

Research and Application of Image Retrieval Improved Algorithm Based on BOF

Lu Hai Hong¹ and Zhu Zhi Xiang²

1. Institute of IoT & IT-based Industrialization Xi'an University of Post & Telecommunication, Xi'an, Shanxi, 710072, China
2. Institute of IoT & IT-based Industrialization Xi'an University of Post & Telecommunication, Xi'an, Shanxi, 710072, China

Abstract

The feature description of image and the index mechanism of the feature are the keys to realize the content-based image retrieval, aiming at the problem of massive image data and “dimension curse”, this paper proposes the BOF-based image retrieval improved algorithm, and combines with VLAD and soft assignment it generates the soft assignment local aggregation descriptor (SA-VLAD) which has a better ability to resist the dimension reduction and a higher recognition rate. When the index mechanism IVFADC is at query time, to ensure the recall ratio and precision rate of the result, the candidates inverted index chain are increased, which leads to the problems of distance calculation and the query time's increasing. For this point, in the index phase, the scattered distribution is carried out aiming at the database vector, which reduces the burden of distance calculation, and improves the quality of the query results at the same time. The experimental results show that the algorithm in this paper obtains a good effect in the content-based massive image database retrieval.

Keywords: Image Feature, Index Structure, Soft Assignment, Membership Weight

1. Introduction

With the rapid developments of the Internet and the multimedia technology, people can access to the multimedia data, especially the image data, and the number of the image data increases sharply. How to realize the similarity retrieval of the image content becomes a very important research topic. At present, the domestic and foreign research institutions have achieved many related results, and developed a series of commercial and prototype systems. In domestic, there are TV-FI (TV-Find It) system from Tsing hua University and Web scope CBR from Zhejiang University [1-4]. In foreign, there are QBIC (Query by Image Content) and Virage developed by IBM, Photo book developed by the multimedia laboratory of MIT, Visual SEEK and Video Q developed by Columbia etc [4-6]. Now, there are two important problems need to be solved in the content-based image retrieval system.

(a) Image features. Image features include the high-level semantic feature and the low-level feature. Due to the existence of “semantic gap”, it is difficult for the high-level semantic feature to be obtained directly through the computer, and to retrieve image with the semantic keywords is still a difficult task. And it is another important method of image retrieval to use the similarity matching approach of the image's low-level feature for image retrieval, but this method depends on the effectiveness of the image feature [7-10]. For its features of robustness, locality and strong discrimination ability, SIFT feature becomes one of the most commonly used low-level features in the image retrieval field, but its features of high dimension and quantity lead to the difficulty of indexing and querying [12-14]. BOF

(Bag of Feature) method aggregates couple of local feature vectors of an image into a high-dimension vector in the basis of the local feature vectors, through the encoding of the aggregated vector and the organization using the inverted index, which reduces the storage space and the cost of query. However, the price of reducing the feature vectors is that the single feature's data dimension of the image is very high, which is easy to cause the "dimension disaster". VLAD (Vectors of Locally Aggregated Descriptors) is the improvement of BOF method, by using the Fisher Kernel thoughts it produces the more simple aggregated descriptors with better query performances. But during the polymerization, VLAD uses the hard allocation strategy to assign the local feature vectors to the clustering categories, which may lead to the situation that two close descriptors be assigned to completely different clustering, thus fails to reflect the distribution features of the vector itself, reducing the efficiency of the retrieval.

(b) The index structure and methods. For massive image data retrieval, its feature data need to be indexed to improve the retrieval efficiency. While most of the image features data are high-dimension data, and no matter based on the feature driven methods, such as k-d-tree or Quad-tree, or based on the data driven methods, such as R^* -Tree, R_{\pm} Tree, SR -Tree and X -Tree in the R -Tree family, it is difficult to deal with the "dimension disaster" brought by high-dimension feature.

Locality-Sensitive Hashing (LSH) and the inverted index based on BOF are the index methods that can better cope with the "dimension disaster" in recent years. Although LSH can deal with the high-dimension feature data in a certain extent, it increases the storage space of indexing by several times, which makes it difficult to adapt to the massive database retrieval. And while the vocabulary tree based on BOF can save the indexing space greatly relative to LSH, it still can not meet the performance demands of the massive image retrieval. On the basis of VLAD, Jegou and others make use of the methods of Product Quantization (PQ) and Asymmetric Distance Computation (ADC) to realize the Inverted File with Asymmetric Distance Computation (IVFADC) and build index for massive image library [15]. But only under the 20 byte image coding can this approach ensure the query efficiency which is higher than that of BOF.

