Large-scale Signature Matching using Multi-Stage Hashing

Xianzhi Du  Wael AbdAlmageed  David Doermann

Language and Media Laboratory
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742
Email: xianzhi,wamageed,doermann@umiacs.umd.edu

Abstract—In this paper, we propose a fast large-scale signature matching method based on locality sensitive hashing (LSH). Shape Context features are used to describe the structure of signatures. Two stages of hashing are performed to find the nearest neighbours for query signatures. In the first stage, we use $M$ randomly generated hyperplanes to separate shape context feature points into different bins, and compute a term-frequency histogram to represent the feature point distribution as a feature vector. In the second stage we again use LSH to categorize the high-level features into different classes. The experiments are carried out on two datasets – DS-I, a small dataset contains 189 signatures, and DS-II, a large dataset created by our group which contains 26,000 signatures. We show that our algorithm can achieve a high accuracy even when few signatures are collected from one same person and perform fast matching when dealing with a large dataset.

Keywords— image retrieval, signature matching, locality sensitive hashing, Tobacco litigation

I. INTRODUCTION

The continuing growth of digital imaging has lead to a significant increase in the number of larger repositories in all aspects of government and business. For example, the now well known Tobacco Litigation has produced a document database of approximately 50 million pages. The massive increase in document database sizes has rekindled interest in the difficult challenges of indexing, querying and retrieving relevant documents.

Signatures constitute strong discriminative features for documents authorized by a specific individual. Hence, signature matching is an important problem in document image retrieval and forensic applications [10][6][11]. Given an image of a query signature, the problem is to find all documents in a (possibly large) database, which were signed by the same person. Signature matching is usually treated as nearest neighbour problem, with emphasis on the numerical representation of the signature and the similarity measures used to compute distances between these representations.

Methods used for shape matching can generally be grouped into two categories based on whether or not a correspondence between the points of the query shape and points of the database shapes is used. Methods that do not explicitly solve a correspondence problem construct global shape representations, using, for example, Fourier descriptors [7] or other structural descriptors [9], and directly compare the shape representations to find the nearest shapes. The main limitation of these methods is that it is typically challenging to extract global descriptor from real images and this therefore affects the matching accuracy. Moreover, extracting global shape representations restricts these methods to shapes with high degrees of rigidity.

Methods that depend on extracting point features on the contours of the shapes being compared, and try to solve a correspondence problem between these points to recover an unknown transformation between the shapes [9]. These methods typically produce better matching performance, since they tolerate lower degrees of rigidity. However, they are generally computationally expensive and require solving the correspondence problem, which renders these methods intractable as the size of the dataset grows. In [9], for example, Shape Context (SC) [1] features are extracted around the contours of the query and test signatures and a correspondence problem is solved using thin-plate splines. Four different similarity measures were computed between the query and test signatures. The method is computationally expensive and as the number of test signatures in the dataset increases, it becomes computationally intractable, since this correspondence problem must be solved between the query signatures and all other signatures in the dataset.

In this paper, we introduce a new signature matching method specifically designed to scale to large-scale datasets. We use a Locality Sensitive Hashing (LSH) [3] in a multi-stage approach to avoid comparing the query signature against all other signatures in the dataset. Moreover, we use LSH to reduce the dimensionality of extracted feature vectors, without the need to perform any clustering or vector quantization.

LSH is a probabilistic dimensionality reduction technique. The idea is to hash points that exist in a high-dimensional feature space such that the probability of collision is much higher for near-by points than for those that are far apart. Points whose hashing values collide with each other fall into the same bucket. LSH has been used for nearest neighbour search [3][4], content-based image retrieval [5] and large-scale clustering [2]. In this paper, we employ Random Projection LSH (RP-LSH)
[1], which uses randomly generated hyperplanes to partition the higher dimensional feature space. RP-LSH depends on generating a set of random vectors $r_i$ and $i = 1, \ldots, M_1$, where $M_1$ is the number of generated random vectors, and use them to hash an input vector $v$, as shown in Equation 1.

$$h_i = \begin{cases} 0 & \text{if } v.r_i \geq 0 \\ 1 & \text{if } v.r_i < 0 \end{cases} \quad (1)$$

Each random vector $r_i$ produces a 1-bit hash value for an input vector $v$. The $M_1$-bits are used to generate a 1D decimal hash code for each input vector. Figure 1 illustrates the idea in 2D feature space. Two hashing vectors $r_1$ and $r_2$ are using to create 2-bit hash codes for the input vectors. Each random vector partitions the feature space into two halves. Points on the left of the random vector are hashed with 0 while points on the right are hashed with 1.

The remainder of this paper is organized as follows. Section 2 provides a detailed description of the proposed approach. Experimental results are discussed in Section 3 using various signature datasets. Finally, Section 4 concludes the paper and highlights open problems.

