Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Efficient shape matching for Chinese calligraphic character retrieval*

Wei-ming LU, Jiang-qin WU‡, Bao-gang WEI, Yue-ting ZHUANG
(*School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)
E-mail: {luwm, wujq, wbg, yzhuang}@zju.edu.cn

Received Jan. 2, 2011; Revision accepted July 13, 2011; Crosschecked Sept. 28, 2011

Abstract: An efficient search method is desired for calligraphic characters due to the explosive growth of calligraphy works in digital libraries. However, traditional optical character recognition (OCR) and handwritten character recognition (HCR) technologies are not suitable for calligraphic character retrieval. In this paper, a novel shape descriptor called SC-HoG is proposed by integrating global and local features for more discriminability, where a gradient descent algorithm is used to learn the optimal combining parameter. Then two efficient methods, keypoint-based method and locality sensitive hashing (LSH) based method, are proposed to accelerate the retrieval by reducing the feature set and converting the feature set to a feature vector. Finally, a re-ranking method is described for practicability. The approach filters query-dissimilar characters using the LSH-based method to obtain candidates first, and then re-ranks the candidates using the keypoint- or sample-based method. Experimental results demonstrate that our approaches are effective and efficient for calligraphic character retrieval.

Key words: Calligraphy, Shape feature, Character retrieval, Efficient matching
doi:10.1631/jzus.C1100005 **Document code:** A **CLC number:** TP391.4

1 Introduction

Chinese calligraphy works are a valuable part of Chinese culture heritage. As more and more calligraphy images are digitized, preserved, and exhibited in digital libraries, it is urgent to provide an efficient approach for calligraphic character retrieval. However, traditional optical character recognition (OCR) and handwritten character recognition (HCR) technologies do not work well for calligraphic characters due to:

1. Complexity: Different writing styles exist in Chinese calligraphy, such as seal script, clerical script, standard script, semi-cursive script, and cursive script (Fig. 1). Moreover, XingKai and XingCao mix different styles together, which makes recognition more difficult. In addition, authors tend to write the same character in different shapes even for the same style (Fig. 2).

2. Deformation: Strokes are not regular in calligraphic characters, in that some strokes are connected with each other and some are broken. Taking Fig. 2 as an example, the first stroke in the first character is broken, but it is connected with the second stroke in the fifth character.

3. Degradation: Calligraphic works tend to be degraded by natural changes, which makes characters more noisy.

We tried to use commercial software Hanvon (http://www.hanvon.com/en/) to recognize calligraphic characters, but this failed. The optical character recognition (OCR) results of characters in Figs. 1 and 2 are shown in Fig. 3.

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 60673088, 61070066, and 61103099), the China Postdoctoral Science Foundation (No. 20110491781), the Major National Science and Technology Special Project of China (No. 2010ZX01042-002-003), and the China Academic Digital Associative Library Project

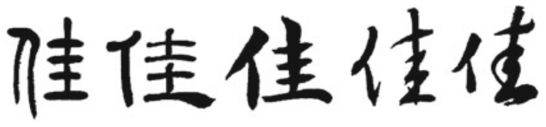


Fig. 1 Five main styles of one character. From left to right: seal script, clerical script, standard script, semi-cursive script, and cursive script

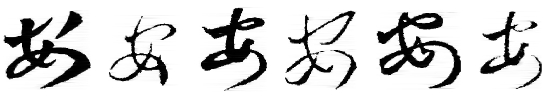


Fig. 2 Characters written by Xi-zhi WANG. Although written by the same author and with the same style, they are quite different

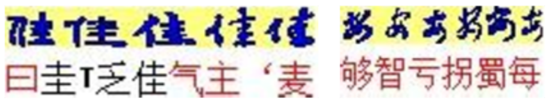


Fig. 3 OCR results of Fig. 1 and Fig. 2 through commercial software Hanvon

Shape feature is robust and would be less affected by complexity, deformation, and degradation. Thus, it can be used for calligraphic character retrieval. Various shape features have been designed, including skeletal representations (Luo *et al.*, 2001; Sebastian *et al.*, 2004; Torsello and Hancock, 2004; Aslan and Tari, 2005; van Eede *et al.*, 2006; Torsello *et al.*, 2007; Bai and Latecki, 2008), shape signature (Yankov *et al.*, 2008), shape context (Belongie *et al.*, 2000), and statistical features (Brandt *et al.*, 2002). We have implemented a calligraphic character retrieval system based on shape matching (Zhuang *et al.*, 2004; 2005). In this system, each character is described by approximate point context (APC), which is based on shape context. When calculating the shape similarity between two characters, a correspondence based shape matching method is applied through point-to-point matching. APC describes each contour point by its relationships with the remaining points. However, it treats all the remaining points equally, while neglecting the fact that the points nearby the reference point are much more important and should be described in more detail. Meanwhile, each point should find its corresponding point for matching, which will take considerable time for a large number of contour points. Thus, it is impractical because of its low accuracy and expensive computation in shape matching.

To address the low accuracy and low efficiency problem in our previous implementation (Zhuang *et al.*, 2004; 2005), here we propose a novel shape representation by integrating global and local features to enhance the shape discriminability. Then two efficient matching methods, keypoint-based matching and locality sensitive hashing (LSH) based matching, are proposed to accelerate the method. Finally, a re-ranking method is described by combining these two efficient methods to ensure the retrieval quality and speed.

2 Related work

Computer aided calligraphy research has attracted an increasing amount of interest recently, and several novel approaches have been proposed to address calligraphic problems, including calligraphy processing and analysis (Wang and Lee, 2001; Wong *et al.*, 2006; Zhang *et al.*, 2006), calligraphy retrieval (Zhuang *et al.*, 2005; 2007), calligraphic character recognition (Yu *et al.*, 2008), visualization (Wu *et al.*, 2006), specific style rendering (Zhang *et al.*, 2010), calligraphy automatic generation or synthesis (Xu *et al.*, 2005; Yu and Peng, 2005; Wong *et al.*, 2008), style relationships discovery (Lu *et al.*, 2009; Zhuang *et al.*, 2009), calligraphy grading (Chou *et al.*, 2005; Xu *et al.*, 2007), and verification (Zhang and Zhuang, 2007).

Calligraphic character retrieval is quite crucial in calligraphy research, and it can involve shape-based approaches. Shape is an important visual feature for describing image content, and can be classified into two categories: skeleton-based representation and contour-based representation.

Skeleton-based recognition is widely used for shape description and matching (Luo *et al.*, 2001; Sebastian *et al.*, 2004; Torsello and Hancock, 2004; Aslan and Tari, 2005; van Eede *et al.*, 2006; Torsello *et al.*, 2007; Bai and Latecki, 2008), and has often been represented as a tree or graph. Many researchers have developed several matching approaches for graph- or tree-based representation, such as graph edit distance (Sebastian *et al.*, 2004; Torsello and Hancock, 2004) or shortest path similarity (Bai and Latecki, 2008). Unfortunately, tree- or graph-based representation is available only for simple objects. Thus, we cannot apply it for calligraphic characters which have complex shapes. Although we have proposed a skeleton-based Chinese calligraphic character recognition method (Yu *et al.*, 2008), we did not represent characters by trees or graphs.

