

Towards Interactive Indexing for Large Chinese Calligraphic Character Databases

Yi Zhuang Yueting Zhuang
College of Computer Science
Zhejiang University
Hangzhou, P.R.China
{zhuangyi, yzhuang}@cs.zju.edu.cn

Qing Li
Department of Computer Science
City University of Hong Kong
Kowloon, HKSAR, P.R.China
itqli@cityu.edu.hk

Lei Chen
Department of Computer Science
HKUST
Kowloon, HKSAR, P.R.China
leichen@cs.ust.hk

ABSTRACT

In this paper, based on a novel shape-similarity-based retrieval method, we propose an interactive partial-distance-map (PDM)-based high-dimensional indexing scheme to speed up the retrieval performance of the large Chinese calligraphic character databases. Specifically, we use the approximate minimal bounding hypersphere of query character to search the PDM and utilize the users' relevance feedback to refine the search process. We conduct comprehensive experiments to testify the efficiency and effectiveness of the proposed method.

Categories and Subject Descriptors: H.2.8 [DATABASE APPLICATIONS]: Database Applications – *Image Databases*

General Terms: Algorithm, Performance

Keywords: Chinese calligraphic character, high-dimensional index

1. INTRODUCTION

The large amount of Chinese calligraphic scripts in existence is a valuable part of the Chinese cultural heritage. Although these scripts can be available through public libraries or Internet, they can hardly be retrieved by optical character recognition (OCR) which performs well only on machine printed characters against clean background. No effective techniques can support the retrieval of Chinese calligraphic characters written in different styles due to their complexity, deformation and degradation.

Moreover, so far, no efficient techniques have been proposed to retrieve and index large Chinese calligraphic character databases. In essence, the indexing issue of Chinese calligraphic character belongs to the category of high-dimensional data indexing when we represent each character as a vector. With respect to the high-dimensional indexing issue, considerable research works have been done [1]. Unfortunately, these existing high-dimensional indexing methods can not be directly applied to the Chinese calligraphic characters due to their unique characteristics:

- To depict each Chinese calligraphic character, a set of contour points is extracted from each Chinese calligraphic character image [1]. Due to the complexity of Chinese characters, the number of these contour points are usually very large (in general, above 150 dimensions). As a consequence, the dimensionality of the character's vector representation is very high and conventional multi-dimensional indexing techniques can not be used to index due to the "curse of dimensionality"[2].
- The number of contour points of each character is different

from each other owing to their shape's complexity. Thus, for different characters, the dimensionalities of their representative vectors may vary. Many existing high-dimensional indexing schemes assume that the indexed vectors have fixed dimensionality.

In this paper, we address the efficient issues of retrieving Chinese calligraphic character. We propose an interactive high-dimensional indexing scheme based on *partial distance-map* (PDM), which is specifically designed for indexing the large Chinese calligraphic characters.

2. INTERACTIVE PARTIAL DISTANCE MAP

The key problem in Chinese calligraphic characters retrieval is the matching of similar isolated characters. In this paper, we search the character using the *Approximate Point Context* (APC)-based retrieval method [1]. In order to speed up the retrieval efficiency, we present a novel interactive high-dimensional indexing technique, called the *Partial Distance Map* (PDM). PDM is designed to establish the link between semantic-level concepts of characters and low-level shape features through the user relevance feedback, where a user only needs to mark which characters he or she thinks are similar to the query character. During the retrieval process, the PDM and its *Pruning Distance Table* (PDT) are updated dynamically via the relevance feedback to correctly reflect the users' query needs.

In PDM, each character is regarded as a reference one, the distances between the character and its neighboring characters are pre-calculated to generate a partial distance map with the constraint of the pruning distance (*PD*) which is set through user relevance feedback.

DEFINITION 1. A *Partial Distance Map* (PDM for short) is an adjacency list where $d_{ij} \in \text{PDM}$ and d_{ij} refers to the distance between the i -th character and its j -th nearest neighboring character.

