Locality Sensitive Pseudo-Code for Document Images

Kengo Terasawa and Yuzuru Tanaka
Meme Media Laboratory, Hokkaido University
N-13, W-8, Sapporo, 060–8628, Japan
{terasawa,tanaka}@meme.hokudai.ac.jp

Abstract

In this paper, we propose a novel scheme for representing character string images in the scanned document. We converted conventional multi-dimensional descriptors into pseudo-codes which have a property that: if two vectors are near in the original space then encoded pseudo-codes are 'semiequivalent' with high probability. For this conversion, we combined Locality Sensitive Hashing (LSH) indices and at the same time we also developed a new family of LSH functions that is superior to earlier ones when all vectors are constrained to lie on the surface of the unit sphere. Word spotting based on our pseudo-code becomes faster than multi-dimensional descriptor-based method while it scarcely degrades the accuracy.

1. Introduction

A sliding window approach has been widely used for document image processing. In such approaches, each sub-images clipped by the narrow window are converted to multi-dimensional descriptors and the sequence of those descriptors is sent to further processing, e.g., recognition, word spotting, document retrieval, etc.

The multi-dimensional descriptors, also called feature vectors, are computed in various ways and sometimes have very high dimensionality. It is well known that high dimensionality not only causes the increase of computational cost both in space and in time but also makes it more difficult to develop efficient searching algorithm due to "the curse of dimensionality."

In this paper, we propose a novel method for converting multi-dimensional descriptors into encoded representation. Original descriptors, that are high dimensional feature vectors with continuous components, are encoded to a set of integers without significant loss of information. Although it does not represent a specified letter, the obtained set of integers can be used as the character code for the purpose of substring detection and keyword extraction, hence it is worth to be called 'pseudo-code.'

Pseudo-code representation of string images enables us to use classical string matching techniques that make some processing faster than original vector-based method. One of the examples will be described in Section 5 of this paper, where we solve word spotting problem in linear time by means of classical inexact matching algorithm based on edit distance.

Our pseudo-code consists of multiple attributes and two codes are regarded to be *semiequivalent* if at least one of their attributes takes the same value. As the attribute value, we used the indices of Locality Sensitive Hashing (LSH) [1, 2, 3], which is a famous probabilistic approximate nearest neighbor method. We also designed a new family of LSH functions that is superior to previously proposed ones when all vectors are constrained to lie on the surface of the unit sphere. The advantage of our approach is experimentally confirmed in Section 5.

2. Related Work

As mentioned before, sliding window technique and multi-dimensional descriptors are widely used. Rath and Manmatha [4] used sliding window for word spotting for historical documents. Marti and Bunke [5] and Zimmermann and Bunke [6] also used sliding window for word recognition. Fink and Plötz [7] tested appearance-based features and compared it with heuristic features. Terasawa et al. [8] developed principal component analysis-based descriptors and they also developed gradient distribution features [9] for word spotting for Japanese handwritten documents.

The idea of vector encoding for document image is found in Transmedia Machine [10, 11]. They used "ambiguous incomplete encoding" where the same character might be expressed in various codes. In searching, they exhaustively examined all possible combination of codes using exact set matching algorithm of Aho-Corasick. The drawback of their method is that they needed to know every possible expression for all alphabets in advance. C. L. Tan et al. [12]

Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)
0-7695-2822-8/07 $25.00 © 2007 IEEE

classified every character image without OCR, and executed n-gram analysis for the sequence of the class and showed that this was enough for document retrieval. Marinai et al. [13] used SOM-based clustering for character-like coding and used string edit distance for word retrieval.

3. Locality Sensitive Pseudo-Code (LSPC)

In converting the vector into pseudo-code, it is desirable that the encoding system has the following property: if two vectors u, v are close in distance then the encoded codes correspond with high probability and if they are far in distance then the encoded codes correspond with small probability. One possible idea is to employ vector quantization method such as k-means. However, we adopted another idea that enables the correspondence probability more sensitive to the distance than simpler vector quantization method. In our approach, encoded pseudo-code has multiple attributes and two codes are considered to be *semiequivalent* if at least one of their attributes are equivalent. Since the pseudo-code produced by this idea is locality sensitive, we named it "Locality Sensitive Pseudo-Code (LSPC)."

In encoding LSPC, we use Locality Sensitive Hashing (LSH) [1, 2, 3] scheme, which is a probabilistic algorithm for approximate nearest-neighbor-search problem that runs significantly faster than other existing method especially in high dimensional spaces.