Contour is also a discriminative feature for shape, and several methods have been proposed for shape description and matching, such as shape context (SC) (Belongie *et al.*, 2000), inner distance shape context (IDSC) (Ling and Jacobs, 2007), dynamic aligned shape descriptor (D-Shape) (Fornes *et al.*, 2010), structural feature histogram matrix (SFHM) (Zhang and Liu, 2009), and multiple references histogram matrix (MRHM) (Wang *et al.*, 2010). SC characterizes each point by the spatial distribution of the other points relative to the reference point, and has been widely used in many applications (Kortgen *et al.*, 2003; Mori and Malik, 2003; Mortensen *et al.*, 2005; Zhu *et al.*, 2008). For example, a shape detection framework called contour context selection was proposed in Zhu *et al.* (2008) for detecting objects in cluttered images. Kortgen *et al.* (2003) extended shape context for 3D shape retrieval and matching. SC was also used to identify words in adversarial clutter for breaking a visual completely automated public turing test to tell computers and humans apart (CAPTCHA) (Mori and Malik, 2003). D-Shape encodes the spatial probability of appearance of the shape pixels and their context information, and then uses a cyclic version of the dynamic time warping algorithm to match two shapes. However, D-Shape considers only the local context information of the voting points located in concentric circles, which has less information than the SC descriptor. SFHM is generated by statistically gathering the length-ratio histogram and angel histogram for the centroid of a shape. However, it takes only one fixed reference point to compute the histograms; thus, it is too rigid to represent shape. MRHM extends SFHM by using multiple references, and it would be more robust than SFHM for calligraphic character retrieval. IDSC extends SC by replacing Euclidean distance with inner distance, which is defined as the length of the shortest path within the shape boundary. Thus, it is not suitable for shapes with several segments. In other words, it cannot be used for character retrieval, which consists of disconnected radicals commonly. Moreover, because inner distance is sensitive to shape topology, coherent and broken strokes in calligraphic characters would cause IDSC matching problems. Thus, we implement a shape context based calligraphic character retrieval system, as mentioned before.

Shape matching based on shape context is essentially a feature-set matching problem. Each shape is described by a set of features, and then the similarity between two shapes is computed by comparing the two corresponding sets of features. Many studies have focused on efficient feature-set matching recently. Pyramid match kernel (Grauman and Darrell, 2005) and random projection (Dong *et al.*, 2008) were proposed for scale-invariant feature transform (SIFT) feature set matching. In Grauman and Darrell (2005), feature sets were mapped to multi-resolution histograms, and then the histograms were compared with a weighted histogram intersection measure to approximate the feature-set matching. Dong *et al.* (2008) presented a randomized algorithm to embed a set of features into a single high-dimensional vector by LSH to simplify the feature-set matching problem.

To address the shape context set matching problem, two algorithms were proposed in Mori *et al.* (2005) for rapid shape retrieval: the representative shape contexts by performing comparisons based on a small amount of shape contexts, and the shapemes by using vector quantization to obtain prototypical shape pieces. However, it is much difficult to select the representative shape contexts and to determine the number of shapemes.

Many have tried to use other approaches to speed up calligraphic character retrieval. An interactive partial-distance-map (PDM) based high-dimensional indexing scheme was designed by Zhuang *et al.* (2007) specifically to speed up the retrieval. However, it still resorts to shape context feature, and it is difficult to select the reference points when indexing. Meanwhile, there is only a focus on reducing the amount of one-to-one matching, without speeding up the matching method. Zhang *et al.* (2007) sped up character retrieval by filtering out some characters in advance, according to the knowledge of calligraphic characters, such as character complexity, stroke density, and stroke protrusion. However, the method is much more sensitive with shape deformation, which is common in calligraphic characters.

3 Shape representation

In this section, we integrate a global descriptor and a local descriptor as an SC-HoG descriptor for shape matching, and then use a gradient descent algorithm to determine the optimal combining parameter.

3.1 SC-HoG descriptor

The SC-HoG descriptor integrates the SC descriptor and the HoG (histogram of oriented gradients) descriptor to describe shape. Thus, it considers both the positions of points in the whole scope and the direction of nearby points.

The SC descriptor is similar to that described in our previous work (Zhuang *et al.*, 2004), and we use a polar coordinate system to describe each pixel in the character shape. For direction, we use eight bins in the same degree to divide the whole space into eight regions. Then four bins are obtained for each region by dividing the radius with the value of $r = \{R, R/2, R/4, R/8\}$, where R is the radius of image, and the radii are determined relative to the image size. In our case, the character images are scaled to 128×128; thus, $R = 64$ and $r = \{64, 32, 16, 8\}$. Finally, we obtain a total of 32 bins. By using polar coordinate for each pixel, the SC feature for pixel p_i can be represented as $\mathrm{sc}_i =< w_i(1), w_i(2), \cdots, w_i(K) >$, where $w_i(k) = \#\{q_j \neq p_i : q_j \in \mathrm{bin}_i(k)\}, k = 1, 2, \cdots, K$, and $K = 32$ is the number of bins. Fig. 4a shows an example of how these bins are built and computed for a calligraphic character.

The gradient can describe points more accurately than using location only. Thus, we use HoG to describe the nearby points. Although HoG was initially proposed for human detection (Dalal *et al.*, 2005), it has also been used for object recognition (Suard *et al.*, 2006; Bosch *et al.*, 2007). The HoG descriptor accumulates a weighted vote for an edge orientation histogram channel based on the orientation gradient for each pixel in each bin. The circular region was divided into five bins, as in Fig. 4b where the central cell is not divided, called C-HoG (Dalal *et al.*, 2005). In our case, the radii of the central and the surrounding bins are 16 and 32, respectively. In each bin, the gradient orientation is evenly divided into $B = 8$ parts. Thus, we can obtain the HoG feature for bin k of pixel p_i as $h_i(k) =< h_{ik}(1), h_{ik}(2), \ldots, h_{ik}(B) >$, where $h_{ik}(b) = \sum_{(x,y)\in \mathrm{bin}_i(k), 360A(x,y)/B=b} G(x, y)$. Here $G(x, y)$ and $A(x, y)$ are the gradient value and gradient orientation of point (x, y), respectively. Finally, we concatenate the HoG feature of each bin and obtain a total KB-dimensional feature for pixel p_i as $\mathrm{hg}_i =< h_i(1), h_i(2), \cdots, h_i(K) >$, where $K = 5$ is the number of bins.

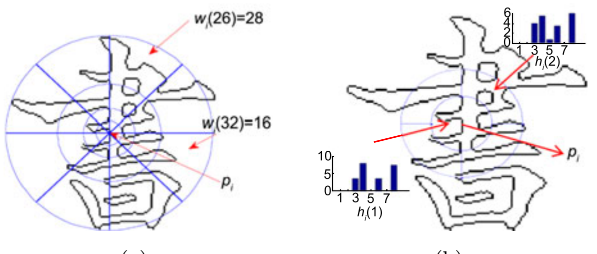


Fig. 4 SC-HoG descriptor, which integrates the SC descriptor (a) and HoG descriptor (b)

Finally, the SC-HoG descriptor can be represented as $< \mathrm{sc}, \mathrm{hg} >$.

