Efficient Image Retrieval With Multiple Distance Measures

Andrew Berman and Linda Shapiro

July 1, 1996

1 Introduction

Consider the following simple model of an image database system: The user presents the system with a query image and asks for all images in the database which are "similar" to the query. The system then uses a pre-defined distance measure to compare the query image to each image in the database. It then returns the images which have the smallest computed distance to the query image.

Some distance measures are computationally expensive to calculate. This cost can be prohibitive if the database of images is very large and the database system must compare the query image to each image in the database. Thus, it is desirable to reduce the number of total distance measure calculations performed for each query. For certain distance measures and data sets, indexing or clustering schemes can be used to reduce the number of direct comparisons. There are also schemes in the literature based on the triangle inequality which can reduce the number of distance measure calculations[4, 3, 1, 2].

The simple database system described above is inflexible in its use of a single distance measure. Given the wide variety of application domains and uses within those domains, it would be more appropriate to allow the user a choice of distance measures, or even allow the creation of a new distance measure at query time. Allowing this functionality may conflict with indexing schemes that pre-suppose a given distance measure. We know of no algorithm in the literature that allows runtime creation of a distance measure, yet does not require direct comparisons with all the elements in the search space.

We have designed and implemented a prototype database system that allows the user a great deal of flexibility in runtime distance measure selection, while also reducing the number of direct distance measure calculations between a given query object and the database elements. FIDS, or "Flexible Image Database System", is currently being tested on a database of approximately one thousand images. It allows the user to take polynomial combinations of several pre-defined distance measures to create new distance measures with which to query the database. FIDS then searches the database and returns images similar to the query.

FIDS can often return results without directly comparing the query image to more than a few dozen images.

2 Reducing Direct Distance Comparisons with The Triangle Inequality

2.1 Previous Work

There are several schemes in the literature[4, 3, 1, 2] which take advantage of the triangle inequality to reduce the number of direct object comparisons in a database search. The intuition behind the schemes is that the distance between two objects cannot be less than the difference in their distances to any other object. More formally, given objects I, Q, K :

$$d(I, Q) \geq |d(I, K) - d(Q, K)| \tag{1}$$

Thus, by comparing two objects to a third *key* object, a lower bound on the distance between the two objects can be obtained.

Burke and Keller[4] first proposed the idea of using the triangle inequality to reduce comparisons. Berman[3] and Baeza-Yates, et. al.[1] refined Burke and Keller's algorithm by creating a single set of keys to use for all the objects in the database. Barros, et. al., [2] successfully used a single set of keys and the triangle inequality in an real image database. What follows is a description of the general algorithm used in [3, 1, 2]:

Equation (1) can be extended naturally by substituting a set of objects, $(K_1, \ldots, K_m)$ for K as follows:

$$d(I, Q) \geq max_j |d(I, K_j) - d(Q, K_j)| \tag{2}$$

Consider a large set of database objects, $\{I_1, \ldots, I_n\}$ and a much smaller set of key objects, $\{K_1, \ldots, K_m\}$. Pre-calculate $d(I_i, K_j)$ for all $\{1 \leq i \leq n\} \times \{1 \leq j \leq m\}$. Now consider a request to find all database objects I_i such that $d(I_i, Q) \leq t$ for query image Q and threshold value t. We can calculate lower bounds on $\{d(I_1, Q), \ldots, d(I_n, Q)\}$ by calculating $\{d(Q, K_1), \ldots, d(Q, K_m)\}$ and repeatedly using equation (2). If we prove that $d(I_i, Q) > t$ then we immediately eliminate I_i from our list of possible matches to Q. We then do a linear search through the uneliminated objects, comparing each to Q in the standard fashion. This scheme involves $m + u$ distance measure calculations, and $O(nm)$ floating point and boolean operations, where u is the number of uneliminated objects. The hope is that $m + u$ is sufficiently smaller than n to result in an overall time savings.

A slightly modified scheme allows the return of the "best" match. In this scheme, images are compared in increasing order of their proven lower bound distances. An image is returned when its calculated distance is less than the proven lower bounds of the remaining images. Both schemes have been tested and shown to work with a variety of data objects. The efficiency of the algorithm is highly dependant on the selection of keys, the relative expense of distance measure calculation, and the statistical behavior of the distance measure over the set of database objects.

