Query Refinement for Multimedia Similarity Retrieval in MARS *

Kriengkrai Porkaew
Department of Computer Science
University of Illinois at Urbana-Champaign

Kaushik Chakrabarti
Department of Computer Science
University of Illinois at Urbana-Champaign

Sharad Mehrotra
Information and Computer Science Department
University of California at Irvine

1 Introduction

Advances in image processing, database management, and information retrieval has resulted in *content-based multimedia retrieval* to emerge as an important area of research. Typical content-based retrieval systems allow users to specify queries by providing examples of objects similar to the ones they wish to retrieve. Due to the subjective nature of retrieval, it is unlikely that the answers to the 'starting query' will satisfy the user's information need. Rather, among answers retrieved, the user may find one or more objects that are closer to what she has in mind compared to the original examples.

In the Multimedia Analysis and Retrieval System (MARS), we have explored query refinement techniques to modify the query based on the relevance feedback of the user on the retrieved objects. Query refinement in MARS consists of *query reweighting* (QR) and *query modification* (QM) techniques. QR learns the user's notion of similarity between objects and adjusts the weights of different components of the query. It has been studied in [5, 4]. QM, on the other hand, uses the feedback information to change the query representation to better suit the user's information need. In [5, 2] *query point movement* (QPM) approach to QM is explored in which a query is represented by a single point in each feature space. At each iteration, the query point is moved to the centroid of the points marked relevant by the user. In this paper, we study a different approach to QM based on query expansion (QEX) which, at each iteration, uses a clustering technique to identify a set of (one or more) objects to be added to the query representation. We study efficient query processing techniques to implement the QEX approach as well as efficient techniques to execute refined queries for both QEX and QPM models. Our experimental results show that query expansion significantly outperforms query point movement both in terms of retrieval effectiveness and execution cost for all visual features used in MARS.

*This work was supported by NSF awards IIS-9734300, and CDA-9624396; in part by the Army Research Laboratory under Cooperative Agreement No. DAAL01-96-2-0003.

ACM Multimedia '99 10/99 Orlando, FL, USA

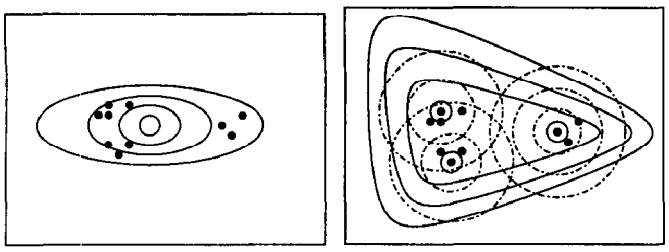
2 Content-Based Multimedia Retrieval in MARS

A *multimedia object* is represented as a collection of extracted features that describe its perceptual properties such as color, texture, and shape. Each feature can be viewed as a multidimensional space. A feature vector of an image is a point in the multidimensional space of that feature. A metric distance between the points is used to define the dissimilarity between the corresponding feature vectors.

A *query* is also represented as a collection of features. A user may use multiple objects as a query. In this case, we represent a query as a collection of features. Each feature consists of multiple instances. Each instance corresponds to the feature vector of each object in the query. Each component of a query is associated with a weight indicating the relative importance of that component compared to other components in the query. Similarity between the query and the object is computed as follows. Let Q be a query node and $F_1, \ldots, F_m$ (the feature nodes) be the children of Q. Let w_i be the weights of the feature nodes. Let $R_{i1}, \ldots, R_{in}$ (the object nodes) be the children of a feature node F_i. Let w_{ij} be the weights of the object nodes. Let Sim_{ij} be the similarity of an object O with an object node R_{ij} based on the i^{th} feature. The similarity Sim_i of O to Q with respect to feature F_i is defined as $Sim_i = \sum_{j=1}^{n} w_{ij} Sim_{ij}$ where $\sum_{j=1}^{n} w_{ij} = 1$ and the overall similarity Sim of O to Q are defined as $Sim = \sum_{i=1}^{m} w_i Sim_i$ where $\sum_{i=1}^{m} w_i = 1$.