II. MULTI-STAGE SIGNATURE MATCHING VIA LOCALITY SENSITIVE HASHING

The primary motivation of the method presented in this paper is to be able to perform fast signature matching against large-scale signature datasets. We formulate the problem as a nearest neighbour search problem. However, instead of matching the query signature against the entire signature dataset, we limit the search space using Locality Sensitive Hashing (LSH), effectively limiting the similarity measures to a few signature that share the same hash code with the query signature. Moreover, rather than using simple global features to represent the signature, such as Fourier descriptors, and to accurately capture the structure of the signatures, we use Shape Context (SC) features computed on the contour. Furthermore, to avoid solving the computationally expensive point correspondence problem, we use LSH to reduce the dimensionality of the SC features and construct a unified representation for the signature with low computational overhead.

Figure 2 shows a sample signature and extracted contour. The contour is evenly sampled and a fixed number of points, $K$, are extracted. SC is calculated around each of contour sample, producing a $D$-dimensional feature vector, where $D = n_r \times n_\theta$ and $n_r$ and $n_\theta$ are the number of concentric circles and the number of angular sections, respectively. Hence, each signature $i$ is essentially represented using a $K \times D$-dimensional feature vectors, $v_1^i, v_2^i, \ldots, v_K^i$.

When a given image is represented using multiple multi-dimensional feature vectors, it is customary to use a clustering method to create a codebook for reducing the dimensionality of the vectors by assigning the vector to the nearest cluster centroid. Sivic and Zissermann [8] use k-means to cluster a set of training vectors and create a codebook that is used for dimensionality reduction. However, this approach has two main limitations. First, clustering methods are computationally expensive in terms of both memory and time (e.g. k-means is $O(N^2)$ in both memory and time). Therefore, as the number of samples increases, using clustering methods becomes computationally intractable. Second, the number of samples needed to build the codebook using clustering must be proportional to the size of the dataset, since it has to properly cover the feature space. Therefore, as the size of the dataset increases, the increased number of samples renders clustering methods computationally unusable.

Let $M_1$ be a number of $D$-dimensional zero-mean, unity-variance random vectors, where $M_1$ represents the length of the locality-sensitive hash code. For a given signature $i$, all vectors $v_j^i$ are hashed using the $M_1$ vectors as shown in Equation 2.

$$h_m = \begin{cases} 0 & \text{if } v_j^i \cdot r_m \geq 0 \\ 1 & \text{if } v_j^i \cdot r_m < 0 \end{cases} \quad (2)$$
where \( m = 1, 2, \ldots, M \) and \( j = 1, 2, \ldots, K \), which produces \( M \)-bit hash code for each vector. Hash codes for all vectors extracted from the dataset are pooled to construct a dictionary of distinct hash codes. For each signature \( i \), the \( K \) hash codes are used to calculate a Term-Frequency (TF) (i.e. histogram), which is used a unified signature representation.

The TF is a higher level representation of the signature and it encodes the structural features of the signature represented by the shape context features. Therefore, signatures created by the same author should have similar TF representations. Given the TF representation of a query signature, in order to find other signatures created by the same author, we must perform a linear search to compute the similarity between the query and test signatures. However, as the size of the signature dataset increases, the linear search becomes computationally inefficient.

We use a second set \( M_2 \) of zero-mean, unity-variance random vectors to compute locality-sensitive hash codes for the TF representations of the signatures. However, rather than converting the hash codes to decimal numbers, signatures are compared by computing the Hamming distance between binary hash codes. The Hamming distance is used to find other signatures in the dataset that shares the same \( M_2 \) binary hash with the query signature, thus avoiding the linear search.

### III. Experiments

To evaluate the performance of the proposed method, two sets of experiments were conducted. The first set of experiments was on a small dataset to study the performance of our signature matching algorithm when the number of samples of each author is small. The second set of experiments was performed on a larger data set to illustrate the performance of the algorithm as the size of the dataset grows and demonstrate the scalability of our method.

#### A. Evaluation protocol and datasets

We use a top-N accuracy strategy to assess the performance of the proposed signature matching method. For each query signature, if another signature from the same author appears in the top-N neighbours, selected by our method using the Hamming distance, we regard this as a successful match. The process is repeated for all signatures in the dataset and the average of all signatures is reported.

Accuracy statistics were collected on two datasets. The first dataset (DS-I) contains 189 signatures collected from 63 authors with three signatures from each author. Signatures in this dataset are fairly clean, since they are extracted from high-quality document images with little noise or machine-printed text on them. Several signatures in this dataset are shown in Figure 3.

The second dataset (DS-II) contains 26,661 signatures, from 890 individuals, extracted from the Tobacco litigation document dataset. This dataset is more representative of a realistic signature matching problem. Unlike the previous dataset, signatures here are blocks extracted from letters and memorandums. Several common and unrelated content and degradations appear, such as:

- Machine-printed text or lines which do not belong to the signature
- Signatures may be occluded
- Different styles may be used by the same writer, such as, the use of initials in place of names
- Images may be of different sizes and qualities
- Some signatures may be overly simple

Figure 4 shows some samples of the signatures in DS-II illustrating the varying levels of image quality, underlying text, and signature complexity. DS-II will be publicly available on the Language and Media Processing Laboratory website (http://lamp.cfar.umd.edu/media.htm).