3.2 Shape matching

Since points are described with the SC-HoG descriptor, the cost of matching two points p_i and p_j can be calculated by the following linear equation:

$$D(p_i, p_j, \lambda) = \lambda f_1(p_i, p_j) + (1 - \lambda) f_2(p_i, p_j), \quad (1)$$

where $f_1(p_i, p_j) = (2K)^{-1} \sum_{k=1}^{K} \{[w_i(k) - w_j(k)]^2 \cdot [w_i(k)+w_j(k)]^{-1}\}$ denotes the cost of matching with the SC descriptor and $f_2(p_i, p_j)$ denotes the cost of matching with the HoG descriptor, which is also calculated like the SC descriptor. λ is the combining parameter, and when λ equals 1 or 0, it degrades to the SC descriptor or HoG descriptor. The shape similarity between two characters I_i and I_j with parameter λ is

$$\begin{aligned} d(I_i, I_j, \lambda) &= \sum_{s=1}^{|I_i|} D(p_s, p_{\pi(s)}, \lambda) \\ &= \sum_{s=1}^{|I_i|} (\lambda f_1(p_s, p_{\pi(s)}) + (1 - \lambda) f_2(p_s, p_{\pi(s)})) \\ &= \lambda \sum_{s=1}^{|I_i|} f_1(p_s, p_{\pi(s)}) + (1 - \lambda) \sum_{s=1}^{|I_i|} f_2(p_s, p_{\pi(s)}) \\ &= \lambda F_1(I_i, I_j) + (1 - \lambda) F_2(I_i, I_j), \end{aligned} \quad (2)$$

where $\pi(s) = \arg\min_t D(p_s, p_t, \lambda), p_s \in I_i, p_t \in I_j$. That is, for each point $p_s \in I_i$, we should find the 'best' matching point $p_t \in I_j$, and then obtain the total cost by accumulating all the matching costs.

With the orientation property of Chinese character, two match points should be adjacent to each other. Thus, p_s and p_t should be subject to the constraint

$$||p_s - p_t||_2 \leq \text{IMAGESIZE}/4. \tag{3}$$

This accelerates the matching by reducing the search scope. Meanwhile, some noisy matching pairs will occur, whose matching cost is larger than the average cost. Therefore, we filter some matching pairs whose matching cost is larger than one half the average matching cost, thereby reducing the impact of noise. Since the global feature is too coarse and the local feature lacks a global view, using only one feature will bring incorrect matches. Thus, these two features can be integrated to improve the matching accuracy. Fig. 5 shows some incorrect matches with global and local features.

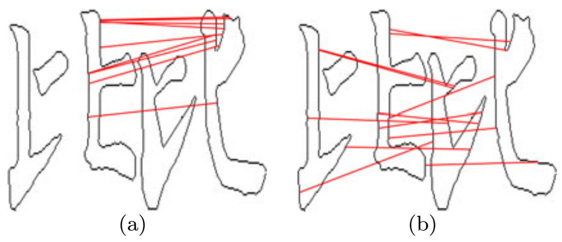


Fig. 5 Match results with different λ, where colored lines indicate the incorrect matches. (a) Local feature, $\lambda = 0.0$; (b) Global feature, $\lambda = 1.0$

3.3 Parameter learning

As described in Section 3.2, the parameter λ is much important for shape matching. Thus, we focus on the λ optimization in this subsection.

Given a query q, let $l^*(q)$ be the optimal rank list and $l(q, \lambda)$ be the rank list generated by Eq. (2). The similarity between these two rank lists can be measured by Kendall's τ, which is defined as

$$\tau(l^*(q), l(q)) = \frac{X - Y}{X + Y} = 1 - \frac{4Y}{M(M-1)}, \tag{4}$$

where X and Y are the numbers of concordant and discordant pairs in two rank lists, respectively, M is the number of characters in the dataset, and we will obtain $X+Y = M(M-1)/2$. Thus, the optimization problem is as follows:

$$\begin{aligned}\lambda &= \arg\max_{\lambda} \sum_{q\in Q} \tau(l^*(q), l(q)) \\ &= \arg\max_{\lambda} \sum_{q\in Q} \left(1 - \frac{4Y}{M(M-1)}\right) \\ &= \arg\min_{\lambda} \sum_{q\in Q} Y.\end{aligned} \tag{5}$$

We modify the optimization object by introducing a loss function:

$$\lambda = \arg\min_{\lambda} \sum_{q\in Q} \sum_{I_i, I_j \in l^*(q)} f(d(q, I_i, \lambda) - d(q, I_j, \lambda)), \tag{6}$$

where $f(y)$ is a Huber penalty function that is everywhere-differentiable, as follows:

$$f(y) = \begin{cases} 0, & y \leq 0, \\ \dfrac{y^2}{2W}, & y \in (0, W], \\ y - \dfrac{W}{2}, & y > W. \end{cases} \tag{7}$$

To minimize the penalty mentioned in Eq. (6), the gradient descent algorithm can be adopted. Thus, $\frac{\partial f(d(q,I_i,\lambda)-d(q,I_j,\lambda))}{\partial\lambda} = f'(d(q, I_i, \lambda) - d(q, I_j, \lambda))(d'(q, I_i, \lambda) - d'(q, I_j, \lambda))$ should be calculated, and they can be easily obtained using Eq. (7) and Eq. (2).

The parameter learning algorithm is shown in Algorithm 1, where Q is the query set, N_q and R_q are dissimilar and similar characters with the query q, respectively, and $\eta = 0.001$.

4 Efficient matching

In this section, we propose two efficient matching methods, the keypoint-based method and the LSH-based method, and then propose a re-ranking method as a trade-off between search speed and retrieval quality by combining these two methods.

4.1 Keypoint-based matching

Considerable time is required to find the 'best' matching point for each point in shape matching. Although we reduce the search scope to 1/4 of the image area, the matching speed is still too slow. If we decrease the number of points, the matching

Algorithm 1 Parameter learning

```
Input: training samples $S = \{(I_i, I_j)_q | q \in Q, I_i \in N_q, I_j \in R_q\}$
Output: optimal parameter $\lambda$
Initialize $\lambda$ by $\lambda = 1.0$;
while there is significant change in $\lambda$ do
  Initialize loss = 0.0 and violation number = 0;
  for all $(I_i, I_j)_q \in S$ do
    if $d(I_i, q, \lambda) < d(I_j, q, \lambda)$ then
      $\lambda = \lambda - \eta f'(d(I_j, q, \lambda) - d(I_i, q, \lambda))(F_1(I_j, q) - F_2(I_j, q) - F_1(I_i, q) + F_2(I_i, q))$;
      Increase loss by $f(d(I_j, q, \lambda) - d(I_i, q, \lambda))$ according to Eq. (6);
      Increase the violation number by 1;
    end if
  end for
end while
```

speed can be significantly improved. Since two adjacent points have similar features and removing one point will not affect the shape similarity, an intuitive method can be used to sample points for some intervals, as shown in Fig. 6, a sample-based method, which has been used in our previous system (Zhuang *et al.*, 2004; 2005). After sampling, the average number of points of each character is reduced from 694.1966 to about 359.6924, but the matching speed is still too slow and we should filter more points.

Inspired by the SIFT descriptor (Lowe, 1999), the SC-HoG feature can be extracted at keypoints. After keypoint detection, we need only to extract the feature at about 45.0042 points for each character, which is much less than that of the sample-based method. Meanwhile, keypoints contain stroke endpoints and cross-points in the character (Fig. 7). These points play an important role in shape recognition.