LSH is based on the family of functions called LSH family defined as follows:

Definition 1. For a domain S of the vector set, a family $\mathcal{H} = \{h : S \to U\}$ is called (r_1, r_2, p_1, p_2)-sensitive if for any $v, q \in S$,

$$\text{if } d(v,q) \le r_1 \text{ then } \Pr_{\mathcal{H}}[h(q) = h(v)] \ge p_1$$
$$\text{if } d(v,q) > r_2 \text{ then } \Pr_{\mathcal{H}}[h(q) = h(v)] \le p_2$$

where $d(v, q)$ is the distance between v and q, and it has to satisfy inequalities $p_1 > p_2$ and $r_1 < r_2$.

Each element of LSH family $\mathcal{H}$ is a hash function in the sense that it maps multi-dimensional vector $v \in \mathbb{R}^n$ into an integer $h(v) \in \mathbb{N}$. The basic idea of LSH is to hash every vector into hash tables using a hash function randomly chosen from LSH family. In finding nearest neighbor, LSH scans only the vectors that have the same hash index as the query vector.

In order to amplify the difference of collision probabilities, LSH takes direct product of hash functions, i.e.,

$$g(p) = \{\ h_1(p), h_2(p), \ldots, h_k(p)\ \} \tag{1}$$

where h_i is a (r_1, r_2, p_1, p_2)-sensitive hash function randomly chosen from the LSH family $\mathcal{H}$. For nearest-neighbor-search, LSH scans only the vectors whose hash indices $g(p)$ is the same as that of the query vector $g(q)$. Since the process is probabilistic, it could occur that the query vector and nearest vector stay away from each other. In order to reduce such false negatives, the LSH algorithm makes L hash tables, and scans the vectors in the union of the buckets corresponding to each of $g_1(q), g_2(q), \ldots, g_L(q)$.

The key idea of LSPC encoding is to use the hash indices of LSH as the pseudo-code. While LSH treats the set of vectors to solve the nearest neighbor problem, our objective is to treat the sequence of vectors and to solve the word spotting problem by means of classical string matching algorithm.

The definition of LSPC encoding is as follows:

Definition 2. For a vector p, LSPC $C(p)$ is defined as:

$$C(p) = \{\ g_1(p), g_2(p), \ldots, g_L(p)\ \} \tag{2}$$
$$g_i(p) = \{\ h_{i1}(p), h_{i2}(p), \ldots, h_{ik}(p)\ \} \tag{3}$$

where h_{ij} is randomly chosen from the LSH family $\mathcal{H}$.

In our problem the range of each hash function h_{ij} is limited integer $\mathcal{N} = \{1, 2, \ldots, M\}$. Let us focus on $g_i(p)$. Although it has k component, we can represent $g_i(p)$ as a single integer since the range of each component is the limited integer. Therefore, we can represent $g_i(p)$ as an integer between 1 and M^k, with the consequence that LSPC $C(p)$ is represented as a set of integer. We call each integer $g_i(p)$ as an attribute of $C(p)$. Every $C(p)$ has L attributes.

Most important and distinct property of LSPC is the following binary relation. In our pseudo-code scheme, a pair of codes are regarded to be *semiequivalent* if at least one of their attributes are equivalent.

Definition 3. LSPC $C(p) = \{g_i(p)\}$ and $C(q) = \{g_i(q)\}$ are regarded to be semiequivalent if $\exists i$ s.t. $g_i(p) = g_i(q)$.

This definition of the code and the binary relation has an advantage that: if two vectors p, q are close then they should be semiequivalent with high probability and if they are far away from each other then they should be semiequivalent with small probability. This property will be precisely discussed in the next section.

4. Implementing LSPC

4.1. Material multi-dimensional descriptor

In this section we describe how to implement LSPC encoding system to a specific multi-dimensional descriptor (feature vector). The material descriptors employed in this study is Gradient Distribution Feature [9], which shows about 97–99% average precision for word spotting task of the Japanese historical manuscript when used with DTW. This descriptor has 64 dimensionality, and each component of the vector is a positive real number, and the norm of the vector is normalized to unity.

4.2. Spherical LSH (SLSH)

In the preliminary experiment, we have found that LSPC-based retrieval performs as the same accuracy as original descriptor-based retrieval if the following property holds: the pseudo-code of a pair of vectors are invariably semiequivalent when their distance is under 0.8 in the original space, and invariably not-semiequivalent when their distance is over 0.8. Based on this observation, the desirable property of pseudo-code is that: the probability of two pseudo-codes being semiequivalent with respect to their distance rapidly decrease around distance is 0.8 or so.