The Berman and Baeza-Yates papers also introduced the idea of combining all the pre-calculated distances into a *trie*. Use of this data structure can, in some circumstances, reduce the number of simple calculations to below $O(nm)$. Berman showed an $O(u \log n + n^{1-\epsilon})$ expected upper bound on simple calculations for using the Hamming distance on random binary strings. The efficiency of this method is highly variable depending on the statistical behavior of the distance measure.

2.2 Indexing with Multiple Distance Measures

The previous section detailed an indexing scheme for a given set of database objects $\{I_1, \ldots, I_n\}$, keys $\{K_1, \ldots, K_m\}$ and distance measure d. We now describe a brand new scheme with multiple distance measures:

Let $D = d_1, \ldots, d_p$ be a set of distance measures. For every triple $\{d_i, I_j, K_k\}$, we pre-calculate $d_i(I_j, K_k)$ and store the result. Given some query object Q and database object I, we can use these values along with the triangle inequality to calculate a set of lower bounds $L_1, \ldots, L_p$ such that $d_i(I, Q) \geq L_i$ for each $\{1 \leq i \leq p\}$.

Now consider a new distance measure d' that is of the form $d'(I, Q) = \sum_{i=1}^{p} C_i d_i(I, Q)^{E_i}$ with non-negative coefficients and exponents. Then $d'(I, Q) \geq \sum_{i=1}^{p} C_i L_i^{E_i}$. This can be seen by noting that substituting L_i for $d_i(I, Q)$ does not increase the corresponding term in the summation. Thus, by calculating $L_1, \ldots, L_p$, we can compute a lower bound on $d'(I, Q)$ without directly calculating $d'(I, Q)$ itself. We can then use these calculated lower bounds to eliminate images from direct comparison as in the previous section.

3 FIDS: Flexible Image Database System

We are developing a system called FIDS to test this retrieval method. FIDS is composed of a viewer, a database engine, and an index. The viewer is shown in Figure 3. The leftmost large image is the query image to be matched against the database. Below the two large images is the set of “fundamental” distance functions. The user loads a query image into the leftmost large image window and assigns coefficients and exponents to the distance functions. A checkbox is provided if the user wishes to de-select a distance function. Upon pressing the “Find Close Matches” button, FIDS then searches the database for the closest matches to the image. The best image is returned alongside the query image in the right large image window. The next sixteen best images are returned in the sixteen smaller image windows.

FIDS is designed to allow the easy insertion of new distance functions into the system. At present, the distance measures in the database consist of several color measures and texture measures. They differ from each other in graininess of the color distribution and grid

size. We intend to add standard distance metrics such as energy and entropy in the near future, as well as more exotic metrics such as one based on wavelets[5].

3.1 Work In Progress

The system will include a variety of image distance measures, including measures of color, texture, shape relationships and overall appearance. We intend that the measures span a wide variety of computational costs, from quick color histogram measures, to more expensive measures such as graph matching. We will test the efficiency of our indexing scheme over these measures as well as over combinations of these measures.

Given some set of base distance functions, and a method of combining them, is the space of creatable distance functions wide enough to meet the needs of users? The weighted sum of exponentials we have implemented was chosen for simplicity as a first step towards a larger investigation of how to usefully combine distance measures. Assuming a combination of distance functions exist that meets a user's particular need, there is also the question of how the user will find this correct combination. Minka and Picard[6] have created a system which can learn appropriate features for image retrieval from the user. FIDS can also be enhanced to allow it to create useful distance functions through interaction with the users. We intend to carry out a study with a variety of users to help us answer these questions.

References

[1] R. Baeza-Yates, W. Cunto, U. Manber, and S. Wu. Proximity matching using fixed-queries trees. In *Combinatorial Pattern Matching*, pages 198–212. Springer-Verlag, June

1994.

[2] J. Barros, J. French, W. Martin, P. Kelley, and M. Cannon. Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. In *IS&T/SPIE - Storage and Retrieval for Still Image and Video Databases*, volume IV, Jan 1996.

[3] A. Berman. A new data structure for fast approximate matching. Technical Report 1994-03-02, Dept. of Computer Science, University of Washington, 1994.

[4] W. A. Burkhard and R. M. Keller. Some approaches to best-match file searching. *Communications of the ACM*, 16(4):230–236, Apr 1973.

[5] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. Technical Report 1995-01-06, Dept. of Computer Science, University of Washington, 1995.

[6] T. P. Minka and R. W. Picard. Interactive learning using a "society of models". Technical Report 349, M.I.T. Media Laboratory Perceptual Computing Section, 1995.