge directions, invariant moments and eccentricities were used to increase the robustness of the system. We also used the original images, the thinned images and the normalized images to extract the shape features. The weighting factors of the shape attributes in the dissimilarity function were determined by the supervised learning method using GA. The experimental results proved the feasibility and practicability of the proposed scheme. Future work deals with adding other effective shape features to our current system and increasing the database size. Our final target is to build a tool for the calligraphers and artists to do comparative study on Chinese cursive script characters depending on different dynasties, authors, periods of the same author and etc.

References

[1] J. Ashley, R. Barber, M. Flickner, J. Hafner, D. Lee, W. Niblack, and D. Petkovic. Automatic and semi-automatic methods for image annotation and retrieval in QBIC. In *Proceedings of Storage and Retrieval for Image and Video Databases III*, volume 2420, pages 24–35, February 1995.

[2] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[3] D.Y.M. Chan and I. King. Genetic algorithm for weights assignment in dissimilarity function for trademark retrieval. In *Third International Conf. on Visual Information and Information Systems (VISUAL'99)*, volume 1614 of *Lecture Notes in Computer Science*, pages 557–565, The Netherlands, 1999. Springer.

[4] K.A. De Jong. Adaptive system design: A genetic approach. *IEEE Transactions on Systems, Man, and Cybernetics*, 10(9):556–574, 1980.

[5] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligient Information Systems*, 3(3/4):231–262, July 1994.

[6] J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):122–128, 1986.

[7] J. Holland. *Adaptation in natural and artifical systems.* The University of Michigan Press, 1975.

[8] M.K. Hu. Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, 8, 1962.

[9] B. Jahne. *Digital Image Processing: Concepts, Algorithms and Scientific Applications.* Springer-Verlag, Berlin; New York, 4 edition, 1997.

[10] A.K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, 1996.

[11] A.K. Jain and A. Vailaya. Shape-based retrieval: A case study with trademark image databases. *Pattern Recognition*, 31(9):1369–1390, 1998.

[12] Y.S. Kim and W.Y. Kim. Content-Based Trademark Retrieval System Using Visually Salient feature. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 307–312, 1997.

[13] C.P. Lam, J.K. Wu, and B. Mehtre. STAR - A System for Trademark Archival and Retrieval. In *2nd Asian Conf. on Computer Vision*, volume 3, pages 214–217, 1995.

[14] H.H. Lu. *Han Tai Chien Tu Tsao Tzu Pien.* Shang-hai : Shang-hai Shu Hua Chu Pan She, 1989., ti 1 pan. edition, 1989.

[15] K.C. Ma. *Shu Pu I Chu / Ma Kuo-Chuan I Chu.* Hong Kong : Shao-Hua Wen Hua Fu Wu She, 1977.

[16] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In *First International Workshop on Image Databases and Multi-Media Search*, pages 35–42, Amsterdam, The Netherlands, August 1996.

[17] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: querying images by content using color, texture and shape. In *Storage and Retrieval for Image and Video Databases*, volume 1908, pages 173–187, 1993.

[18] G. Roth and M.D. Levine. Geometric primitive extraction using a genetic algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(9), Sep 1994.

[19] A. Soffer and H. Samet. Using negative shape features for logo similarity matching. In *14the International Conf. on Pattern Recognition*, volume 1, pages 571–573, 1998.

[20] J.K. Wu, B.M. Mehtre, Y.J. Gao, Chian-Prong Lam, and A. Desai Narasimhalu. STAR—A Multimedia Database System For Trademark Registration. In Witold Litwin and Tore Risch, editors, *Applications of Databases, First International Conf.*, volume 819 of *Lecture Notes in Computer Science*, pages 109–122, Vadstena, Sweden, 21–23 June 1994. Springer.