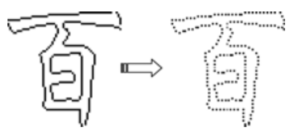


Fig. 6 Point sample

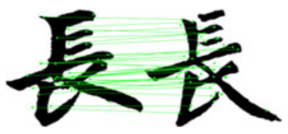


Fig. 7 Keypoints match

4.2 LSH-based matching

If we can bypass the point correspondence problem, the retrieval process can be greatly accelerated. Inspired by Dong *et al.* (2008), we also use LSH functions to hash global features and local features for buckets, and then create histograms by counting the buckets. Thus, the SC-HoG feature set can be mapped into two histograms (Fig. 8).

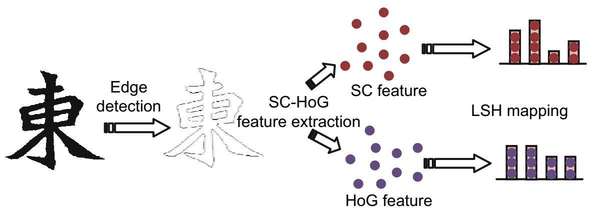


Fig. 8 Locality sensitive hashing (LSH) mapping procedure

We define the LSH function as

$$f(\boldsymbol{p}) = \left\lfloor \frac{\boldsymbol{A} \cdot \boldsymbol{p} + b}{W} \right\rfloor \bmod M, \tag{8}$$

where $\boldsymbol{p}$ is an SC or HoG feature vector, $\boldsymbol{A}$, b, and W are as defined in Dong *et al.* (2008), and M makes the hash value in range $[0, M)$. In this way, each point can obtain two hash values according to the SC feature and the HoG feature. When mapping all points into range $[0, M)$, we can create an M-dimensional vector. In practice, we should run the mapping procedure K times to obtain K independent random histograms and improve accuracy. Then the K histograms are concatenated to form the final KM-dimensional vector. Thus, the feature-set matching problem is converted into a high dimensional vector similarity calculation problem.

In our experiment, we set $W = 10, M = 20$, and $K = 50$. Thus, each character is represented by two 1000-dimensional vectors, and then the principal component analysis (PCA) approach is applied for dimensionality reduction.

4.3 Re-ranking

The LSH-based method is much faster, but its performance is not high enough (Section 5.4). To guarantee search speed and retrieval quality, we propose the re-ranking method. We obtain the top K characters with the LSH-based method first, and then re-rank their characters with the sample-based method (denoted as ReRank(L+S)) or keypoint-based method (denoted as ReRank(L+K)). We can filter dissimilar characters with the LSH-based method to save search time, and guarantee retrieval quality with the sample- or keypoint-based method.

In addition, a graph-based transduction algorithm (label propagation) (Bai *et al.*, 2010b) is applied to further improve the retrieval accuracy. In the algorithm, the affinity matrix is constructed for the top K similar characters first, and then the probabilistic transition matrix is defined through the affinity matrix row-wise normalization. Finally, the query label is propagated on the probabilistic transition matrix for computing the new similarity measure.

5 Experiments

5.1 Dataset

We have scanned some calligraphy books such as “Chinese Calligraphy Collections” with 600 dpi resolution in the China-US Million Book Digital Library Project (www.cadal.zju.edu.cn), and then built the calligraphic character dataset (called MBPSet) through preprocessing, which includes binarization, segmentation, and character scaling. In our work, we normalized the character image into 128×128 in pixels. Finally, the MBPSet consists of 1648 calligraphic characters with different styles and authors, comprising 106 Chinese characters. Fig. 9 shows the MBPSet with its number under each character, where some characters are more similar.

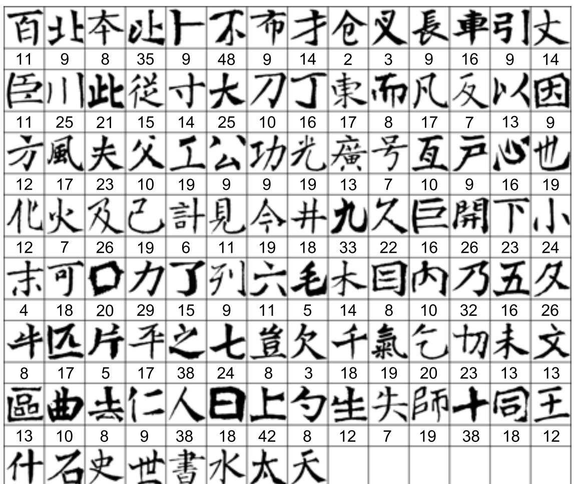

11 9 8 35 9 48 9 14 2 3 9 16 9 14
11 25 21 15 14 25 10 16 17 8 17 7 13 9
12 17 23 10 19 9 9 19 13 7 10 9 16 19
12 7 26 19 6 11 19 18 33 22 16 26 23 24
4 18 20 29 15 9 11 5 14 8 10 32 16 26
8 17 5 17 38 24 8 3 18 19 20 23 13 13
13 10 8 9 38 18 42 8 12 7 19 38 18 12
2 10 11 14 12 9 23 14

Fig. 9 Calligraphic character dataset—MBPSet

We also collected 80 Chinese characters in Microsoft Office to form a standard Chinese calligraphic dataset (STDSet), where each Chinese character has 12 writing styles, including SongTi, YanTi, KaiShu, LiShu, XingKai, ShuTi, YaoTi, YouYuan, Microsoft-YaHei, XinWei, FangSong, and XingShu. Thus, we collected 960 calligraphic characters with different styles in STDSet. Some examples of STDSet are shown in Fig. 10.

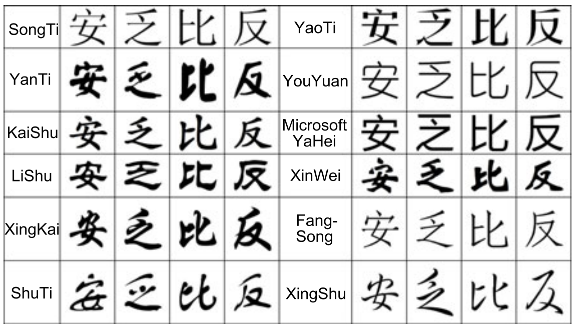

SongTi
YaoTi
YanTi
YouYuan
KaiShu
Microsoft YaHei
LiShu
XinWei
XingKai
Fang-Song
ShuTi
XingShu

Fig. 10 Some examples of the calligraphic character dataset from Microsoft Office—STDSet

5.2 Optimal parameter learning

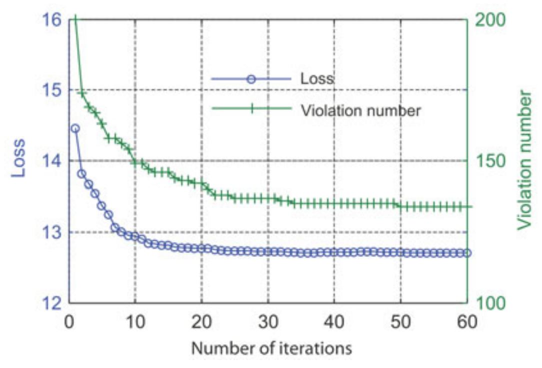
As mentioned earlier, we used a gradient descent algorithm to find the optimal parameter λ, and generated some training samples for parameter learning. To obtain some significant samples, we randomly selected 10 calligraphic characters as queries, and then retrieved top 40 characters for each query according to the SC-HoG descriptor. In the search result, the samples $\{(I_i, I_j)_q | q \in Q, I_i \in N_q, I_j \in R_q, d(q, I_i, \lambda) < d(q, I_j, \lambda), I_i, I_j \in l(q, \lambda)\}$ were collected, and finally 200 samples were obtained as the training dataset S.

