SVD-Based Hierarchical Algorithm for Similarity Indexing in Quadratic Form Distance Space

Luo Ming, Bai Xuesheng, Xu Guangyou

Abstract-- Search based on content similarity in large multimedia library is essentially K-nearest neighbor search in high-dimensional feature spaces. In order to address the main issue influencing the real time property of similarity search for quadratic form distance – high dimension, this paper presents a hierarchical similarity indexing algorithm based on SVD (Singular Value Decomposition) technology, which first does considerably cheap approximate searching in the most significant subspace determined by SVD of the similarity matrix for quadratic form distance function based on similarity indexing structure, and then does linear exact searching in the full high-dimensional feature space on the results filtering through the first step. Experiments on a large (size>10,000) indexable image database demonstrate the effectiveness and efficiency of our approach even in high dimensions such as 512 and 256.

Index Terms– similarity indexing, feature space, quadratic form distance, SVD （Singular Value Decomposition）.

I. Introduction

In recent years, multimedia content-based retrieval has become an important research problem. Most researchers use feature space to describe and handle this problem. Multimedia objects are represented using vectors in a certain multi-dimensional feature space. Thus, similarity search is essentially K-nearest neighbor search in multi-dimensional space.

Low dimensional indexing techniques that have traditionally been used for data retrieval is not adequate for multimedia data, and hence similarity indexing techniques for higher dimensional data have been developed, such as R*-tree, TV-tree, SS-tree and X-tree [1]-[7]. However, there is often a very high dimension demand (256-4096) for vectors to exactly describe features of objects, such as color histogram and sketch. This problem of high dimension (called "dimensionality curse" by researchers) is prohibitive while using a quadratic form distance function as the measure of feature space, which is more suitable for human's vision. Even similarity indexing techniques are not enough for disposing of such queries in real time. Some methods have been proposed to solve the problem of high dimension [8]. They do approximate searching in low dimensional subspace to get acceptable efficiency. Researches on this aspect are quite active, but most of them are based on simple linear searching, which limits the improvement of system performance.

With combination of hierarchical idea and similarity indexing structure, this paper proposes a hierarchical similarity indexing algorithm for quadratic form distance based on SVD (Singular Value Decomposition) technology. We create similarity indexing structure (such as SS-tree) in the full dimensional feature space. When disposing of similarity search queries, system first does approximate searching in low dimensional subspace determined by SVD of the similarity matrix of quadratic form distance function on similarity indexing structure, and then does naive linear searching using the exact quadratic form distance function on the results filtering through the first step. We built a large (size>10,000) indexable image database and conducted experiments on it using 512-dimensional color histogram feature and 256-dimensional sketch feature. Experiment results unveil significant speed-up over systems without using hierarchical approximation idea or similarity indexing structure.

In section II we explain quadratic form distance and introduce the motivation and processing of our hierarchical algorithm. In section III we expatiate on the property of matrix's SVD, which is the key for our approach. The experiment results and analysis are shown in section IV. Section V concludes the paper.

II. Motivation and Algorithm

A. Quadratic Form Distance

Let x and y be N-dimensional vectors representing a multimedia object each. For retrieval based on similarity of objects, a distance between the two vectors can be defined as a match measure of them. Euclidean distance function as (1)

Manuscript received June 16, 2001. This work was supported in part by the State " 863" High-Tech Funding, P. R. China, under Grant No. 863-306-ZT04-02-01 and 863-306-QN99-01.

Luo Ming, is now a graduate with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084 P.R.China. (e-mail: mingluo@media.cs.tsinghua.edu.cn).

Bai Xuesheng, is with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084 P.R.China. (e-mail: xsbai@media.cs.tsinghua.edu.cn).

