Fast Instance Search Based on Approximate Bichromatic Reverse Nearest Neighbor Search

Masakazu Iwamura, Nobuaki Matozaki, Koichi Kise
Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuencho, Naka, Sakai, Osaka 599-8531, Japan
{masa,kise}@cs.osakafu-u.ac.jp

ABSTRACT

In the TRECVID Instance Search (INS) task, it is known that use of BM25, which is an improvement of the TFIDF, greatly improves retrieval performance. Its calculation, however, requires tremendous amount of computational cost and this fact makes its use intractable. In this paper, we present its efficient computational method. Since the BM25 is obtained by solving the bichromatic reverse nearest neighbor (BRNN) search problem, we propose an approximate method for the problem based on the state-of-the-art approximate nearest neighbor search method, bucket distance hashing (BDH). An experiment using the TRECVID INS 2012 dataset showed that the proposed method reduced computational cost to less than 1/3500 of the brute-force search with keeping the accuracy.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Search process

General Terms

Algorithms

Keywords

Bichromatic reverse nearest neighbor search; Approximate algorithm; Video retrieval; BM25; TFIDF

1. INTRODUCTION

Finding a specific thing such as person, object and place from a large-scale video by a computer is a challenge. It is tackled in the Instance Search (INS) task of the TREC Video Retrieval Evaluation (TRECVID)[1].

The most critical problem is how to realize high accuracy. Distinguishing a target from others inherently requires human-like intelligence. For a computer that does not have it, methods that compensate for lack of it are required. A feasible way is based on keypoint matching. That is, the method extracts keypoints from both reference and query video images and casts votes for likely topics based on matching results of keypoints. Recently, it is shown that introducing BM25 ranking function [13] and its improvement, called exponential BM25, as weights of votes into the keypoint matching is quite effective in the INS task [10, 11]. In particular, a method using the exponential BM25 was ranked at the second place in the TRECVID INS 2013 [11].

This fact, however, leads to another critical problem, which is the high computational cost. In order to calculate the TFIDF, BM25 and exponential BM25, all reference keypoints whose nearest neighbors are a given query keypoint have to be found. This is known as the bichromatic reverse nearest neighbor (BRNN) search problem [5]. With the naive method, i.e., the brute-force search, the distances of all combinations of reference and query keypoint pairs have to be calculated. This tremendous amount of computational cost makes their use intractable. Some methods that more efficiently solve the problem than the naive method (e.g., [7, 15, 4]) are known. They, however, do not scale because they are designed to solve the problem *without error*. In the nearest neighbor search problem, which is a related but different problem, it is well-known that approximate methods require much less computational cost by allowing small error (i.e., with slight reduction of retrieval accuracy), compared with the non-approximate methods.

In this paper, we propose an approximate BRNN search method based on the state-of-the-art approximate nearest neighbor search (ANNS) method, bucket distance hashing (BDH) [2]. An experiment using the TRECVID INS 2012 dataset showed that the proposed method reduced computational cost to less than 1/3500 of the brute-force search with keeping the accuracy.

2. RECOGNITION METHOD

2.1 Overview

Figure 1 illustrates a keypoint matching based recognition method that we use in the paper. In (1), keypoints such as SIFT [8], Color SIFT [9] and OpponentSIFT [16] can be used from masked query images of a topic. In (2), for pairing keypoints from reference and query images, fast keypoint matching based on ANNS can be used. However, if the weights of votes such as the TFIDF [3], BM25 [13] and other methods are considered in the succeeding process, it

[1]http://www-nlpir.nist.gov/projects/trecvid/

MM'14, November 3–7, 2014, Orlando, Florida, USA.

http://dx.doi.org/10.1145/2647868.2654988.