Fig. 11a shows the loss (which is equal to $\sum_{q\in Q}\sum_{(I_i,I_j)_q\in S} f(d(I_j, q, \lambda) - d(I_i, q, \lambda)))$ and violation number ($= \sum_{q\in Q}\sum_{(I_i,I_j)_q\in S} \delta(d(I_j, q, \lambda) - d(I_i, q, \lambda)))$ changing along with the iterations for MBPSet, where $\delta(x) = 1$ when $x > 0$, or $\delta(x) = 0$. In the figure, we can see that the loss and violation number gradually converge to the minimal value.

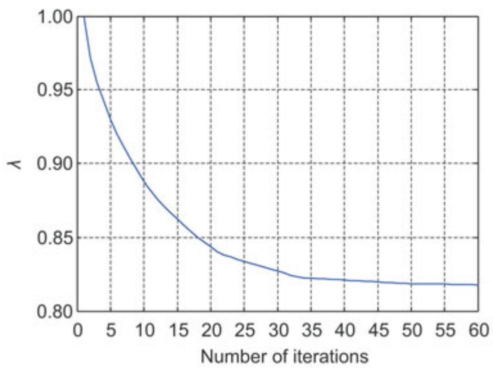
Fig. 11b shows the parameter λ calculated in each iteration in Algorithm 1, which also converges to a stable value gradually. The final value $\lambda = 0.818$ is the optimal parameter for MBPSet, denoted as λ^*. Similarly, we determined $\lambda^* = 0.84$ for STDSet.

5.3 Performance of the SC-HoG descriptor

We randomly selected 10 calligraphic characters as queries for evaluation in each dataset. Fig. 12 shows some retrieval examples in MBPSet using the sample-based method with different λ, where λ^* is the optimal parameter determined by the gradient descent algorithm. In the figure, the characters in

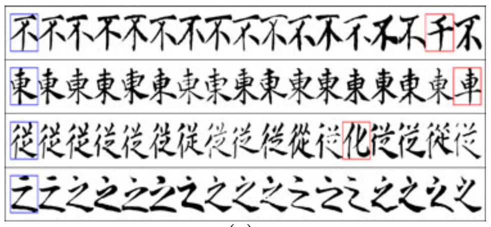


(a)

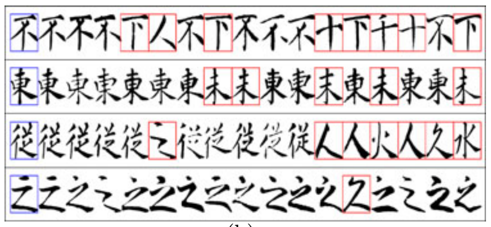


(b)

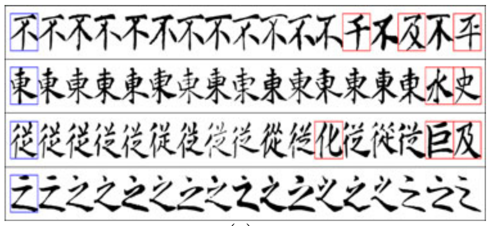
Fig. 11 Convergence of loss and violation number (a) and λ (b) in parameter learning for MBPSet



(a)



(b)



(c)

Fig. 12 Search results with different λ in MBPSet. (a) $\lambda = \lambda^*$, SC-HoG descriptor; (b) $\lambda = 0.0$, HoG descriptor; (c) $\lambda = 1.0$, SC descriptor. The characters in the first column are queries and the remaining characters are the search results ranked by similarity with the query, where only top 16 results are shown and the characters with boxes are incorrect

the first column are queries and the remaining characters are the search results ranked by similarity with the query, where only top 16 results are shown and the characters with boxes are incorrect. We can see that when $\lambda = \lambda^*$, the method reaches the best performance, for it integrates global and local features. The result with $\lambda = 1.0$ is better than that with $\lambda = 0.0$, which indicates that the global feature is more discriminative than the local feature. Although the search accuracy at $\lambda = 1.0$ is close to the result with $\lambda = \lambda^*$, some incorrect characters with $\lambda = \lambda^*$ are still more similar with queries. For example, in the second line of the results, the incorrect characters with $\lambda = \lambda^*$ are also more similar with the query.

We used precision and recall to measure the character retrieval performance, and they are defined as follows:

$$\text{precision} = \frac{\{\text{relevant}\} \cap \{\text{retrieved}\}}{\{\text{retrieved}\}}, \tag{9}$$

$$\text{recall} = \frac{\{\text{relevant}\} \cap \{\text{retrieved}\}}{\{\text{relevant}\}}, \tag{10}$$

where {relevant} denotes the query-relevant characters in the dataset and {retrieved} denotes the returned characters. Fig. 13 shows the average precision of different top K returned results (AP@K) under different parameter λ. From the figure, we can see that: (1) The optimal parameter $\lambda = \lambda^*$ can make the SC-HoG descriptor based character retrieval reach the maximal precision, and it also proves that the gradient descent algorithm indeed learned the optimal parameter. (2) Global and local feature integration could basically improve the retrieval performance such as when $\lambda = 0.3, 0.4, \ldots, 0.9$. However, if we select a parameter poorly, such as $\lambda = 0.1$, the performance will degrade, and it would even be worse than that of using only the global feature. (3) The global feature has more discriminability power than the local feature.

Furthermore, to verify the performance of the SC-HoG descriptor, we compared the SC-HoG descriptor with SFHM, MRHM, D-Shape, SC, and

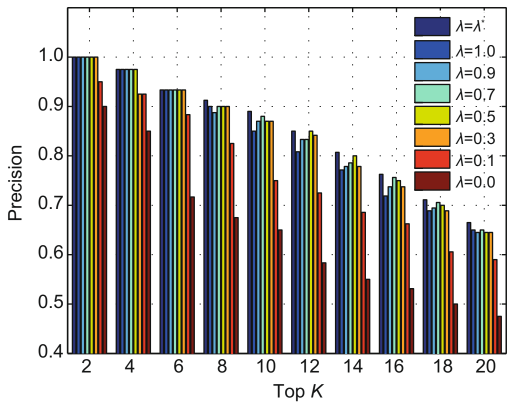


Fig. 13 AP@K under different λ in MBPSet

HoG descriptors, which can be used for calligraphic character retrieval, and the results are shown in Tables 1 and 2. From the tables, we can find that the SC-HoG descriptor performs the best for calligraphic character retrieval.