Xu Guangyou, is with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084 P.R.China. (e-mail: xgy-dcs@mail.tsinghua.edu.cn).

and quadratic form distance function as (2) are used commonly.

$$d(x,y)=(x-y)^T(x-y) \qquad (1)$$

$$d(x,y)=(x-y)^T A(x-y) \qquad (2)$$

where $A=[a_{i,j}]$ is a matrix and the weight $a_{i,j}$ denotes similarity between the ith component and jth component in vectors. These weights are all in range of [0,1], and the more similar are two components, the larger is the weight. Euclidean distance is a particular example for quadratic form distance. For Euclidean distance, A is an identity matrix, thus the similarity between two different components of vectors is neglected. While other quadratic form distance functions define a certain similarity weight for every two components of vectors. So quadratic form distance function with appropriate matrix A is more convictive in describing similarity of two vectors, and thus two objects. The similarity matrix A for quadratic form distance must be PSD (Positive Semi-definitive) to guarantee the distance always makes sense, i.e. expression (3) must be satisfied:

$$\forall x,y\in E^d,\ d(x,y)=(x-y)^T A(x-y)\geq 0 \qquad (3)$$

B. Hierarchical Algorithm

In human's perspective to multimedia objects, different components of feature vectors have some similarity to a different extent. For example, in color histogram feature, the similarity between blue bin and purple bin is larger than that between blue bin and yellow bin. Therefore, quadratic form distance with different similarity weights for different components better corresponds to human judgment of visional similarity. It is fit for similarity measure.

But the computational complexity of quadratic form distance is *O(N^2)*, with N as the dimension of feature space. It is an unacceptable computational complexity when the size of multimedia library is beyond 10,000 and N is beyond 256. This paper proposes a hierarchical algorithm according to the thought of lower-dimension filtering. We construct a low dimensional distance function d_k ($k<N$) to approximate the full dimensional distance function d_N, and use it to filter the searching set on similarity indexing structure, with expression (4) is satisfied:

$$\forall x,y\in E^N,\ d_k(x,y)\leq d_N(x,y) \qquad (4)$$

Expression (4) guarantees that the filter is secure without error misses. Naive linear searching is done on the filtering results using d_N as measure function.

The first step of hierarchical algorithm (filtering) has computational complexity of *O(k*N)* on every target matching, which is much smaller than that of using N-dimensional quadratic form distance function directly - *O(N^2)*. And expensive N-dimensional quadratic form distance computation is only done on a small fraction of the total data set in the second step. The smaller is k, the more efficient and less effective is the filter, making the first step's time down and the second step's time up. So, we should choose appropriate k to get the best performance (generally k is selected one or several numeric scales smaller than N).

The SVD of a matrix has fairly good property (see next section), which assures that the approximation of the similarity matrix based on SVD provides such a low dimensional distance function d_k as we seek. Fig. 1 shows the diagram of system modules and data flow.

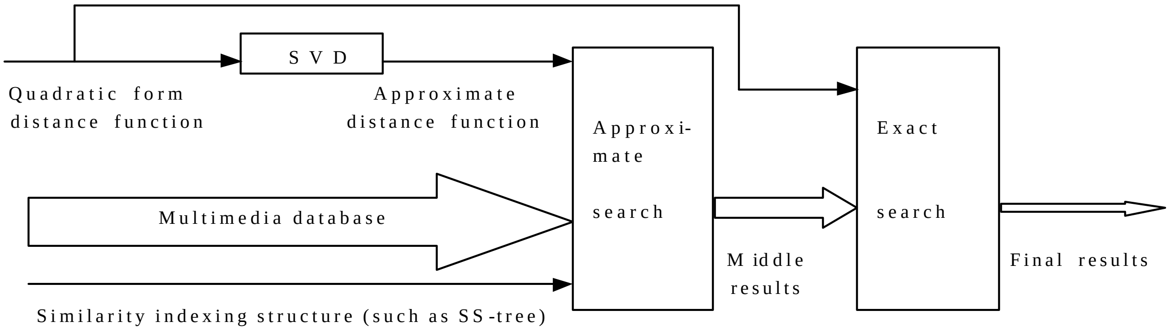

SVD
Quadratic form distance function
Approximate distance function
Approxi-mate search
Exact search
Multimedia database
Middle results
Final results
Similarity indexing structure (such as SS-tree)

Fig. 1. System modules and data flow

III. SIMILARITY MEASURE BASED ON SVD APPROXIMATION

Denote A as the similarity matrix for N-dimensional quadratic form distance, and A_k as a k-rank approximation matrix for A. So there are equations as (5) and (6):

$$d_N(x,y)=(x-y)^T A(x-y) \qquad (5)$$

$$d_k(x,y)=(x-y)^T A_k(x-y) \qquad (6)$$

To ensure correctness and effectiveness of the hierarchical algorithm, we seek A_k at which the following extremum is attained,

$$\inf_{A_k}(\sup_{x,y\in E^N}((x-y)^T(A-A_k)(x-y))) \qquad (7)$$

with restriction 1) and 2).