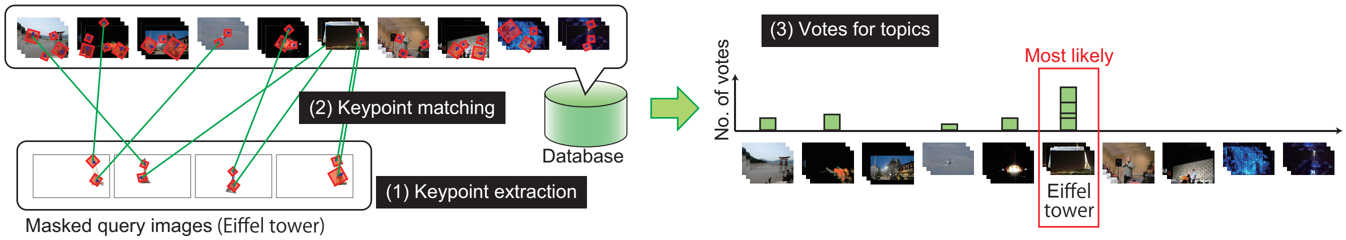


Figure 1: Keypoint matching based recognition method. (1) keypoints are extracted from masked query images of a topic, (2) keypoints from reference and query images are paired, (3) votes for topics are cast based on the keypoint matching result.

is not the case. In (3), votes for topics are cast based on the matching result in (2).

2.2 BM25

The BM25 [13] is an improvement of the TFIDF [3] in terms of taking into account the length of documents. They are originally developed for the document retrieval task. To apply them to the instance search task from videos, a document and a word are regarded as a video (a set of images) and a keypoint, respectively.

For the preparation of defining the weight of BM25 for the instance search task, the Term Frequency (TF) and the Inverse Document Frequency (IDF) are introduced. The TF describes a video v by a query keypoint q_i. The TF $\mathrm{TF}(q_i, v)$ is defined by the number of occurrence of the query keypoint q_i in the video v. The IDF represents the importance of the keypoint q_i. The IDF is given by

$$\mathrm{IDF}(q_i) = \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right), \tag{1}$$

where N and n_i are the total number of videos, and the number of videos that contain the query keypoint q_i, respectively. In [12], instead of Eq. (1),

$$\mathrm{IDF2}(q_i) = \log\left(\frac{N - n_i^2 + 0.5}{n_i^2 + 0.5}\right) \tag{2}$$

was used. If the argument of log in Eq. (2) is less than one, $\mathrm{IDF2}(q_i) = 0$, while it is expected to rarely happen. IDF2 is intended to increase the discrimination power of keypoints and shown to be more effective than IDF in [12]. We follow this.

Finally, the weight of the BM25 for a video v and a set of query keypoints $Q = (q_1, \ldots, q_n)$ is given as

$$w(Q, v) = \sum_{q_i \in Q} \frac{\mathrm{IDF2}(q_i) \cdot \mathrm{TF}(q_i, v) \cdot (k+1)}{\mathrm{TF}(q_i, v) + k(1 - b + b\frac{|v|}{\mathrm{avdl}})}, \tag{3}$$

where avdl and $|v|$ are the average number of keypoints per video, and the number of keypoints in the video v, respectively. k and b are parameters. Following [11], $k = 2$ and $b = 0.75$ were used in the paper.

In order to use $\mathrm{TF}(q_i, v)$, how many times the query keypoint q_i appears in the video v has to be calculated. Similarly, in order to use n_i, how many videos contain the query keypoint q_i has to be calculated. They are solved by finding the closest query keypoint(s) to keypoints in videos. This is known as the Bichromatic Reverse Nearest Neighbor (BRNN) search problem. In the next section, we present an approximation algorithm for solving the problem.

3. APPROXIMATE BRNN SEARCH

3.1 Reverse Nearest Neighbor Search

Reverse nearest neighbor (RNN) search problem is to find the nearest neighbor (query) to a datum [5]. This is closely related to the well-known nearest neighbor (NN) search problem, which finds the nearest neighbor (datum) to a given query. The NN search can be formulated as

$$\mathrm{NN}(\boldsymbol{q}) = \{\boldsymbol{r} \in S | \forall \boldsymbol{p} \in S : d(\boldsymbol{q}, \boldsymbol{r}) \leq d(\boldsymbol{q}, \boldsymbol{p})\}, \tag{4}$$

where $d(\boldsymbol{a}, \boldsymbol{b})$ is the distance between $\boldsymbol{a}$ and $\boldsymbol{b}$, and S is the set of data. On the other hand, the RNN search can be formulated as