The basic idea of the pruning distance table is to record and update the *PD* of each character via user relevance feedback so as to adjust the PDM dynamically.

DEFINITION 2. The *Pruning Distance Table* (PDT for short) is defined as a sequence of pairs which contains the corresponding *PDs* of the different characters, formally denoted as :

$$\text{PDT} ::= \langle \langle 1, PD(V_1) \rangle, \langle 2, PD(V_2) \rangle, \dots, \langle n, PD(V_n) \rangle \rangle \quad (1)$$

where $PD(V_i)$ refers to the pruning distance of the i -th character.

3. PSEUDO k -NN SEARCH

In this section, we present the hyper-centre relocation (*HCR*), to find 1-NN character of the query one, and pseudo k -NN to support PDM-based k -NN search with previously retrieved 1-NN result.

3.1 Centre Relocation of Hypersphere

Approximate Minimal Bounding Hypersphere – Given a query hypersphere $\Theta(V_q, r)$, the approximate minimal bounding hypersphere (*AMBH*) of it is a new hypersphere $\Theta(V_p, R)$ where V_p is the 1-NN character of V_q and $R = VQR(V_p)$. That is to say, $AMBH(V_q, r) = \Theta(V_p, R)$ either *contains* or *approximates* to $\Theta(V_q, r)$.

Clustering Characters – In order to further reduce the search region, the characters in Ω are first grouped into T clusters using agglomerative hierarchical clustering algorithm, e.g., BIRCH[3], the result of which is saved in an auxiliary structure. For a cluster $C_i, i \in [1, T]$, we randomly select a character in C_i as the centroid, O_i of this cluster, except the character at the boundary of this cluster. Thus we model a cluster as a tightly bounded hypersphere described by its *centroid* and *radius*.

DEFINITION 3 (CLUSTER RADIUS). Given a cluster C_i , the distance between O_i and the character which is farthest to O_i is defined as the cluster radius of C_i , denoted as CR_i .

Given a cluster C_i , the cluster hypersphere of it is denoted as $\Theta(O_i, CR_i)$, which is represented as a dash circle in Figure 8, where O_i is the centroid of cluster C_i , CR_i is the cluster radius.

DEFINITION 4 (CENTROID DISTANCE). Given a character V_i , its centroid distance is defined as the distance between V_i and O_j , the centre of cluster that V_i belongs to:

$$CD(V_i) = d(V_i, O_j) \quad (2)$$

AMBH is proposed to approximately represent the query hypersphere. We now present how to quickly get the new center, by using the *DDM*-based approach consisting of the uniform-start-distance and centroid-distance-based methods.

DEFINITION 5 (START DISTANCE). Given a character V_i , the Start-Distance (*SD* for short) of it is the distance between character V_i and character V_o , formally defined as:

$$SD(V_i) = d(V_i, V_o) \quad (3)$$

where $d(V_i, V_o)$ is the distance between V_i and V_o , namely. The dimensionality of V_i is the same to that of V_o and the two-dimensional coordinate values of each point in V_i is $\langle 0, 0 \rangle$.

DEFINITION 6 (UNIFORM START-DISTANCE). Given a characters V_i with dimensionality d_i , the Uniform Start-Distance (*USD* for short) of V_i are formally defined as:

$$USD(V_i) = \frac{SD(V_i)}{d_i} \times D \quad (4)$$

where d_i is the dimensionality of V_i , D refers to the uniform dimensionality and it satisfies $D \leq d_i, i \in [1, n]$.