The earlier proposed LSH family is also available to designing pseudo-code. However, we developed another LSH family that performs better than earlier ones. While earlier LSH family [1, 2, 3] is considering arbitrary vectors in $\mathbb{R}^n$ space, we are considering arbitrary *unit* vectors in $\mathbb{R}^{64}$ space. In other words, since all of our vectors are on a unit sphere, all we have to do is partition the surface of the unit sphere of $\mathbb{R}^{64}$ in contrast to [1, 2, 3] had to partition entire $\mathbb{R}^n$ space. Hence, we developed a novel hash family for partitioning the surface of the unit sphere in high dimension. We named this process as SLSH (Spherical LSH). We reported the evaluation of SLSH as an approximate nearest-neighbor algorithm in [14] . Here we introduce the outline of SLSH.

SLSH uses randomly rotated regular polytope to partition the surface of the unit sphere. Regular polytope is the generalization of regular polyhedron to higher dimensions. It is known that there exists only three kinds of regular polytopes in higher ($n \geq 5$) dimensions, namely, *simplex* with $n+1$ vertices, *orthoplex* with $2n$ vertices, and *hypercube* with 2^n vertices. Suppose that we randomly rotate the regular polytope inscribed in a unit sphere. We can partition the surface of the unit sphere so that all vectors belong to the nearest vertex of the rotated regular polytope.

Definition 4. Let $V = \{v_1, v_2, \ldots, v_N\}$ ($||v_i||^2 = 1$) be a set of vertices that forms a regular polytope in $\mathbb{R}^n$, and let A be a rotation matrix. For an arbitrary unit vector p, a hash function $h_A(p)$ is defined as:

$$h_A(p) = \text{argmax}_i(Av_i \cdot p) \tag{4}$$

Apparently, the range of $h(p)$ is $\{1, 2, \ldots, N\}$, thus the range of $g(p)$ in (3) is $\{1, 2, \ldots, N^k\}$.

By considering A as arbitrary rotation matrix in $\mathbb{R}^{64}$ space, $\mathcal{H} = \{h_A\}$ satisfies the definition of LSH family. SLSH uses this LSH family for hashing.

4.3. Superiority of SLSH in Sensitivity

In this subsection we confirm the superiority of SLSH in the sense that probability of two codes being semiequivalent is sensitive to their distance in the original space. Here we compare SLSH and the random projection method, which is one of the earlier proposed LSH family described in [2].

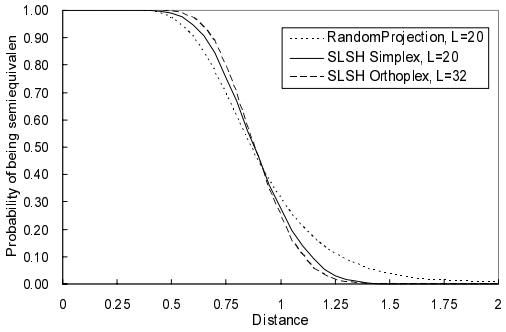


Figure 1. amplified collision probability w.r.t. distance between two vectors

Let $p(c)$ represent the collision probability of single hash function with respect to distance c. For the random projection method, $p(c)$ is analytically obtained by integrating the density functions. For SLSH, we obtained $p(c)$ computationally by Monte-Carlo method.

In the manner of LSH, $p(c)$ is amplified by concatenating hash functions k times and repeating them L times. This amplification is expressed by the following equation:

$$P_{semieq} = 1 - \left(1 - p(c)^k\right)^L \tag{5}$$

where P_{semieq} is the probability of constructed pseudo-code being semiequivalent.

In Fig. 1, the amplified probabilities with respect to distance between two vectors are plotted. In the figure, dotted line represents probability curve based on random projection method, solid line represents SLSH using simplex, and dashed line represents SLSH using orthoplex. For each case L was set as displayed in the figure and k was properly selected (described in the subsequent subsection). We can observe that SLSH curve is rapidly decreasing than earlier LSH curve both in simplex case and in orthoplex case.

4.4. Selection of the parameter

We have a constraint that the range of each attribute N^k should be small, hopefully 16-bit integer, therefore both N and k should be moderate. Recall that N is the number of vertices of the employed polytope, i.e., $N = n + 1$ for simplex, $N = 2n$ for orthoplex, and $N = 2^n$ for hypercube. It is reasonable to think that 2^n is too large when we consider $\mathbb{R}^{64}$ space, thus we excluded a hypercube from our choices. Therefore, N is either 65 in simplex case or 128 in orthoplex case.