1) A_k is PSD

2) $A-A_k$ is PSD

Extremum (7) means A_k is the closest approximation of A at rank k. So the most powerful filter at rank k is attained, which guarantees the effectiveness of hierarchical algorithm. Restriction 1) and 2) are determined by expression (3) and (4) respectively, which guarantee the correctness of hierarchical algorithm. Fortunately, SVD of matrix A provides a constructive answer to the extremum problem. Let SVD of A is:

$$A=U\Sigma V \qquad (8)$$

where $\Sigma=diag(s_1,......,s_N)$, $s_1\ge s_2\ge......\ge s_N\ge 0$, which are called singular values of A, and U and V are both orthogonal matrices. Thus, the k-rank approximation based on SVD is the solution to the extremum problem above (see [9]):

$$A_k=U_k\Sigma_k V_k \qquad (9)$$

where $\Sigma_k=diag(s_1,......,s_k)$, U_k is the matrix whose columns are the first k columns of U, and V_k is the matrix whose rows are the first k rows of V.

The essence of k-rank approximation based on SVD is projecting vectors in N-dimensional space to the most significant k-dimensional subspace determined by the k largest singular values of similarity matrix. We use cheap distance in the subspace to approximate expensive distance in the full space as a filter in the first step. This filter is not only efficient and effective but also secure because of the good property of SVD.

IV. EXPERIMENT RESULTS AND ANANYSIS

We implemented a content-based retrieval system on image database on Windows platform with Java applet in WWW browser as query interface. The system accomplishes similarity search on the database composed of 11,804 multifarious images using dominating color, texture, color histogram, color distribution and sketch as features. Among them, only color histogram and sketch are measured by quadratic form distance, others are simple enough to be measured by Euclidean distance fairly well. SS-tree is used as similarity indexing structure to organize the database.

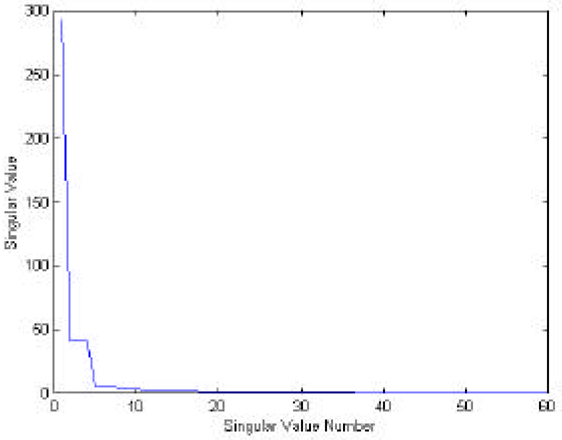
A. Selection of Quadratic Form Distance

We use our hierarchical algorithm when query is based on color histogram similarity or sketch similarity. Feature extraction modules developed by us generates 512-dimensional color histogram feature vectors in HVC (Munsell) color space and 256-dimensional sketch feature vectors. For these two kinds of features, similarity matrix $A=[a_{i,j}]$ is both determined as following expression:

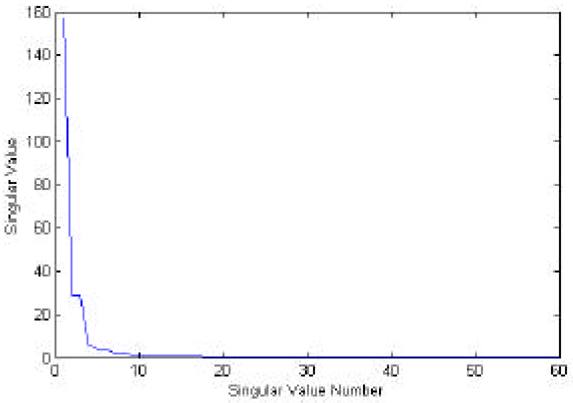
$$a_{i,j}=1-d_{i,j}/d_{\max} \qquad (10)$$

where $d_{i,j}$ is the Euclidean distance of the ith bin and the jth bin, and $d_{\max}=\max_{i,j}\{d_{i,j}\}$.