3 Query Refinement in MARS

When a user submits an initial query, the system returns a ranked list of answers in the decreasing order of similarity to the query. Only the top few answers are returned. Subsequently, the user marks the answers she considers relevant to the query and submits her feedback to the system. The query refinement models exploit the multiple levels of relevance input by the user to improve



(a) Query Point Movement (b) Query Expansion

Figure 1: Query Refinement Approaches: the figure shows iso-similarity contours in a given feature space.

the query in subsequent iterations. MARS refines the query in two ways: *query reweighting* (QR) and *query modification* (QM) as described in Section 1. Both refinement mechanisms are combined seamlessly. In this paper, we concentrate only on query modification and the techniques developed can be easily integrated with query reweighting techniques discussed in [4, 5]. MARS supports two query modification approaches as described below.

Query Point Movement (QPM): QPM allows only a single object per feature as the query. When the user uses multiple examples to construct the query, the centroid is used as the single point query. Similarly, at each iteration of relevance feedback, when a user marks several objects as relevant, the weighted centroid of the feature value of the relevant objects for each feature space is used as the refined query. The weights used are obtained from the level of relevance provided by the user. Let $E_1, \ldots, E_i, \ldots, E_n$ denote the n objects marked relevant by the user. For simplicity, we assume that the E_i's also denote the corresponding feature vectors. Let $w_1, \ldots, w_i, \ldots, w_n$ be the corresponding levels of relevance. Let $E_i[j]$ denotes the value of point E_i along the j^{th} dimension of the feature space, $1 \leq j \leq m$, m being the dimensionality of the feature space. The weighted centroid C is defined as: $C[j] = \frac{\sum_{i=1}^{n} w_i E_i[j]}{\sum_{i=1}^{n} w_i}$.

Besides changing the location of the query point in the feature space corresponding to each feature F_i, QPM adjusts the distance function used to compute the distance of the query from the objects in F_i's feature space. This is achieved by associating weights with each dimension of the feature space of F_i. The weights assigned are inversely proportional to the standard deviation of feature values of the relevant objects along that dimension. Intuitively, among the relevant objects, the higher the variance along a dimension, the lower the significance of that dimension [5]. Figure 1 (a) shows how QPM refines the query. The figure shows contours representing equidistant ranges from the new query.

Query Expansion (QEX): QEX allows multiple objects per feature as a query. Such queries are referred to as *multipoint queries*. When the user marks several points as relevant, a small number of good representative points are selected to construct the multipoint query. For this purpose, we cluster the set of relevant points and choose the centroid of the clusters as the representatives. The details of the clustering algorithm used can be found in [3]. Note that in constructing the query, objects deemed relevant during previous iterations are also incorporated into the clusters. Implicitly, relevant points get added while the non-relevant ones get dropped as we move from one iteration to the next.

We do not directly use the non-relevant answers to guide the search away from non-relevant answers since our experience shows no significant improvement by doing so. Intuitively, the reason is that feature based representation does not fully capture the visual perception of a user. For example, two objects may have similar color histograms but maybe visually very different from the user's perspective. Therefore, directing a search away from non-relevant points in a feature space may actually cause the refined query to move away from the optimal representation in that feature space.

The distance of an object from a multipoint query is defined as the weighted combination of the individual distances from the representatives in the query, where the weight associated with a representative in a multipoint is proportional to the number of of relevant objects in its cluster. Figure 1 (b) shows the distance function for multipoint queries. The dashed lines are contours representing equidistant ranges for each of the representatives while the solid lines are contours representing equidistant ranges from the entire multipoint query.

Comparison: In each iteration of relevance feedback, QPM moves the query point and reweighs each dimension of the feature space to reduce the distances between the relevant points. This changes the distance function in a limited fashion. Although the weights are modified, the "function" for computing the distance is not changed. On the other hand, QEX does not change the distance function for each query point individually. But by adding relevant points and removing irrelevant ones, QEX implicitly changes the distance function from the multipoint query as a whole as shown in Figure 1 (b). Furthermore, while QEX captures local clusters among relevant points, QPM ignores these clusters and treats all relevant points equivalently. Our experiments show that these differences have a significant impact on the retrieval effectiveness. In terms of execution cost of the query, QEX may appear to be more expensive since it involves evaluation of queries consisting of multiple objects. In Section 4, we discuss evaluation techniques for such queries. With the developed techniques, QEX is more efficient compared to QPM in terms of execution

cost as well.