2. The Soft Assignment Based on Local Aggregation

Same as VLAD, SA-VLAD is the polymer of one image's multiple local descriptors. SA-VLAD adopts the soft assignment method to replace the hard assignment, which makes up for the inadequacy of VLAD and generates the more accurate image descriptors.

A. The Calculation of the Membership Weight

The soft assignment thought is derived from the fuzzy set theory proposed by Zadeh in 1965, the theory holds that each element belongs to a particular collection in a certain degree, and can also belong to several sets in different degrees [7]. Compared with the traditional "yes or no" two-valued logic, the fuzzy set theory can better reflect the object subordinate relation in the actual application. The soft assignment method distributes the different weights according to the distance from the feature vector to the center of the cluster, which is called the membership weight of the feature vector relative to the cluster.

The calculation of the membership weight adopts the membership function in the fuzzy k-means method. In the fuzzy k-means, the error sum of squares between the vectors in the clustering and the center of the clustering is required to be the smallest, it usually uses the criterion function to measure. And the criterion function is as follows:

$$D_i = \sum_{i=1}^k \sum_{j=1}^m [t_i(k_x)]^2 \quad (1)$$

In the above formula, D_i is the cluster center of the i cluster, k is the number of the cluster, t_i is the j vector in the n local feature vectors, $t_i(k_x)$ is the membership degree of the feature vector t_i relative to the i cluster, and b is constant which is greater than 1 and is used to control the fuzzy degree of the cluster. Here, the value of b is 2.

The fuzzy clustering method demands that the sum of the membership degrees of each local feature vector relative to each cluster to be 1. It is shown as follow:

$$\sum_{i=1}^k t_i(k_x) = 1, i = 1, 2, \dots, m \quad (2)$$

Get the minimum value of i_n in the case that the sum of the membership degree is 1, and set the partial derivative of i_n relative to t_i and $t_i(k_x)$ to be 0, the obtained necessary condition is as follows:

$$D_i = \frac{\sum_{j=1}^m [t_i(k_x)]^2 t_i}{\sum_{i=1}^m [t_i(k_x)]^2}, i = 1, 2, \dots, n \quad (3)$$

$$t_i(k_x) = \frac{(1 - \|t_i - d_i\|^2)^{\frac{1}{(c-1)}}}{\sum_{k=1}^c (1 - \|t_i - d_i\|^2)^{\frac{1}{(c-1)}}}, i = 1, 2, \dots, m, i = 1, 2, \dots, n \quad (4)$$

Thus, the membership function of t_i in the i cluster $t_i(k_x)$ is gained.

B. The Generating Algorithm of SA-VLAD

SA-VLAD uses the k-means method to obtain the k cluster center, and then adopts the distribution principle with the minimum error, uses the soft assignment approach of the fuzzy clustering method for reference to distribute the local feature vectors, finally obtains the membership weight and calculates the aggregation descriptor. The generation process is shown in Figure 1.

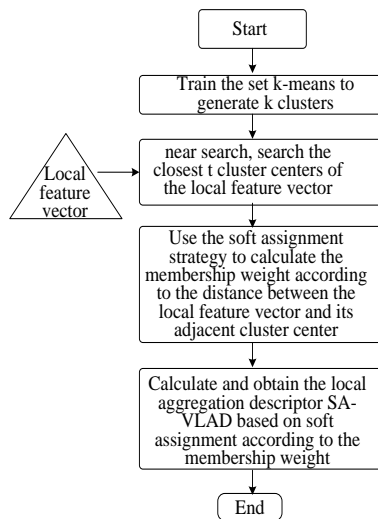


Figure 1. The Generation Schematic Diagram of SA-VLAD

Based on k clusters to carry out the aggregation of the local feature vector to generate the SA-VLAD descriptor, the detailed steps are as follows:

(1) Initialize SA-VLAD to the zero vector s_v with the dimension of $m \times n$. m is the number of the cluster centers and n is the dimension of the image's local feature vector.

(2) Through the near search, the local feature vector S of each image finds out t cluster centers that have the closest distance with the feature vector in the entire cluster center.

$$s(k)_e = s = \left\{ s_i \in \text{codebook} \mid \forall s_j \in \text{codebook} / (s_i \cup s(t))_m \right\}, \quad (5)$$

$$\|r - s_i\| \leq \|r - s_j\|, 1 \leq i \leq k, 1 \leq j \leq k, k \leq m \leq n$$

In the formula, $s(k)_e$ represents the h close cluster center vector with the vector s . The difference between s and $s(k)_e$ reflects the distribution of s after the mapping of the cluster center, and codebook is the code book vector formed by the aggregation of all the vectors in the cluster center.