Such a quadratic form distance function implicates the similarity of bins of feature. On the other hand, for sketch feature A always satisfies expression (3), and for color histogram feature A satisfies it under restriction of $\Sigma_i(x_i-y_i)=0$, which is confined by the property of color histogram feature (see [10]). The distributions of singular values of such similarity matrices are shown in Fig. 2. For the similarity matrix of color histogram feature, the 20 largest singular values range from 292.99 to 1.05, and the others are smaller than 1.00. For the similarity matrix of sketch feature, the 11 largest singular values range from 157.22 to 1.06, and the others are smaller than 1.00.



(a) For color histogram feature



(b) For sketch feature

Fig. 2. Distribution of singular values of similarity matrices

B. Experimental Evaluation

After creating similarity indexing structure using full dimensional quadratic form distance, we did image retrieval based on color histogram feature similarity and sketch feature similarity separately. We search the 20 images most similar to a query image using such 3 methods:

1) Hierarchical algorithm on similarity indexing structure (SVD approximation scales are k=20 for color histogram feature and k=11 for sketch feature, respectively)

2) None-hierarchical algorithm on similarity indexing structure

3) Hierarchical algorithm on linear searching (without similarity indexing structure and the approximation scales are the same as method 1)

To avoid the influence of other loads such as local loads and network loads during processing, we evaluate methods using the number of multiplications and additions in distance computation and that of IO instead of searching time. The experimental results are shown in Table I, the data in which are averages with query objects ranging over the data sets.

Table I. Performance Comparison between 3 methods

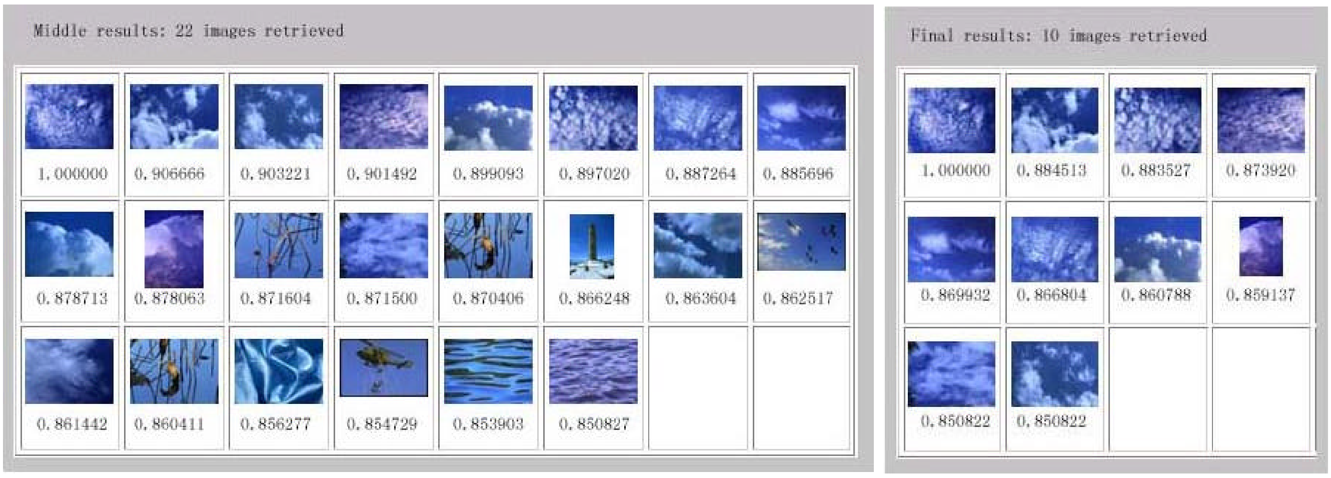
Data set	Searching cost	method 1	method 2	method 3
Real data set	Multiplications	212,632,110.4	4,849,300,493.7	252,703,840.0
(11,804 multifarious	Additions	207,192,206.7	2,424,650,246.8	247,224,880.0
images)	IO times	6,758.3	6,231.1	11,804.0
Uniform distributed	Multiplications	117,438,708.6	2,732,698,107.9	215,685,760.0
data set (10,000	Additions	112,091,586.0	1,366,349,054.0	210,242,880.0
objects)	IO times	2,002.1	2,002.1	10,000.0
Clustered distributed	Multiplications	390,113,947.8	9,077,464,356.9	2,083,005,760.0
data set (101,000	Additions	372,351,199.6	4,538,732,178.5	2,075,742,880.0
objects)	IO times	6,650.5	6,650.5	101,000.0