5.4 Searching performance and time cost comparison

To validate the two efficient matching methods, we compared their performance in terms of precision-recall curves (Fig. 14) and search time (Table 3). Keypoint- and LSH-based methods have a performance similar to that of the sample-based method, while their search speeds are about 83 and 4048 times faster than that of the sample-based method, respectively. When $\lambda = \lambda^*$, the methods reach peak performance, but using only the global feature ($\lambda = 1.0$) degrades the performance greatly. This observation encourages us to integrate the global feature and the local feature.

To verify the performance of the keypoint-based method, we also compared the keypoint-based method with the random-based method, which randomly samples the same number of points as the keypoint-based method, and the results are shown in Fig. 15. From the figure, we see that the keypoint-based method is much better than the random-based method, which indicates that the keypoints are the representative points in the character.

5.5 Re-ranking performance

Fig. 16 shows the performance of the re-ranking method with different K. The performance of the LSH-based method is much close to that of the sample- or keypoint-based method through

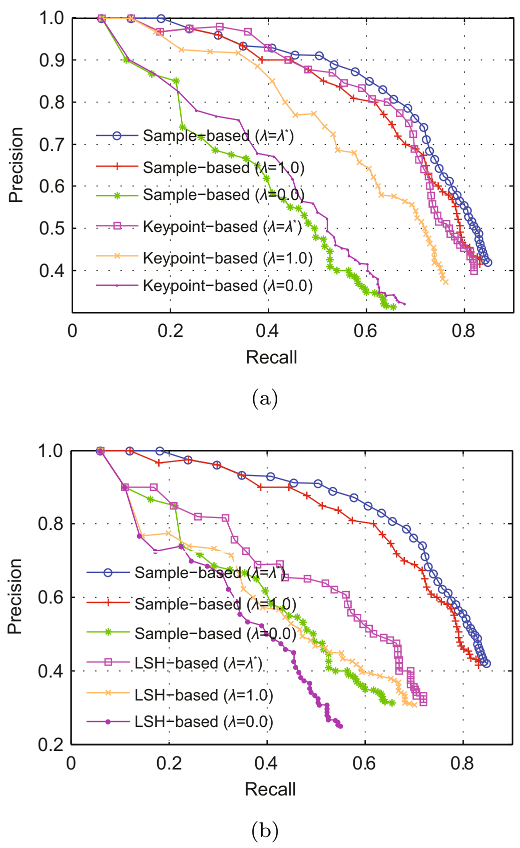


Fig. 14 Precision vs. recall with different methods in MBPSet. (a) Keypoint-based vs. sample-based; (b) LSH-based vs. sample-based

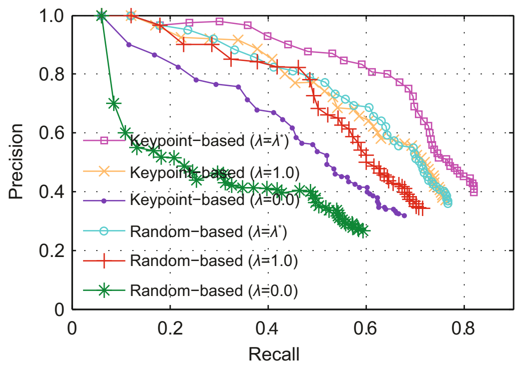


Fig. 15 Keypoint-based method vs. random-based method in MBPSet

re-ranking. Specifically, when the recall is small, their precisions are the same. This indicates that LSH can filter dissimilar characters efficiently.

Table 4 shows the search time of re-ranking methods with different K. Even when $K = 200$, ReRank(L+S) is about 8.25 times faster than the sample-based method, and ReRank(L+K) is about 6.95 times faster than the keypoint-based method.

Table 1 Average precision of different top K returned results (AP@K) of different descriptors for MBPSet

Descriptor	AP@K				
	$K=4$	$K=8$	$K=12$	$K=16$	$K=20$
SFHM	0.67 ± 0.24	0.46 ± 0.22	0.38 ± 0.20	0.29 ± 0.16	0.27 ± 0.14
MRHM	0.67 ± 0.21	0.51 ± 0.23	0.41 ± 0.21	0.38 ± 0.19	0.33 ± 0.17
D-Shape	0.85 ± 0.24	0.78 ± 0.26	0.67 ± 0.22	0.54 ± 0.16	0.47 ± 0.14
SC	0.97 ± 0.08	0.9 ± 0.15	0.81 ± 0.17	0.72 ± 0.17	0.65 ± 0.19
HoG	0.85 ± 0.21	0.68 ± 0.21	0.58 ± 0.20	0.53 ± 0.20	0.47 ± 0.19
SC-HoG	$\mathbf{0.97}\pm0.08$	$\mathbf{0.91}\pm0.19$	$\mathbf{0.85}\pm0.18$	$\mathbf{0.76}\pm0.21$	$\mathbf{0.67}\pm0.19$
SC-HoG+LP	$\mathbf{0.98}\pm0.08$	$\mathbf{0.96}\pm0.12$	$\mathbf{0.89}\pm0.20$	$\mathbf{0.83}\pm0.20$	$\mathbf{0.75}\pm0.20$

Table 2 Average precision of different top K returned results (AP@K) of different descriptors for STDSet

Descriptor	AP@K				
	$K=4$	$K=8$	$K=12$	$K=16$	$K=20$
SFHM	0.35 ± 0.17	0.24 ± 0.15	0.19 ± 0.11	0.16 ± 0.10	0.13 ± 0.08
MRHM	0.47 ± 0.22	0.28 ± 0.13	0.20 ± 0.11	0.17 ± 0.08	0.16 ± 0.06
D-Shape	0.70 ± 0.26	0.50 ± 0.18	0.40 ± 0.14	0.32 ± 0.10	0.27 ± 0.08
SC	0.66 ± 0.25	0.46 ± 0.17	0.37 ± 0.16	0.30 ± 0.12	0.27 ± 0.09
HoG	0.61 ± 0.26	0.45 ± 0.16	0.38 ± 0.14	0.33 ± 0.11	0.29 ± 0.09
SC-HoG	$\mathbf{0.73}\pm0.26$	$\mathbf{0.52}\pm0.19$	$\mathbf{0.42}\pm0.16$	$\mathbf{0.35}\pm0.13$	$\mathbf{0.30}\pm0.10$
SC-HoG+LP	$\mathbf{0.79}\pm0.23$	$\mathbf{0.56}\pm0.20$	$\mathbf{0.43}\pm0.15$	$\mathbf{0.37}\pm0.12$	$\mathbf{0.31}\pm0.11$

Table 3 Search time with different methods in MBPSet

Method	Average search time (s)	Average matching time (s)
Sample-based	9084.4	5.4905
Keypoint-based	109.58	0.0665
LSH-based	2.2439	0.0014

Table 4 Search time of re-ranking methods with different K in MBPSet

Method	Search time (s)				
	$K=200$	100	80	60	40
ReRank(L+S)	1100.52	551.37	441.55	331.71	221.87
ReRank(L+K)	15.76	9.14	7.61	6.29	4.93

The precisions of ReRank(L+S) and ReRank(L+K) are similar (Fig. 17), but ReRank(L+K) needs only about 10 s for retrieval, which makes the method more practical.