Constant t represents the number of the clusters that most feature vectors belong to. When the value of t is too large, the affiliated clusters will be large, the time of clustering will be is long and the clusters with a remote distance from the feature vector will become meaningless for their too small membership weights. If the value of t is too small, it cannot find out the actual subordinate relation between the feature vector and the cluster. Here, the value of t is set to 4.

Calculate the membership weights of feature vector S in the t closest clusters, and use the membership weights to calculate the difference between S and the t closest cluster centers. The collection of the differences between all the local feature vectors in one image and their closest cluster center is namely the SA-VLAD descriptor of the image. sk_i is the vector of d dimension, which stands for the sum of differences of the image's SA-VLAD descriptor in the i cluster center position. The value of sk_i is as follows:

$$sk_i = \sum \frac{1}{t_i(s)} (s - t_i) \quad (6)$$

$$sk = (sk_1, \dots, sk_1, \dots, sk_i) \quad (7)$$

sk is the SA-VLAD feature value of the image.

SA-VLAD can be directly involved in the sequential retrieval or organize the sequential retrieval by using the index mechanism. However, because SA-VLAD has a high dimension, the dimension reduction is needed before indexing. And it can be known from the characteristics of the image itself that during the category distribution, the image's many local feature vectors, such as SIFT, will be assigned to the same cluster, which means a large number of SIFT vectors have the same adjacent cluster center, and there are clusters without any feature vectors. After the calculation of soft assignment, there may be a large number of consecutive zero values in the cluster center position without any feature points. Thus the linear dimension reduction method can be used to reduce dimension for the high-dimension SA-VLAD. The section 4 of the experiment shows that, to reduce dimension with the same amplitude, SA-VLAD has a smaller loss of precision than VLAD, and the loss of the average precision is even smaller at query time, which states that SA-VLAD has a better ability to resist dimension reduction. After the dimension reduction, the aggregation descriptor can be sequential scanned and queried or be indexed and queried by using any index mechanisms that is suitable for the feature vector.

3. The Inverted Index Mechanism Based on Scattered Assignment

The index mechanism in the core of the content-based image retrieval, DA-IVFAD index mechanism makes up for the inadequacy of IVFADC mechanism, realizes the simplified coding, effective distance calculation and high-efficient inverted chain storage of the feature vector.

A. The Generation of the Inverted File with Asymmetric Distance Computation Based on Scattered Assignment

(1) Quantize all of the database vectors and carry out the scattered assignment. Use the k-means method in advance to cluster the training set, obtaining k clusters and their cluster center vectors, each cluster is considered to be the last inverted index chain in the storage structure. For the input database vector y , n cluster centers that have the nearest distance with y are obtained.

The value field of constant n is [2, 4], the too big value of n will increase the calculation amount of the distance. Here, the value of n is set to the minimum value 2, which means that the maximum number of the database vectors assigned to the cluster is 2. At this moment, obtain the cluster center t_1 which has the most nearest distance with vector y and the cluster center t_2 which has the second nearest distance with vector y .

(2) Calculate $c(x, t_1)$ and $c(x, t_2)$, and determine how to assign the database vectors to the two clusters. If $c(x, t_2) - c(x, t_1) \geq \alpha$, only assign the database vectors to the most nearest cluster and calculate the residual vector $t_1(x)$. Otherwise, assign the database vectors to the two clusters at the same time and calculate the residual vectors $t_1(x)$ and $t_2(x)$.

The value of σ is a threshold value, which will greatly increase the storage space if it is too large, and make DA-IVFADC degrade to IVFADC if it is too small. The purpose of the σ value is to assign the objects which are located between the clusters or have very small distances with the cluster center to multiple clusters at the same time, to provide convenience to the subsequent queries.

The residual vector r is the difference between the original vector and its overall quantized cluster center, the residual vector of vector y is as follows:

$$t(x) = x - E(x) \quad (8)$$

(3) Quantize the residual vectors and obtain $E(t(x))$, respectively add the obtained codes, images or vector ID into the corresponding inverted index link.