#### B. Results

The first set of experiments was conducted on DS-I. The resolution of the signatures is \( 480 \times 152 \) and the number of contour points extracted is \( K = 300 \). To extract shape context (SC) features, we use \( n_r = 5 \) and \( n_\theta = 12 \). Therefore, for each signature in the dataset, we extract 300 60-D feature vectors.

To counteract the randomization property of hash functions, we repeat hash function generation process 40 times. Each
Fig. 5. Accuracy of the proposed signature matching algorithm on DS-I. The reported accuracy is the average of 10 different runs of the algorithm.

Table I: Peak accuracy of the proposed signature matching algorithm for DS-I

<table>
<thead>
<tr>
<th>Top-N</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-10</td>
<td>92.80%</td>
</tr>
<tr>
<td>Top-5</td>
<td>92.28%</td>
</tr>
<tr>
<td>Top-3</td>
<td>90.58%</td>
</tr>
<tr>
<td>Top-1</td>
<td>87.41%</td>
</tr>
</tbody>
</table>

DS-I was used to evaluate the performance of the method proposed by Zhu et al. [12]. They reported a less than 10% accuracy on this dataset due to the small number of signatures from each author. Meanwhile, the execution time for [12] requires approximately 11 hours to train and the same amount of time to find the nearest neighbours for all signatures in the dataset, for a total of 22 hours. On the other hand, the execution time for our algorithm is approximately 1.77 seconds.

The second set of experiments was conducted on DS-II. All signatures were normalized to the mean signature size of 482 × 182. Other system parameters were set similar to the previous experiment. Figure 6 shows the performance of the proposed algorithm. Peak accuracy was obtained at $M_1 = 8$.

The execution time of the proposed algorithm, to find matching signatures for all samples in the dataset, was 24 minutes. We were not able to test the method proposed in [12] due to the computational intractability, since it is estimated that it years to run. Tables I and II show the peak performance of the proposed method for various top-N accuracies, for both DS-I and DS-II, respectively.

Table II: Peak accuracy of the proposed signature matching algorithm for DS-II

<table>
<thead>
<tr>
<th>Top-N</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-25</td>
<td>86.24%</td>
</tr>
<tr>
<td>Top-10</td>
<td>84.15%</td>
</tr>
<tr>
<td>Top-5</td>
<td>82.27%</td>
</tr>
<tr>
<td>Top-1</td>
<td>76.99%</td>
</tr>
</tbody>
</table>

IV. CONCLUSIONS

We propose a new approach for fast, scalable signature matching. The motivation was to develop a signature matching method that efficiently scales to large-scale datasets. The proposed method formulates the signature matching problem as a nearest neighbour problem. However, in order to achieve scalability we use locality sensitive hashing to limit the signature-signature comparisons to a small number, rather than performing linear search against a very large number of signatures.

Moreover, rather than using global image descriptors, which does not properly encode the structural features of the signature, we extract the contour and sample it. Shape context features are computed at each contour sample point. We use another level of locality sensitive hashing functions to reduce the dimensionality of the SC features and compute a term frequency representation for the signature. This gives us two advantages. First, we avoid using any data-dependent, computationally intensive clustering techniques to compute codebooks for the extracted features. Second, we avoid explicitly solving correspondence problem.

The accuracy of the proposed approach was evaluated on two datasets. The first dataset is a small, relatively clean dataset with a small number of signatures per author. The proposed algorithm outperforms the state-of-the-art algorithm evaluated on this particular dataset, in both accuracy and computational time. The second dataset is a very large dataset extracted from the Tobacco litigation document dataset and suffers from varying levels of image noise. We achieve a peak of 84.15% for top-10 best matches with low computational time.

The basic assumption made in this paper is that the signatures are detected properly in the document image. However, signature detection algorithms are error-prone and often produce partial signatures or partially occluded signatures. One of the main challenges that remains open is handling partially occluded signatures, were the extracted features only encode the structure of parts of the signature.

Finally, a hybrid approach could be designed such that
the our method is tuned to achieve 100% recall, regardless of
the number of false alarms and then another computationally
intensive method (e.g. [12]) which solves point correspondence
problems, is used to find true signature matches.

REFERENCES

recognition using shape contexts. IEEE Transactions on Pattern
Analysis and Machine Intelligence, 24(24), April 2002.

spectral clustering for large-scale datasets,. In ACM International Sym.

dimensions via hashing. In Proceedings of the 25th Very Large Data

removing the curse of dimensionality. In Symposium on the Theory of

metrics. In Proceedings of IEEE Conference on Computer Vision and
Pattern Recognition, June 2008.

verification using hmm for random, simple and skilled forgeries. In
International Conference on Document Analysis and Recognition,

descriptors. IEEE Trans. Pattern Analysis and Machine Intelligence,

object matching in videos. In International Conference on Computer

O. Frieder. Document image retrieval using signatures as queries. In
203, 2006.

tion using compression networks and positional cuttings. In Proceedings
of IEEE Workshop on Neural Networks for Signal Processing, volume 1,

on neural gas based vector quantization. In IEEE International Joint

for document image retrieval. IEEE Transactions on Pattern Analysis
and Machine Intelligence, 31(11), November 2009.