For a character V_i in a cluster, it could be represented as a four-tuple:

$$V_i ::= \langle i, Cid, USD, CD \rangle \quad (5)$$

where i refers to the i -th character and Cid is the ID of the cluster V_i belongs to. Then the *USD* and the *CD* of V_i are combined to get the index key of V_i which is shown as follows:

$$key(V_i) = c * \lfloor CD(V_i), \theta \rfloor + \frac{USD(V_i)}{MAX_USD} \quad (6)$$

where $\lfloor CD(V_i), \theta \rfloor$ denotes a rounded $CD(V_i)$ value with θ decimal places and $\theta = \{1, 2, 3, \dots\}$. The c is a large constant which is used to linearly stretch the $\lfloor CD(V_i), \theta \rfloor$ to an integer. MAX_USD is also a constant which should be set large enough to normalize the value of $USD(V_i)$ into the range of $[0, 1]$ via division by MAX_USD . Thus, it is guaranteed that the search range of *USD* and *CD* would not be overlapping. Given a character V_i , the Δ of V_i is defined as the distance between V_i and its 1-NN character, i.e., $\Delta(V_i) = d(V_i, 1NN(V_i))$, where $1NN(V_i)$ refers to the 1-NN character of V_i .

3.2 Pseudo k -NN Search Algorithm

According to the query rationale of PDM, given a query character V_q , due to the introduction of relevance feedback process, once k is set by a relatively large value, the k -NN search of V_q may not surely guarantee to return k nearest neighbor characters since the number of semantically identical characters to V_q is limited in the database, maybe less than k . Therefore, it is called pseudo k -NN search (*PK-NN*), which is performed in two steps: first, when a user submits a query character V_q , the procedure of hyper centre relocation is invoked to return its 1-NN, V_p . Secondly, the search starts with a small radius, and step by step, the radius is increased to form a bigger query sphere iteratively. Once the number of candidate characters is larger than k , the $(|S| - k)$ characters which are farthest to the query one are identified and removed from S . In this way, just the k nearest neighbor characters of V_p is returned.

4. EXPERIMENTAL RESULTS

We present an extensive performance study to evaluate the effectiveness of PDM and compare it to the following competitive techniques: *iDistance* and *NB-Tree*. To verify the effectiveness of PDM, we used the Chinese Calligraphic characters image data from *China-America Academic Digital Library Project* [4] as the experimental data which contains a set of contour point features extracted from the 12,000 characters images in which each feature point is composed of $\langle x \text{ axis}, y \text{ axis} \rangle$. In our evaluation, we use the number of page accesses and the total response time as the performance metric. The results are reported in Figure 1.

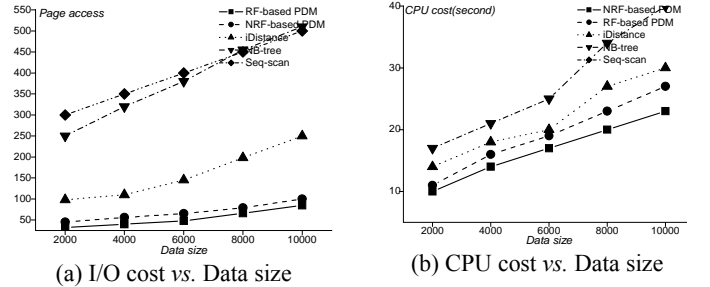


Figure 1: Performance Comparison

Figure 1 reveals that PDM is superior to *iDistance* and *NB-tree* in terms of both I/O and CPU cost.

6. ACKNOWLEDGEMENTS

This research is supported by the China-America Academic Digital Library Project (<http://www.cadal.zju.edu.cn>), and the key program of National Natural Science Foundation of China (No.60533090) and Hong Kong RGC grants DAG05/06.EG03.

7. REFERENCES

- [1] Zhuang, Y.T., Zhang, X.F., et al. Retrieval of Chinese Calligraphic Character Image. In *PCM'04*, 17-24. 2004.
- [2] C. Böhm, S. Berchtold, D. Keim. Searching in High-dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases, *ACM Computing Surveys* 2001.
- [3] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An efficient data clustering method for very large databases. In *ACM SIGMOD'96*, 103-114. 1996.
- [4] THE CADAL PROJECT, <http://www.cadal.zju.edu.cn>, 2005.