The idea of product quantification coding is the "subspace combination coding". In order to avoid the high cost of time and space brought out by the overall quantification when there are too many code words in the code book, the product quantification evenly divides the high-dimension vectors into multiple subdivisions, which are the division of subspaces. Carry out the independent quantification in the subspace, and then combine the code words in the subspace to obtain the code book of the entire space. t is a D-dimension vector, $t = \{x_1, x_2, \dots, x_D\}$, which is evenly divided into m subdivisions each with a dimension of f' , and the quantiser of each subdivision is r_j .

$$r_j = (t_1(x), \dots, r_n(x)) \quad (9)$$

It can be known from formula (9) that each subquantiser r_j , such as the mean clustering, produces independent code words $t_{j,i}$ and the code book h_i in i sunspaces. All the

subspaces produce the same number of code words with a dimension of f' , and the code book of the entire space is the Cartesian product of all the subspace's code books.

$$h = h_1 \times h_2 \times \dots \times h_n \quad (10)$$

It can be known from formula (10) that the code words of the whole space might be the any combinations of any code words in m subspaces. Take k-means as an example, each subspace produces h_k code words. From the angle of combination, the code word of a D -dimension vector in the i subspace might be one of the code words in l' , and the coding of the entire vector is the arrangement of the subdivisions from $1-n$. From the perspective of permutation and combination, the number of the code words for the whole space is as follows:

$$l = (l')^n \quad (11)$$

When $n = w$, each dimension of the vector is seen as a subdivision, which carries out the quantification independently, and is transferred into the quantification of the real space. When $n = 1$, it is the traditional way of overall quantification, which means to carry out the quantification in the entire vector space.

The product quantifications of DA-IVFADC and IVFADC are based on the entire residual vectors of the original vectors, which mean that in the whole framework there is only one time of product quantification. The experiment shows that one single time of product quantification still can obtain good query results, which can save the storage space compared to design product quantification for the residual vectors of each cluster. If each cluster uses different product quantifications to generate different cluster centers, then m subspaces will have m quantisers and there are $n \times l' \times f'$ float-point data need to be stored.

Set the overall quantification for U , the product quantification for m , then the approximate value of database vector y is expressed as follows:

$$\tilde{z} = U(x) + m(x - U(x)) \quad (12)$$

Database vector y can be shown by the binary group $U(x), m(x - U(x))$ of the overall quantification and the product quantification. After the overall quantification, $U(x)$ determines the chain table that the vector belongs to, and the product quantification represents the code of the vector.

After the end of the indexing process, each database vector is identified by using ID and code. ID is the only identifier of the vector in the database, and code is the combination of the cluster centers obtained after the vector's overall quantification and product quantification. ID and code are the eventually expressions of the database vector is the inverted chain table, as shown in Table 1.

Table 1. The Vector Coding of da-ivfadc

Region	Length / bit
ID	8-34
Code	$m \times [\log_2 f']$

C. The Query of the Inverted File with Asymmetric Distance Computation Based on Scattered Assignment

The query process of DA-IVFADC is similar to its indexing process, it also has the process of overall quantification, but there is only product quantification for the database vectors, no product quantification for the residual vectors of the query vectors is carried out. The distance calculation between the vectors after quantification and the vectors without quantification is called the asymmetric distance computation, as shown in Figure 3.

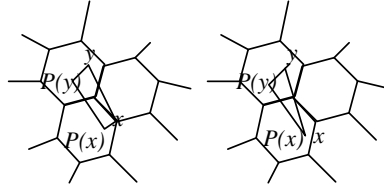


Figure 2. The Symmetric Distance Computation (Left) and the Asymmetric Distance Computation (Right)

y is the database indexing vector, and x is the indexing vector. $U(x)$ and $U(y)$ are respectively the cluster center that x and y belongs to, they are the representative vectors of x and y after the quantification. In the asymmetric distance calculation ADC, after the quantification the database feature vector y is expressed with $U(y)$, while there is not any quantification measures for the query vector x , the distance between y and x is represented as follows:

$$\bar{t}(x, y) = t(x, U(y)) = \sqrt{\sum_{j=1}^n t(t_j(x), U_j(t_j(y)))^2} \quad (13)$$

It can be seen from Figure 3 that the symmetric distance computation (SDC) is to calculate the distance between the query vectors after quantification and the database vectors after quantification. Compared with the asymmetric distance computation, the advantage of SDC is that it codes the query vectors, which saves the space. But the use of asymmetric distance computation can better reflect the similarity between the objects and reduce the distance error brought out by quantification. The query process based on DA-IVFADC is shown in Figure 4.