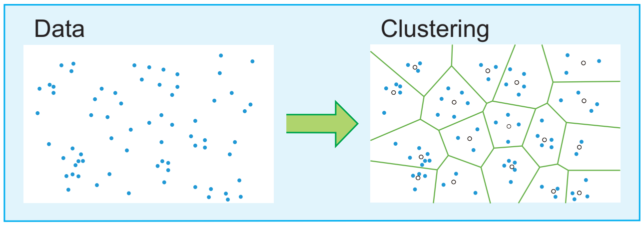
$$\mathrm{RNN}(\boldsymbol{q}) = \{\boldsymbol{r} \in S | \forall \boldsymbol{p} \in S : d(\boldsymbol{r}, \boldsymbol{q}) \leq d(\boldsymbol{r}, \boldsymbol{p})\}. \tag{5}$$

The RNN search problem is divided into monochromatic reverse nearest neighbor (MRNN) search (e.g., [17, 6, 14]) that does not distinguish data and queries, and bichromatic reverse nearest neighbor (BRNN) search (e.g., [7, 15, 4]) that does. Since the problem of our interest is the latter, we focus on that below.

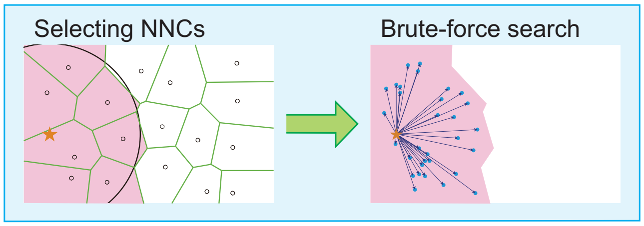
3.2 Bucket Distance Hashing

Before presenting the proposed method for the BRNN search problem, let us present the BDH [2] where the proposed method is based on. The BDH is the state-of-the-art ANNS method. It consists of indexing and retrieval processes, which are performed before and after a query is given. Figure 2 illustrates the processes. In the indexing process in Fig. 2(a), data are clustered for efficient retrieval. For clustering, the k-means algorithm is employed. Obtained clusters are called buckets. The retrieval process in Fig. 2(b) are divided into two steps, as most of ANNS methods are. In the first step, nearest neighbor candidates (NNCs) are selected. They are used for distance calculation in the brute-force search in the second step. Since there is no room to improve in the second step, only the first step, i.e., selection of NNCs, is the key to improve the performance of the ANNS method. The BDH employs the branch and bound algorithm for the selection; NNCs are efficiently selected based on their bucket distances where the distances between the query and the centers of the buckets that the data belong. In [2], it is shown that the BDH reduced computation times by one-third compared with the previous state-of-the-art on an experiment using 100 million SIFT features.

The parameters of the BDH are summarized in Table 1. For clustering of the indexing, the k-means algorithm is applied to each subspace of the feature space. The dimensionality p of the subspaces is a parameter. The data are



(a) Indexing.



(b) Retrieval.

Figure 2: The indexing and retrieval processes of the bucket distance hashing (BDH). NNC represents nearest neighbor candidate that are used for distance calculation in the brute-force search.

Table 1: Parameters of BDH.

p	The dimensionality of the subspaces for indexing
H_{size}	The size of the hash table
C	The minimum number of NNCs

stored in a hash table. Its size is determined by a parameter H_{size}. There is a parameter C that determines the minimum number of NNCs.

3.3 Proposed Method

The key of the proposed method to solve the BRNN search problem is to borrow the first step of the retrieval process of the BDH. The question arisen is whether it is possible to apply a method designed for the NN search problem to the RNN search problem.

As described in Eqs. (4) and (5), both the NN and RNN search problems are to find the datum (or query) that has the minimum distance. However, the minimum distance in what sense is different. The NN search problem finds *the closest datum from the query*, and the RNN search problem finds *the closest query from the datum*. Since the operation of "finding the one that achieved the minimum distance" should be applied to different thing (either datum or query), the second step (the brute-force search) of the retrieval process of the BDH is not directly applicable. On the other hand, the first step can be directly applicable. This is because it just selects NNCs based on their bucket distances. It is also derived in [1] that an RNN search problem can be reduced to an NN search problem. As long as we use a symmetric distance such as Euclidean distance, the calculated distances are symmetry (i.e., $d(\boldsymbol{a}, \boldsymbol{b}) = d(\boldsymbol{b}, \boldsymbol{a})$). This indicates that an ANNS method with better performance is expected to achieve better performance also in the RNN search problem.