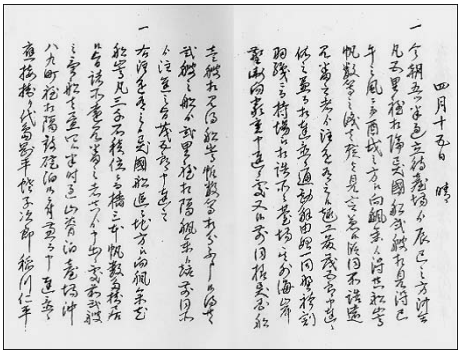


Figure 2. "Akoku Raishiki" written in the mid-19th century.

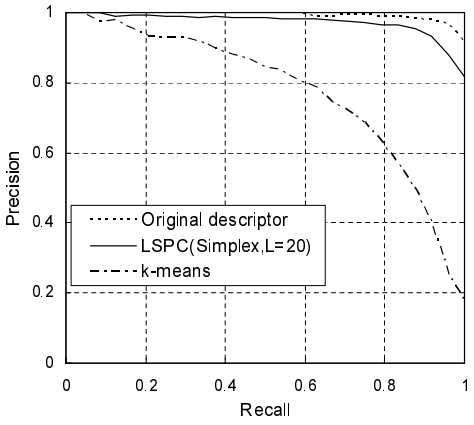


Figure 3. Recall-Precision curve for query word "Inoue Tomizou"

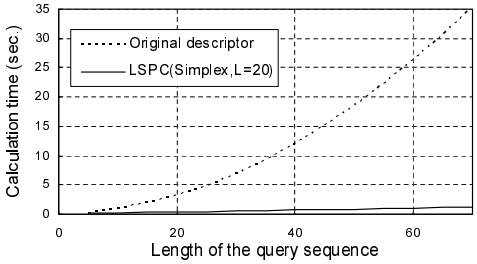


Figure 4. Calculation time vs. query length

If we set $k=2$, N^k can be represented in 16-bit integer both in simplex case and in orthoplex case. Therefore we decided $k=2$ as our standard setting, and $k=3$ as optional setting. For both cases L was selected so that probability curve should rapidly decrease around the distance 0.8 or so. In consequence, we obtained $L=20$ for $k=2$ simplex case, $L=32$ for $k=2$ orthoplex case, and $L=64$ for $k=3$ simplex case. Since L became over 200 for $k=3$ orthoplex case, we discarded it from our choices. Note that increase of L means the increase of space and time.

5. Experiment

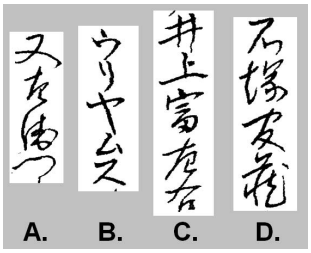
We evaluated the performance of LSPC by the word spotting test. Experimental materials were scanned images of "Akoku Raishiki (The diary of Matsumae Kageyu)" (Fig. 2), a historiography written by a Japanese government employee in the mid 19th century. The tested images consisted of 182 pages, 1553 lines, and 25148 characters. The resolution per single character was about 60×60 pixels.

By using a sliding window, this manuscript was converted to slit sequence with the length of 254,320. Each slit were converted to multi-dimensional descriptor, and subsequently converted to LSPC. For this experiment, we have selected four keywords from the whole document, as shown in Table 1. All selected keywords were human names appearing at least 25 times in the document. For each keyword, each appearance was used as a query. For each query, the ranked list according to the edit distance was obtained. Note that this process could be executed in linear time by dynamic programming. The retrieved region were regarded to be correct if the retrieved region and corresponding manually marked region were overlapping sufficiently (in practice, errors less than 20 slits were allowed).

In order to evaluate the performance of the search process, we have adopted a recall-precision evaluation, which is widely used in information retrieval researches. Recall is the ratio of the number of correctly retrieved words to the number of total relevant words. Precision is the ratio of correctly retrieved words to the number of retrieved words. Measuring precisions at various recall levels, the recall-precision curve is produced. Average precision is the mean of the precision values obtained after each relevant word to the query has been retrieved.