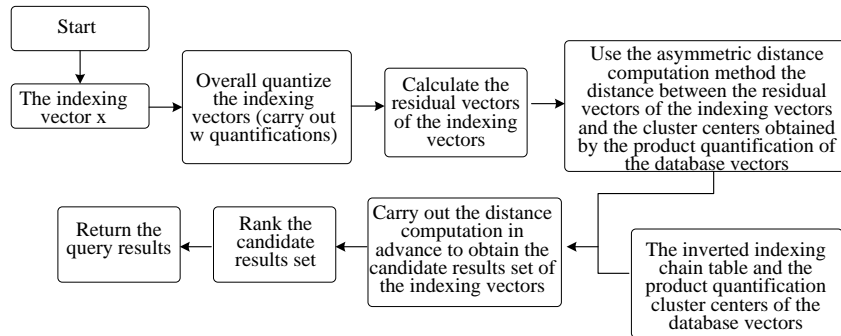


Figure 3. The Query Schematic Diagram of DA-IVFADC

4. The Simulation Experiment and Analysis

A. The Experimental Platform

The software and hardware environments uses in the experiment are as follows: the hardware environments are Intel(R) Core(TM) 2Quad CPU Q8230@ 2.34GHz, 4GB

memory; the operating system is Windows XP; the coding environments are Matlab 7.10.1(R2012a) Microsoft Visual Studio 2007.

In order to validate DA-IVFADC and SA-VLAD, two databases are prepared for DA-IVFADC. Each database contains three subsets, the training set, indexing data set and query data set. Table 2 shows the specific situations of the database vector's dimension and size.

Table 2. The Test Database of DA-IVFADC

Name of database	Dimension of feature vectors	Size of training set / piece	Size of database set / piece	Size of retrieval library / piece
SIFTSMALL	256	26000	20000	103
SIFT	256	200000	200000	20000
RANDOM	128	26000	20000	103

B. The Analysis of the Experimental Results

The Effects of product quantification parameters m and f' on the Results

The two important parameters of product quantification are subspace number m and the cluster number of product quantification f' , which determine that the coding bit of the vector is $L_{code} = m \times \log_2 f'$.

ADC and SDC are two different calculation ways, which respectively constitute two different index mechanisms combined with product quantification. The non-overall quantification process carries out the product quantification for the original vectors directly, and the candidate result is the whole database rather than a similar data that falls into the candidate w chain tables in the DA-IVFADC mechanism. Under the query mechanism without introducing the inverted table, the effects of m and f' on the results recall@ R can be seen clearly. Based on SIFTSMALL, figure 5 and figure 6 are respectively the experimental results figures of recall@100, when there are $m = \{1, 2, 5, 7, 18\}$ and $f' = \{26, 28, 30\}$ in the ADC and SDC index mechanisms.