Table 2: Parameters of the proposed method.

p	The dimensionality of the subspaces for indexing
H_{size}	The size of the hash table
R	The search radius of NNCs

Table 3: Parameters of the proposed method used in the experiment.

Parameters	Values
p	2, 3, 4
H_{size}	2^{29}, 2^{30}, 2^{31}
R	0.45, 0.50, 0.5, 0.525, 0.55

Most of the parameters of the BDH are shared with the proposed method, as summarized in Table 2. One exception is that the parameter C that determines the minimum number of NNCs. Instead, a new parameter R that determines the search radius of NNCs is introduced.

4. EXPERIMENT

We used the TRECVID INS 2012 dataset which consists of 76751 Internet videos as references and images of 21 topics as queries. As for the reference images, the videos were sampled at two frames per second, and keypoints of the OpponentSIFT [16] were extracted from the sampled images. Since the feature vectors of the OpponentSIFT were 384 dimensions and too high, 60 dimensions that have the largest variances were extracted using the principal component analysis. At the 60th dimension, the cumulative percentage contribution reached 80%. Then, all feature vectors were normalized so that their norms became one. As for the query images, keypoints were extracted and processed as well as the reference images. In average, a query topic consists of five images.

For the accuracy measure, the mean average precision (MAP) was used. We employed servers where 4 CPUs (AMD Opteron 6328, 2.6GHz, 12 cores) and 512GB memory were installed. All data were stored on memory. Computational times were converted to the ones in the case programs were executed as a single thread on a single core.

We used the brute-force search as the baseline method. In the method, the minimum cosine measure was set to 0.9 because this achieved the best in the preliminary experiment. This means that queries whose cosine measures were less than 0.9 were discarded. While 0.95 was used as the best value in [11], difference in experimental conditions such as different keypoint and different frame rate may lead to a different conclusion from ours. As for the proposed method, the parameters of the proposed approximate BRNN search were changed. The used parameters are summarized in Table 3. In the brute-force search of the proposed method, queries whose cosine measures were less than 0.9 were discarded. Figure 3 shows the relationship between computational time and accuracy. The baseline method took 46 days to achieve a MAP of 0.205. The computational time and accuracy of the proposed method were in the trade-off relationship. It achieved almost same accuracy with less than 20 min. More precisely, a MAP of 0.203 was achieved in 16.2 min. and that of 0.206 was achieved in 18.8 min. Their com-

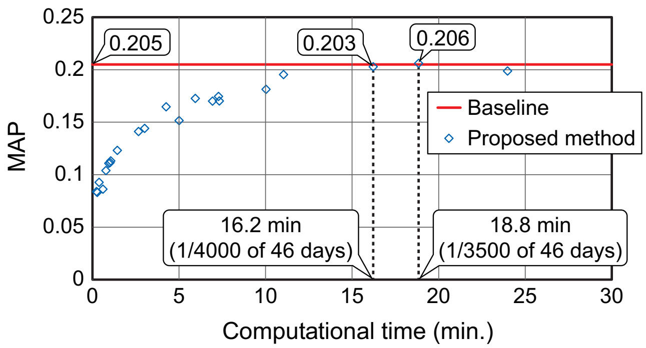


Figure 3: Relationship between computational time (min.) and accuracy (MAP). The computational time of the baseline method was 46 days.

putational times corresponds to about 1/4000 and 1/3500 of 46 days, respectively.

5. CONCLUSION

In this paper, we reduced the computational cost required for achieving high accuracy in the TRECVID Instance Search (INS) task. It is known that use of BM25 and its improvement greatly improve retrieval performance. Their calculation, however, requires tremendous amount of computational cost and this fact makes their use intractable. The BM25 is obtained by solving the bichromatic reverse nearest neighbor (BRNN) search problem.