4 Query Evaluation Techniques

Multimedia Feature Indexing: We index each individual feature space using a multidimensional index structure, referred to as the *Feature Index* or *F-Index*. Similarity queries then correspond to k-nearest neighbor search (k-NN) on the F-Index. The goal of the F-index is to organize the feature vectors on disk so as to minimize the average number of disk accesses required to execute similarity queries on that feature. In addition to the I/O cost, the F-index also reduces the average CPU cost of a query since fewer node accesses implies fewer distance computations required. Even though the mechanism described below can be used in combination with any F-Index, we choose the *hybrid tree* [1] which is particularly suited as the F-Index.

The similarity query is executed using a k-NN algorithm defined as follows. The algorithm starts at the root node and accesses the index nodes in increasing order of their distances from the query. The distances are computed by "calling back" the distance function provided by the application. A priority queue is used to implement the ordered traversal over the index structure. At each step, the node with the minimum distance from the query is popped from the queue and its children are explored. The distance between the query and each child is computed and each child along with its distance from the query is pushed back in the queue. This algorithm guarantees that minimum number of nodes need to be visited. To use this algorithm, we need to define the distance function between the query and an index node. The following section defines the distance between the multipoint query and an index node.

Multiple Point Queries: The query expansion model requires support for similarity queries consisting of multiple query points. We refer to such queries as multipoint queries. A multipoint query is defined as follows.

Definition 1 (Multipoint Query) *A multipoint query $M = \langle n, \mathcal{P}, \mathcal{W}, \mathcal{D} \rangle$ consists of a set of points $\mathcal{P} = \{P_1, \ldots, P_n\}$, a set of weights $\mathcal{W} = \{w_1, \ldots, w_n\}$ and a distance function $\mathcal{D}$ that given two points, returns the distance between them. n is called the size of the multipoint query. We assume that the weights are normalized; i.e., $\sum_{i=1}^{n} w_i = 1$. The distance of M from a point P is defined as $D(M, P) = \sum_{i=1}^{n} w_i \mathcal{D}(P_i, P)$.*

Note that the multipoint query is a generalization of the single point query i.e. the latter is a special case of the former with size equal to 1.

Definition 2 (Similarity Multipoint Query) *A similarity multipoint query Q on the feature database DB is defined by the following association: $Q = \langle M, k \rangle$ where M is the multipoint query and k is the number of objects to be retrieved. Q returns the set $NN_Q^k \subseteq DB$ of k objects such that: $\forall o \in NN_Q^k$, $\forall o' \in DB - NN_Q^k$, $D(M, o) \leq D(M, o')$.*

The distance between a multipoint query and an index node of an F-index is defined as follows.

Definition 3 (MINDIST) *Given the bounding box $R_N = \langle L, H \rangle$ of a node N, where $L = \langle l_1, l_2, \ldots, l_m \rangle$ and $H = \langle h_1, h_2, \ldots, h_m \rangle$ are the two endpoints of the major diagonal of R, $l_i \leq h_i$ for $1 \leq i \leq m$. The nearest point $NP(P_i, N)$ in R_N (including the surface) to each point P_i in the multipoint query M is defined as follows.*

$$NP(P_i, N)[j] = \begin{cases} l_j & \text{if } P_i[j] < l_j \\ h_j & \text{if } P_i[j] > h_j \\ 0 & \text{otherwise} \end{cases}$$

where $P[j]$ denotes the value of point P along the j^{th} dimension of the feature space, $1 \leq j \leq m$. $MINDIST(M, N)$ is defined as:

$$MINDIST(M, N) = \sum_{i=1}^{n} w_i \mathcal{D}(P_i, NP(P_i, N))$$

For further details on the proof of correctness of multipoint query and the k-NN algorithm, we refer readers to [3].

Processing of Refined Queries: A naive approach of processing a refined query is to treat it like a fresh query and execute it from scratch. In each iteration of refinement, the refined query is passed to the F-index which invokes the k-NN algorithm and returns the desired number of answers to the application. This approach is wasteful since the same nodes may be accessed repeatedly by the k-NN algorithm iteration after iteration, leading to large number of unnecessary disk accesses and hence poor performance. Our goal is to eliminate any repeated disk access during the evaluation of the refined query. To achieve the goal, the MARS query processor use the old priority queue to construct a new priority queue based on the distance function of the refined query. Due to space limitations, we refer readers to [3] for further details of the approach.