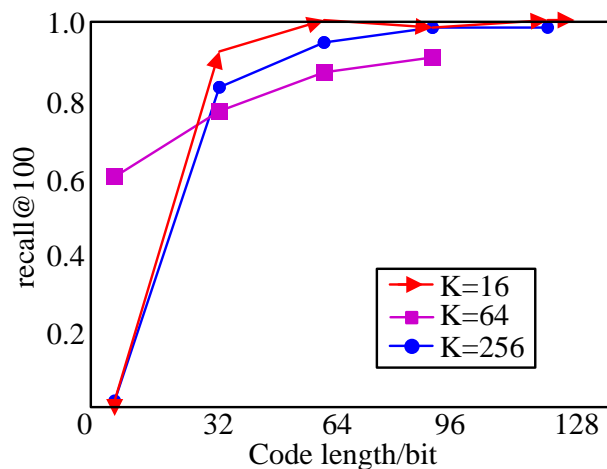


Figure 4. Effects of Parameter Settings on Recall@100 in ADC Index Mechanism

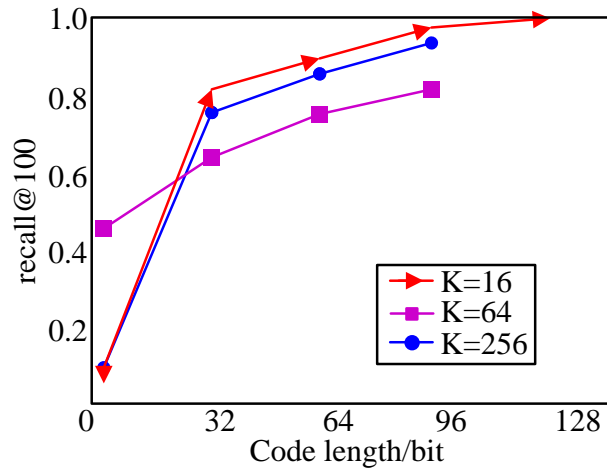


Figure 5. Effects of Parameter Settings on Recall@100 in SDC Index Mechanism

It can be seen from Figure 5 and figure 6 that under the condition of the same coding length, namely the same L_{code} , the number of the subspace cluster centers f' is big, while the number of the subspaces m is small, which can obtain a better effect compared to the situation when the number of the subspaces is big, while the number of the cluster centers is small. The coding bit is 64, when $m=8, f'=256$, the recall@100 which is a triangle line is higher than the recall@100, which is a circular line when $m=18, f'=18$.

When the parameters are the same, the recall@100 value of ADC is greater than that of SDC. In figure 5, the value of recall@100 is close to 1 when $m=8, f'=68$, while in figure 6, the value of recall@100 is close to 1 just when $m=8, f'=256$, and at this moment the storage space for the product quantification cluster centers of SDC is far bigger than that of ADC. The superiority of ADC proves that it is a wise choice to only quantize the data in the database and not quantize the query data in the IVFADC and SA-IFADC index mechanisms.

Aiming at the semi-structured feature data, the produce quantification theory proves that it can guarantee a better retrieval performance to carry out the regular subspace partition for the high-dimension vectors according to the independence between the vector dimensions. It can be known from the SIFT feature extraction method that when $m=8$, each subspace in product quantification has a dimension of 16, and each 16 dimensions of SIFT feature happens to be a set, which is in accordance with the principle that each subspace has a high independence and balanced energy. The experimental results that the recall@100 values of SIFT feature in ADC and SDC are all close to 1, when $m=8, f'=256, L_{code}=68bits$. Thus, in DA-IVFADC the value of m is set to 8 and the value of f' is set to 256.

The Effects of f^*, m and f', w on the Results in DA-IVFADC

w is the number of the candidate chain tables at query time, m is the number of the product quantification subspaces, f' is the number of the cluster centers of each subspace, and f^* is the number of the clusters during the overall quantification, namely the number of the chain tables.

Figure 7 shows the recall@100 data of the SIFT database, the data are recorded when the subspace cluster centers in the DA-IVFADC product quantification are $f'=256$,

$m = \{1, 2, 5, 7, 18\}$, $f^* = \{1024, 4096\}$, $w = \{1, 4, 8, 64\}$, the parameter of scattered assignment is $n = 2$, and the distance threshold is $\alpha = 1000$.

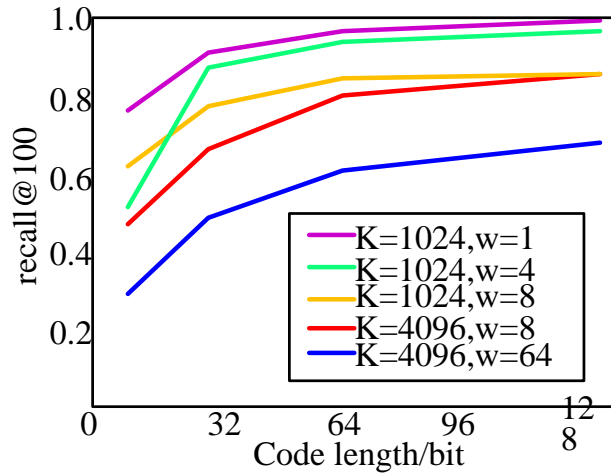


Figure 6. Effects of the Parameters on recall@R in DA-IVFADC Index Mechanism

It can be seen from the curve changes of figure 7 that when the values of f^* and w are determined, the increasing of the code bits cannot obviously improve the query efficiency, and the curve approaches to gentle after the value of m is up to 8, the length of code reaches at 64 bits. The reason is that after the value of f^* is determined, the database objects in each chain table have already been determined, and after the value of w is confirmed, the candidate result objects are confirmed too. When the value of m increases to a certain degree, all the similar objects fall into w chain tables have already returned, and the rest of the neighbor objects in the database have already been filtered out after the confirmation of m , for not falling into any one of the w chain tables, which makes the curve gradually gentle. When the value of f^* is determined, the greater the value of w is, the higher the recall ratio will be, and the recall@100 increases correspondingly. But it is not the bigger w value the better, because with a bigger value of w , there will be a bigger time cost when calculating the distance between the residual vectors in the object subspace and the cluster center of the candidate chain table subspace, which will have impact on the query efficiency. The greater the value of f^* is, the finer the partition of the database vectors will be, and to ensure the quality of the results, the value of w must be increased. If the value of f^* is too small, it will degrade to ADC query, which makes it cannot give full play to the finer classification overall quantification and the accelerated query function of the inverted chain. Therefore, the parameters of DA-IVFADC need to be designed according to the actual circumstances of the database.