Finally, we show the results of the graph transduction algorithm in the SC-HoG+LP column in Tables 1 and 2, where the parameters were determined empirically including $K=80$, the neighborhood size $b=14$, and the adaptive parameter $\alpha=0.3$ for MBPSet, and $K=80$, $b=14$, and $\alpha=0.4$ for STDSet. From the tables, we can see that the graph transduction algorithm indeed improves the character retrieval accuracy.

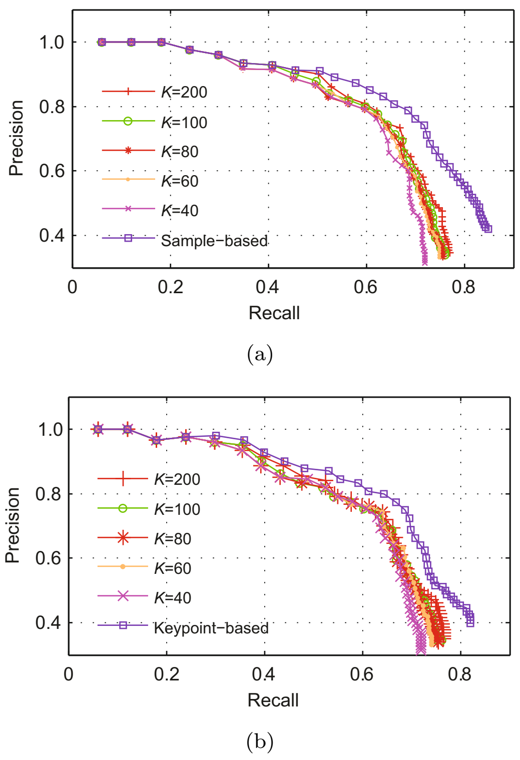


Fig. 16 Performance of the re-ranking methods with different K in MBPSet. (a) ReRank(L+S) vs. sample-based; (b) ReRank(L+K) vs. keypoint-based

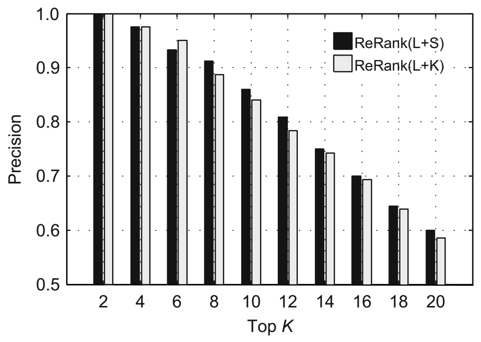


Fig. 17 Performance with different re-ranking methods in MBPSet

6 Conclusions

In this paper, we propose a novel SC-HoG descriptor for shape matching and two efficient matching methods. Experiments on calligraphic character datasets show that the proposed descriptor is effective for calligraphic character retrieval, and the two matching methods are quick. In addition, the reranking method makes our approach more practical.

The search time depends on two factors: match speed for every two characters and the number of characters that need to match. Our two efficient matching methods focus on how to accelerate the match, but all the characters in the dataset should still be involved for the matching, which slows down the search greatly. In the future, we will use high-dimensional index technologies such as iDistance (Jagadish *et al.*, 2005), NB-tree (Fonseca and Jorge, 2003), and VA-file (Weber *et al.*, 1998) to speed up the search by reducing the number of matches between the query and the characters in the dataset.

Furthermore, we will try other graph transduction algorithms (Yang *et al.*, 2009) to improve the accuracy of calligraphic character retrieval and the co-transduction algorithm (Bai *et al.*, 2010a) to combine different shape descriptors for calligraphic character retrieval.

References

Aslan, C., Tari, S., 2005. An Axis-Based Representation for Recognition. 10th IEEE Int. Conf. on Computer Vision, **2**:1339-1346. [doi:10.1109/ICCV.2005.32]

Bai, X., Latecki, L.J., 2008. Path similarity skeleton graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**(7):1282-1292. [doi:10.1109/TPAMI.2007.70769]

Bai, X., Wang, B., Wang, X., Liu, W., Tu, Z., 2010a. Co-transduction for Shape Retrieval. Proc. 11th European Conf. on Computer Vision: Part III, p.328-341.

Bai, X., Yang, X., Latecki, L.J., Liu, W., Tu, Z., 2010b. Learning context-sensitive shape similarity by graph transduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, **32**(5):861-874. [doi:10.1109/TPAMI.2009.85]

Belongie, S., Malik, J., Puzicha, J., 2000. Shape Context: a New Descriptor for Shape Matching and Object Recognition. NIPS, p.831-837.

Bosch, A., Zisserman, A., Munoz, X., 2007. Representing Shape with a Spatial Pyramid Kernel. Proc. 6th ACM Int. Conf. on Image and Video Retrieval, p.401-408.

Brandt, S., Laaksonen, J., Oja, E., 2002. Statistical shape features for content-based image retrieval. *J. Math. Imag. Vis.*, **17**(2):187-198. [doi:10.1023/A:1020689721567]

Chou, C.H., Wu, C.S., Han, C.C., 2005. An Interactive Grading and Learning System for Chinese Calligraphy. IEEE Int. Conf. on Electro Information Technology, p.1-6. [doi:10.1109/EIT.2005.1626973]

Dalal, N., Triggs, B., Rhone-Alps, I., Montbonnot, F., 2005. Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, **1**:886-893.

Dong, W., Wang, Z., Charikar, M., Li, K., 2008. Efficiently Matching Sets of Features with Random Histograms. Proc. 16th ACM Int. Conf. on Multimedia, p.179-188. [doi:10.1145/1459359.1459384]

Fonseca, M.J., Jorge, J.A., 2003. NB-Tree: an Indexing Structure for Content-Based Retrieval in Large Databases. Proc. 8th Int. Conf. on Database Systems for Advanced Applications, p.267-274.

Fornes, A., Escalera, S., Llados, J., Valveny, E., 2010. Symbol Classification Using Dynamic Aligned Shape Descriptor. Proc. 20th Int. Conf. on Pattern Recognition, p.1957-1960.

Grauman, K., Darrell, T., 2005. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. 10th IEEE Int. Conf. on Computer Vision, **1**:1458-1465. [doi:10.1109/ICCV.2005.239]

Jagadish, H.V., Ooi, B.C., Tan, K.L., Yu, C., Zhang, R., 2005. iDistance: an adaptive B^+-tree based indexing method for nearest neighbor search. *ACM Trans. Database Syst.*, **30**(2):364-397. [doi:10.1145/1071610.1071612]

Kortgen, M., Park, G.J., Novotni, M., Klein, R., 2003. 3D Shape Matching with 3D Shape Contexts. 7th Central European Seminar on Computer Graphics, p.55-66.

Ling, H.B., Jacobs, D.W., 2007. Shape classification using the inner-distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, **29**(2):286-299. [doi:10.1109/TPAMI.2007.41]

Lowe, D.G., 1999. Object Recognition from Local Scale-Invariant Features. Proc. 7th IEEE Int. Conf. on Computer Vision, **2**:1150-1157. [doi:10.1109/ICCV.1999.790410]

Lu, W., Zhuang, Y., Wu, J., 2009. Discovering calligraphy style relationships by supervised learning weighted random walk model. *Multimedia Syst.*, **15**(4):221-242. [doi:10.1007/s00530-008-0151-z]

Luo, B., Robles-Kelly, A., Torsello, A., Wilson, R.C., Hancock, E.R., 2001. Discovering Shape Categories by Clustering Shock Trees. Proc. 9th Int. Conf. on Computer Analysis of Images and Patterns, p.152-160.