The result is summarized in Table 1. SLSH using simplex showed better performance than random projection (earlier LSH) method even the value of L were the same. For ease of the comparison, we also tested another vector encoding method, k-means quantization. Among several values of k being examined, $k=20$ showed the best result. We could observe that LSPC significantly outperformed k-means quantization. Although it is true that LSPC-based retrieval was slightly inferior to original descriptor-based retrieval, the difference was small and it became smaller as L increased. Figure 3 is the averaged recall-precision curve for keyword InoueTomizou. Also in this figure, we could confirm that LSPC showed almost comparable performance

Table 1. Average-precision for some keywords of "Akoku-Raishiki"



keywords	frequency	average-precision (%)					
		Original descriptor dim = 64	LSPC RandProj $L=20$	LSPC Simplex $L=20$	LSPC Orthoplex $L=32$	LSPC Simplex $L=64$	k-means quantization
A. Matazaemon	165	97.24	88.93	94.53	95.15	96.00	66.12
B. Uriyamusu	73	97.80	96.34	97.75	97.63	98.27	77.22
C. InoueTomizou	25	99.15	91.43	96.99	97.32	97.11	75.90
D. IshizukaKanzou	25	98.20	85.12	95.71	94.73	95.18	69.70

to the original descriptor-based method, and much better performance than k-means quantization.

Figure 4 depicts the calculation time with respect to the length of the query sequence. While original descriptor-based DTW method spent quadratic time, LSPC-based method needed only linear time owing to the classical dynamic programming method that could calculate edit distance of whole document in linear time. Note that CDP [15] may provide another linear time solution to word spotting in non-segmented document. However our method runs faster than CDP because it needs to examine only whether or not at least one of the attribute takes the same value each other while CDP needs to calculate the distance of vector in high dimensions.

6. Conclusion

In this paper, we have proposed a novel scheme for representing character string in the scanned document image without using OCR. The proposed method could represent high dimensional vector by a set of integers and retrieve the image regions that have similar apparences to the query. Our process could run faster than previously proposed method.

References

[1] A. Gionis, P. Indyk, R. Motwani, "Similarity Search in High Dimensions via Hashing," Proc. VLDB1999, pp.518–529, 1999.

[2] M. Datar, P. Indyk, N. Immorlica, V. Mirrokni, "Locality-Sensitive Hashing Scheme Based on p-Stable Distributions," Proc. Symposium on Computational Geometry 2004, pp.253–262, 2004.

[3] A. Andoni, P. Indyk, "Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions," Proc. FOCS'06, pp.459–468, 2006.

[4] T.M. Rath and R. Manmatha, "Word Spotting for Historical Documents," Int. J. on Document Analysis and Recognition, vol.9(2), pp.139–152, 2007.

[5] U.-V. Marti and H. Bunke, "Handwritten Sentence Recognition," Proc. ICPR'00, pp.3467–3470, 2000.

[6] M. Zimmermann and H. Bunke "Hidden Markov Model Length Optimization for Handwriting Recognition Systems," Proc. IWFHR'02, pp.369–374, 2002.

[7] G.A. Fink and T. Plötz "On Appearance-Based Feature Extraction Methods for Writer-Independent Handwritten Text Recognition," Proc. ICDAR2005, vol.2, pp.1070–1074, 2005.

[8] K. Terasawa, T. Nagasaki, and T. Kawashima, "Eigenspace Method for Text Retrieval in Historical Document Images," Proc. ICDAR2005, vol.1, pp.437–441, 2005.

[9] K. Terasawa, T. Nagasaki, T. Kawashima, "Improved Handwritten Text Retrieval Using Gradient Distribution Features," Proc. Meeting on Image Recognition and Understanding, MIRU2006, pp.1325–1330, 2006. (written in Japanese)

[10] Y. Tanaka, K. Takahashi, M.Mozaffari, "Transmedia Machine," Journal of Information Processing, vol.12, no.2, pp.139–146, 1989.

[11] T. Tanaka and Y. Tanaka, "English Text Image Processing of Transmedia System," IPSJ Journal, vol.38, no.7, pp.1389–1398, 1997. (written in Japanese)

[12] C.L. Tan, W. Huang, Z. Yu, Y. Xu, "Imaged Document Text Retrieval without OCR," IEEE Trans. on PAMI, vol.24, no.6, pp.838–844, 2002.

[13] S. Marinai, E. Marino, G. Soda, "Indexing and Retrieval of Words in Old Documents," Proc. ICDAR2003, vol.1, pp.223–227, 2003.

[14] K. Terasawa and Y. Tanaka, "Sperical LSH for Approximate Nearest Neighbor Search on Unit Hypersphere," Proc. 10th Workshop on Algorithms and Data Structures, WADS2007, LNCS4619, pp.27–38, 2007.

[15] R. Oka, "Spotting Method for Classification of Real World Data," The Computer Journal, vol. 41, no. 8, pp. 559–565, 1998.