The Comparison between DA-IVFADV and IVFADC

Figure 8 is the recall@100 comparison diagram of IVFADC and DA-IVFADC when there are $m = 8$, $f' = 256$, $L_{code} = 68bits$, $f^* = 65$, $w = \{1, 2, 10, 16\}$ in RANDOM test library, because the curve basically has no changes after the value of R is greater than 20, only the part when the value of R is less than 20 is drawn. Figure 9 is the corresponding query time comparison diagram. It can be seen by combining the two figures that with the same value of w , the difference of query time is very little, but the accuracy of DA-IVFADC is obviously

higher than that of IVFADC. When there are $w=16$, $time=13.1$, the value of $recall@20$ is 0.99 in IVFADC, and when there are $w=10$, $time=8.4$, the value of $recall@20$ is more than 0.98 in DA-IVFADC.

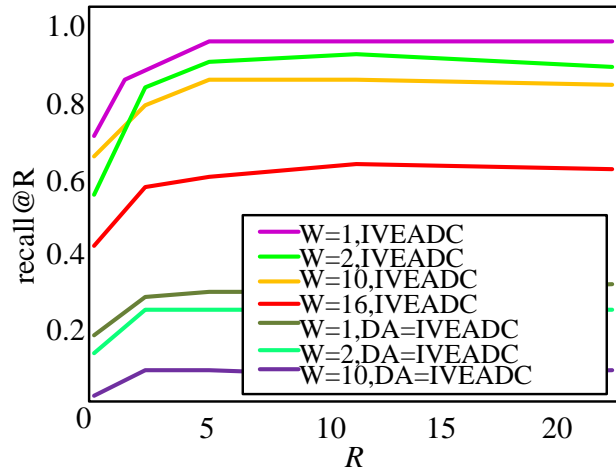


Figure 7. The Value of Recall@ R in RANDOM Database

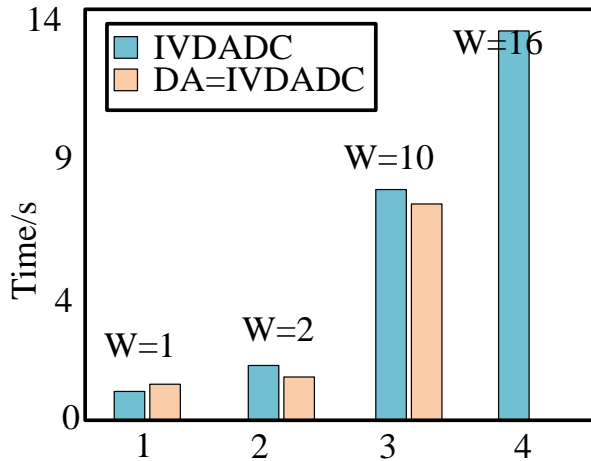


Figure 8. The Query Time Diagram of 500 Retrieval Objects in RANDOM

Figure 10 is the $recall@22$ comparison diagram of IVFADC and DA-IVFADC when there are $m=8$, $f'=256$, $Lcode=68bits$, $f^*=1024$, $w=\{1,2,8,16\}$ of the SIFT feature vectors in the 1MB test library, and figure 11 is the corresponding query time comparison diagram. Figure 9 counts the query time of the 500 retrieval objects, while figure 11 counts the average query time of 10000 retrieval objects. It can be seen by combining the two figures that DA-IVFADC still has a better performance than IVFADC under the condition of large amount of data.