The uniform distributed data set and clustered distributed data set in Table I are artificial, used to approve performance of our method on typical and common data distributions. The former contains 10,000 random vectors in 512/256-dimensional feature space (randomicity implicating the uniformity in statistics sense), and the latter is produced as follows: choose 1000 random vectors as cluster centers (noted $C^i, i=1,......,1000$), and then produce 100 random vectors in every region of R^i as (11).

$$R^i = \{X \mid \forall 1 \le j \le 512, | X_j - C^i_j | \le 0.05\} \qquad (11)$$

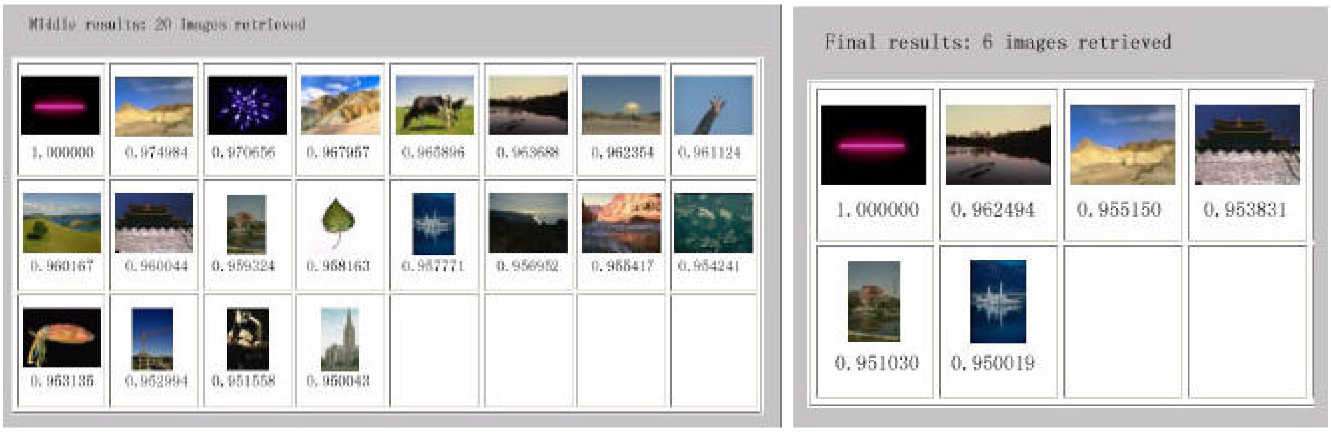
The experiment results show that our method (method 1) is much more efficient than other methods. Averagely, the number of multiplications and additions in distance computation of method 1 is 4% of that of method 2 and 52% of that of method 3. The IO times of method 1 is 104% of that of method 2 and 13% of that of method 3. Running on Windows2000/Pentium966 platform, it can accomplish a query in 7.5 seconds averagely using 512-dimensional color histogram feature and in 2.0 seconds averagely using 256-dimensional sketch feature on image database sized 11,804. It is an applicable real-time level. The traditional algorithm - none-hierarchical algorithm on similarity indexing structure (method 2) cannot complete such a task in real-time because of the extremely complex distance computation. Hierarchical algorithm on linear searching must have all entries in database read and computed on, so its performance gets sharply down with database size increasing.

Fig. 3 shows a real example of color histogram similarity-based retrieval using our method on a large image database. Fig. 4 shows a real example of sketch similarity-based retrieval using our method on the same image database. In Fig. 3 and Fig. 4, we retrieve all images whose similarity measures with a query image are beyond a threshold noted as “Th” (1.00 meaning identical). Middle result and final result are shown in descending order of similarity, from left to right and from up to down, with similarity labeled by number. The query image is the most top-left image.