In this paper, as a scalable solution for the problem, we proposed an approximate bichromatic reverse nearest neighbor (BRNN) search algorithm based on the state-of-the-art approximate nearest neighbor search (ANNS) method, bucket distance hashing (BDH). The idea comes from the inference that an ANNS method with better performance is expected to achieve better performance also in the BRNN search problem. An experiment using the TRECVID INS 2012 dataset showed that the proposed method makes the calculation of BM25 tractable; it reduced computational cost to less than 1/3500 of the brute-force search with keeping the accuracy.

Future work includes experimental comparison with existing approximate BRNN search methods such as [1], whereas the base ANNS method of [1], locality-sensitive hashing (LSH), is far slower than the BDH. Future work also includes application of the proposed method to other tasks than calculation of the BM25 and its variants.

6. ACKNOWLEDGMENTS

This work was supported by JST CREST project, and JSPS KAKENHI #25240028.

7. REFERENCES

[1] D. Arthur and S. Y. Oudot. Reverse nearest neighbors search in high dimensions using locality-sensitive hashing. arXiv.org, Nov. 2010. v1.

[2] M. Iwamura, T. Sato, and K. Kise. What is the most efficient way to select nearest neighbor candidates for fast approximate nearest neighbor search? In Proc. ICCV, Dec. 2013.

[3] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, 28(5):11–21, 1972.

[4] J. M. Kang, M. F. Mokbel, S. Shekhar, T. Xia, and D. Zhang. Continuous evaluation of monochromatic and bichromatic reverse nearest neighbors. In Proc. IEEE 23rd Int'l Conf. on Data Engineering (ICDE 2007), pages 806–815, 2007.

[5] F. Korn and S. Muthukrishnan. Influence sets based on reverse nearest neighbor queries. In Proc. ACM SIGMOD Int'l Conf. Management of Data, pages 201–212, May 2000.

[6] K.-I. Lin, M. Nolen, and C. Yang. Applying bulk insertion techniques for dynamic reverse nearest neighbor problems. In Proc. 7th Int'l Database Engineering and Applications Symposium, pages 290–297, 2003.

[7] Y. Liu, R. C.-W. Wong, K. Wang, Z. Li, C. Chen, and Z. Chen. A new approach for maximizing bichromatic reverse nearest neighbor search. Knowledge and Information Systems, 36(1):23–58, 2013.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. Int'l Journal of Computer Vision, 60(2):91–110, Nov. 2004.

[9] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. Int'l Journal of Computer Vision, 60(1):63–86, 2004.

[10] M. Murata, T. Izumitani, H. Nagano, R. Mukai, K. Kashino, and S. Satoh. Ntt communication science laboratories and national institute of informatics at trecvid 2012 instance search and multimedia event detection tasks. In Proc. TRECVID 2012 Workshop, 2012.

[11] M. Murata, H. Nagano, K. Kunio, and S. Satoh. Ntt communication science laboratories and national institute of informatics at trecvid2013 instance search task. In Proc. TRECVID 2013 Workshop, 2013.

[12] M. Murata, H. Nagano, R. Mukai, K. Kashino, and S. Satoh. Instance video search by images as a query. In IEICE Technical Report, PRMU, volume 112, pages 215–219, Feb. 2013. In Japanese.

[13] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval, 3(4):333–389, 2009.

[14] I. Stanoi, D. Agrawal, and A. E. Abbadi. Reverse nearest neighbor queries for dynamic databases. In Proc. ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pages 44–53, 2000.

[15] Q. T. Tran, D. Taniar, and M. Safar. Bichromatic reverse nearest-neighbor search in mobile systems. IEEE Systems Journal, 4(2):230–242, 2010.

[16] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Color descriptors for object category recognition. In Proc. 4th European Conference on Color in Graphics, Imaging and Vision (CGIV2008), volume 4, pages 378–381, June 2008.

[17] C. Yang and K.-I. Lin. An index structure for efficient reverse nearest neighbor queries. In Proc. 17th Int'l Conf. on Data Engineering, pages 485–492, 2001.