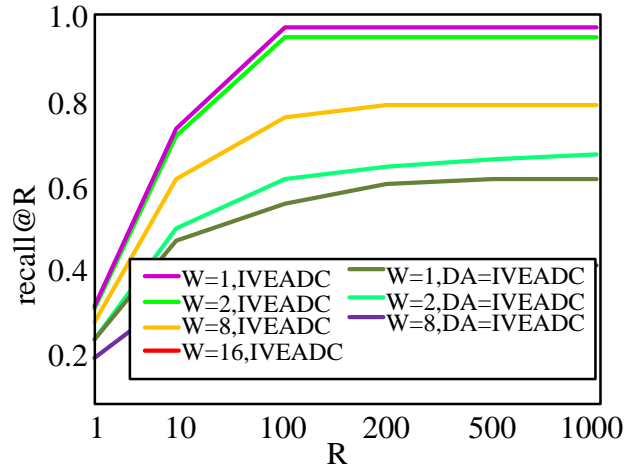


Figure 9. The Value of Recall@ R in SIFT Database

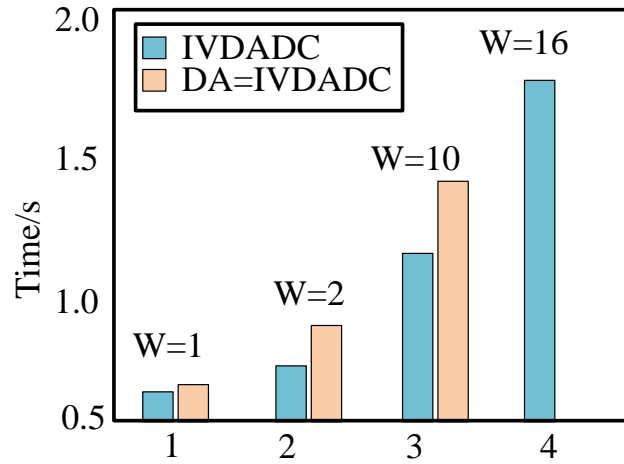


Figure 10. The Average Query Time Diagram of SIFT Database

5. Conclusion

Aiming at the problem of massive image data and “dimension curse”, this paper proposes the BOF-based image retrieval improved algorithm. Through the experiment it is verified that the improved algorithm in this paper improves the image query performance of the database.

References

- [1] X. Huang, X. Ma, B. Chen, A. Markham, Q. Wang and A. W. Roscoe, “Human Interactive Secure ID Management in Body Sensor Networks”, *Journal of Networks*, vol. 7, no. 9, (2012), pp. 1400-1406.
- [2] M. Yang and L. Zhang, “Gabor feature based sparse representation for face recognition with gabor occlusion dictionary”, *ECCV*, vol. 448, no. 61, (2010).
- [3] J. M. Morel and G. Yu, “ASIFT: A new framework for fully affine Invariant Image Comparison”, *SIAM Journal on Imaging Sciences*, (2009).
- [4] B. Zhang, C. Wang and B. Xiao, “Based on Bag-of-phrases image”, *Journal of automation*, vol. 38, (2012), pp. 46-54.

- [5] Y. Zheng, M. Zhao and S. Neo, "Visual synset: towards a higher-level visual representation", in CVPR, IEEE, (2008), pp. 1-8.
- [6] Y. Cao, H. Zhang and Y. Gao, "Matching Image with Multiple Local Features", *ICPR*, (2010), pp. 519-522.
- [7] J. Hare and P. Lewis, "Automatically annotating the mir flickr dataset", in Proceedings of the 2nd ACM international Conference on Multimedia information Retrieval, (2010).
- [8] S. Amir, I. M. Bilasco and M. H. Sharif, "Towards a unified multimedia metadata management solution", *Intelligent Multimedia Databases and Information Retrieval: Advancing Applications and Technologies*, IGI Global, (2010).
- [9] T. Li and L. Li, "Music Data Mining: An Introduction", CRC Press, (2011).
- [10] W. Lee, S. Verzakov and R. Duin, "Kernel combination versus classifier combination", *Multiple Classifier Systems*, (2007), pp. 22-31.
- [11] Y. Zhu, Y. Chen and Z. Lu, "Heterogeneous transfer learning for image classification", in AAAI, (2011).
- [12] SJ. Pan and Q. Yang, "A survey on transfer learning", in *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, (2010), pp. 1345-1359.
- [13] Q. Hu, M. Yeh and K. Cheng, "Multimodal fusion using learned text concepts for image categorization", in proceedings of the 14th Annual ACM International Conference on Multimedia, (2006), pp. 211-220.
- [14] J. Yuan, Y. Wu and M. Yang, "Discovery of collocation patterns: from visual words to visual phrases", in CVPR, IEEE, (2007), pp. 1-8.
- [15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International journal of computer vision*, vol. 60, no. 2, (2004), pp. 91-110.
- [16] C. Carson, S. Belongie and H. Greenspan, "Blobworld; Image segmentation using expectation-maximization and its application to image querying", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, (2002), pp. 1026-1038.

Author



Liu Haihong, (1972), male, master, lecturer, research areas: cloud computing, network multimedia