(a) Middle result after the first filtering step (b) Final result

Fig. 3. Color histogram-based Image retrieval using hierarchical method



(a) Middle result after the first filtering step (b) Final result

Fig. 4. Sketch-based image retrieval using hierarchical method

In Fig. 3, Th=0.85. The final result shown in (b) is the same as that of direct search on the whole database. It got rid of images of leaves (image 11,13,18), monument (image 14), birds (image 16), and helicopter (image 20) with blue sky as background and images of blue cloth (image 19) and water (image 21,22). The images left in final result are all cloud images, which are closest to the query image at content described with color histogram. Noticing that the images order of (a) is not the same as that of (b), it is because the approximation for distance function is not monotonous, while it does not matter because the correctness of the filter is guaranteed by the good property of SVD. In Fig. 4, Th=0.95. Alike with Fig. 3, results of sketch-based retrieval show the hierarchical algorithm is effective and secure.

V. CONCLUSIONS

In this paper, we propose a hierarchical similarity indexing algorithm for quadratic form distance based on SVD technology. It combines the thought of hierarchical filtering and similarity indexing structure harmonically and solves the high-dimension problem for quadratic form distance effectively and efficiently. Experimental evaluation conducted on real system demonstrates that this technique is effective and efficient for various kinds of data and clearly outperforms the state-of-the-art methods.

In our future work, we plan to extend our new methods to even higher dimensions beyond 1024 or 4096 and other complex features. It needs seeking appropriate similarity matrix with more concentrating singular values. We would also like to analytically understand the relationship between the efficiency and effectiveness of the filter in the first step of hierarchical algorithm. Finally, we will explore techniques to support queries in interactive environments (e.g., relevance feedback) efficiently using the hierarchical similarity indexing algorithm.

ACKNOWLEDGMENT

We wish to thank reviewers for their valuable comments, which helped in focusing the paper on the key themes. We would also like to thank Haizhou Ai, Qing Wu, and Hangjun Ye for their good advices. This work was performed while Luo Ming was a graduate, Bai Xuesheng was a research staff member and Xu Guangyou was a head professor in the HCI and Media Integration Group, State Key Laboratory of Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University, Beijing, P. R. China. This work was supported by State " 863" High-Tech Funding, P. R. China.

REFERENCES

[1] N. Beckmann, H. P. Kriegel, R. Schneider and B. Seeger, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles". Proceedings of the ACM SIGMOD International Conference on the Management of Data, 1990, pp. 322-331.

[2] K. I. Lin, H. V. Jagadish and C. Faloutsos, " The TV-tree: An index structure for high-dimensional data" . VLDB Journal, 3(4), Oct. 1994, pp. 517-549.

[3] D. A. White and R. Jain, " Similarity indexing with the SS-tree" . Proc. 12th IEEE International Conference on Data Engineering, New Orleans, Lousiana, 1996, pp. 516-523.

[4] D. A. White and R. Jain, " Similarity indexing：Algorithms and performance" . Proceedings of the SPIE: Storage and Retrieval for Image and Video Databases IV, volume 2670, San Jose, CA, Feb. 1996.

[5] D. A. White and R. Jain, " Algorithms and strategies for similarity retrieval" , Technical Report VCL-96-01, Visual Computing Laboratory, University of California, San Diego, 9500 Gilman Drive, Mail Code 0407, La Jolla, CA 92093-0407, July 1996.

[6] X. Bai, " Content-based retrieval and relevant technologies" . Ph.D. thesis, Tsinghua University, Beijing, Mail Code 100084, China, 1998.

[7] S. Berchtold, D. A. Keim and H. P. Kriegel, " The X-tree: An index structure for high-dimensional data" , Proc. 22th Int. Conf. On Very Large Data Bases, Bombay, India, 1996.

[8] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner and W. Niblack, " Efficient Color Histogram Indexing for Quadratic Form Distance Functions" . IEEE transactions on pattern analysis and machine intelligence, 1995, VOL.17, NO.7: pp. 729~736.

[9] P. Dewilde and E. F. Deprettere, SVD and Signal Processing: Algorithms, Applications, and Architectures. Elsevier Science Publishing Co., 1988.

[10] T. Feder, " On a quadratic form for points in space". IBM Internal Note, 1993.