Mori, G., Malik, J., 2003. Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, **1**:134-141. [doi:10.1109/CVPR.2003.1211347]

Mori, G., Belongie, S., Malik, J., 2005. Efficient shape matching using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, **27**(11):1832-1837. [doi:10.1109/TPAMI.2005.220]

Mortensen, E.N., Deng, H., Shapiro, L., 2005. A SIFT Descriptor with Global Context. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, **1**:184-190.

Sebastian, T.B., Klein, P.N., Kimia, B.B., Center, G.E.G.R., Schenectady, N.Y., 2004. Recognition of shapes by editing their shock graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(5):550-571. [doi:10.1109/TPAMI.2004.1273924]

Suard, F., Rakotomamonjy, A., Bensrhair, A., 2006. Object Categorization Using Kernels Combining Graphs and Histograms of Gradients. Proc. ICIAR, p.23-34.

Torsello, A., Hancock, E.R., 2004. A skeletal measure of 2D shape similarity. *Comput. Vis. Image Underst.*, **95**(1):1-29. [doi:10.1016/j.cviu.2004.03.006]

Torsello, A., Robles-Kelly, A., Hancock, E.R., 2007. Discovering shape classes using tree edit-distance and pairwise clustering. *Int. J. Comput. Vis.*, **72**(3):259-285. [doi:10.1007/s11263-006-8929-y]

van Eede, M., Macrini, D., Telea, A., Sminchisescu, C., Dickinson, S.S., 2006. Canonical Skeletons for Shape Matching. 18th Int. Conf. on Pattern Recognition, **2**:64-69. [doi:10.1109/ICPR.2006.354]

Wang, S.Z., Lee, H.J., 2001. Dual-Binarization and Anisotropic Diffusion of Chinese Characters in Calligraphy Documents. Proc. 6th Int. Conf. on Document Analysis and Recognition, p.271-275. [doi:10.1109/ICDAR.2001.953797]

Wang, T.T., Lu, T., Liu, W.Y., 2010. Robust Shape Retrieval Through a Novel Statistical Descriptor. Proc. 11th Pacific Rim Conf. on Advances in Multimedia Information Processing: Part I, p.330-337.

Weber, R., Schek, H.J., Blott, S., 1998. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. Proc. Int. Conf. on Very Large Data Bases, p.194-205.

Wong, S.T.S., Leung, H., Ip, H.H.S., 2006. Brush Writing Style Classification from Individual Chinese Characters. Proc. 18th Int. Conf. on Pattern Recognition, p.884-887. [doi:10.1109/ICPR.2006.343]

Wong, S.T.S., Leung, H., Ip, H.H.S., 2008. Model-based analysis of Chinese calligraphy images. *Comput. Vis. Image Underst.*, **109**(1):69-85. [doi:10.1016/j.cviu.2007.03.001]

Wu, Y.F., Zhuang, Y.T., Pan, Y.H., Wu, J.Q., 2006. Web Based Chinese Calligraphy Learning with 3-D Visualization Method. IEEE Int. Conf. on Multimedia and Expo, p.2073-2076. [doi:10.1109/ICME.2006.262642]

Xu, S., Lau, F.C.M., Cheung, W.K., Pan, Y., 2005. Automatic generation of artistic Chinese calligraphy. *IEEE Intell. Syst. Appl.*, **20**(3):32-39. [doi:10.1109/MIS.2005.41]

Xu, S., Jiang, H., Lau, F.C.M., Pan, Y., 2007. An Intelligent System for Chinese Calligraphy. Proc. National Conf. on Artificial Intelligence, p.1578-1583.

Yang, X., Koknar-Tezel, S., Latecki, L.J., 2009. Locally Constrained Diffusion Process on Locally Densified Distance Spaces with Applications to Shape Retrieval. IEEE Conf. on Computer Vision and Pattern Recognition, p.357-364. [doi:10.1109/CVPR.2009.5206844]

Yankov, D., Keogh, E., Wei, L., Xi, X.P., Hodges, W., 2008. Fast best-match shape searching in rotation-invariant metric spaces. *IEEE Trans. Multimedia,* **10**(2):230-239. [doi:10.1109/TMM.2007.911824]

Yu, J.H., Peng, Q.S., 2005. Realistic synthesis of cao shu of Chinese calligraphy. *Comput. Graph.*, **29**(1):145-153. [doi:10.1016/j.cag.2004.11.013]

Yu, K., Wu, J., Zhuang, Y., 2008. Skeleton-Based Recognition of Chinese Calligraphic Character Image. Proc. 9th Pacific Rim Conf. on Multimedia: Advances in Multimedia Information Processing, p.228-237.

Zhang, J., Liu, W.Y., 2009. A Pixel-Level Statistical Structural Descriptor for Shape Measure and Recognition. Proc. 10th Int. Conf. on Document Analysis and Recognition, p.386-390. [doi:10.1109/ICDAR.2009.175]

Zhang, J.S., Yu, J.H., Mao, G.H., Ye, X.Z., 2006. Denoising of Chinese calligraphy tablet images based on run-length statistics and structure characteristic of character strokes. *J. Zhejiang Univ.-Sci. A*, **7**(7):1178-1186. [doi:10.1631/jzus.2006.A1178]

Zhang, X.F., Zhuang, Y.T., 2007. Visual Verification of Historical Chinese Calligraphy Works. Proc. 13th Int. Multimedia Modeling Conf., p.354-363.

Zhang, X.F., Zhuang, Y.T., Wu, J.Q., Wu, F., 2007. Hierarchical approximate matching for retrieval of Chinese historical calligraphy character. *J. Comput. Sci. Technol.*, **22**(4):633-640. [doi:10.1007/s11390-007-9077-8]

Zhang, Z., Wu, J., Yu, K., 2010. Chinese Calligraphy Specific Style Rendering System. Proc. 10th Annual Joint Conf. on Digital Libraries, p.99-108. [doi:10.1145/1816123.1816138]

Zhu, Q., Wang, L., Wu, Y., Shi, J., 2008. Contour Context Selection for Object Detection: a Set-to-Set Contour Matching Approach. Proc. 10th European Conf. on Computer Vision, p.774-787.

Zhuang, Y., Zhang, X., Wu, J., Lu, X., 2004. Retrieval of Chinese Calligraphic Character Image. Proc. Pacific Rim Conf. on Multimedia, p.17-24.

Zhuang, Y., Zhang, X., Lu, W., Wu, F., 2005. Web-based Chinese calligraphy retrieval and learning system. *LNCS*, **3583**:186-196. [doi:10.1007/11528043_18]

Zhuang, Y., Zhuang, Y.T., Li, Q., Chen, L., 2007. Interactive high-dimensional index for large Chinese calligraphic character databases. *ACM Trans. Asian Lang. Inform. Process.*, **6**(2):54-85. [doi:10.1145/1282080.1282083]

Zhuang, Y., Lu, W., Wu, J., 2009. Latent style model: discovering writing styles for calligraphy works. *J. Vis. Commun. Image Represent.*, **20**(2):84-96. [doi:10.1016/j.jvcir.2008.11.007]