5 Empirical Evaluation

For our experiments, we use the Corel Image Features dataset available online at `http://kdd.ics.uci.edu`. This collection contains features extracted from around 70,000 photo images. In our experiments, we use the color histogram and co-occurrence texture features. For further details of experimental setting, we refer readers to [3]. The experimental results shown below are averaged over a hundred queries.

Our experiments show that the retrieval performance in term of precision and recall improves from one iteration to the next for both QPM and QEX. Figure 2 shows the precision-recall graph for QEX. Figure 3 compare the

final recall of each approach after retrieving 100 points for each iteration. Both QPM and QEX start with a single example as the query. They produce the same retrieved set and hence the same recall for the initial query. Figure 3 shows that for QEX, the recall increases with each iteration and significantly outperforms QPM.

For QEX, we studied how the number of query points added (i.e. number of clusters produced) affects retrieval effectiveness of the approach. Our experiments show that a small number of well-chosen representatives provide a good approximation of the entire relevant set.

Figure 4 compares the evaluation technique proposed for refined queries to the naive approach. For both QPM and QEX, the proposed technique outperforms the naive approach by several orders of magnitude. In addition to retrieval effectiveness, QEX performs better than QPM in terms of execution cost as well.

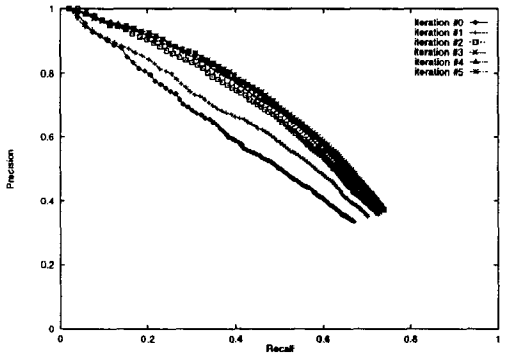


Figure 2: Precision Recall Graph for Query Expansion

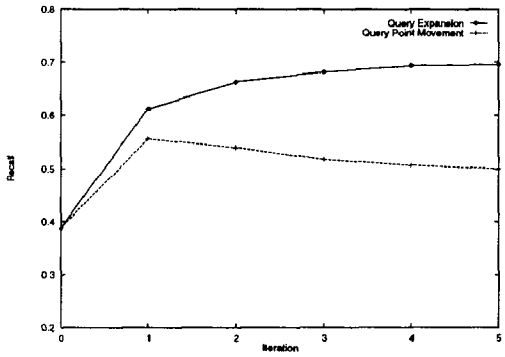


Figure 3: Comparison of retrieval effectiveness between the two approaches

6 Conclusions

In this paper, we studied a *query expansion* (QEX) approach to query refinement in multimedia databases. In QEX, at each iterations of the feedback, for each feature space, multiple well-chosen relevant points are added to the query representation. This is in contrast to the previously proposed approach of *query point movement* (QPM) in which a query point in each feature space

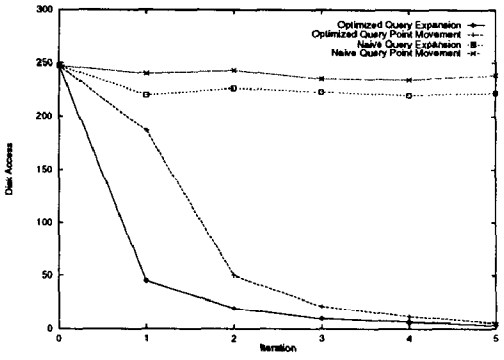


Figure 4: Execution cost of refined queries

is replaced by a weighted centroid of relevant points. We developed efficient techniques to evaluate *multipoint queries* over feature indices that result in the QEX approach. We also developed techniques to progressively evaluate refined queries efficiently by exploiting the work done in previous iterations for both the QEX and QPM approaches. Our experiments over a large image collection show that: (1) the query processing techniques developed very significantly improve the performance of both QEX and QPM, and (2) the QEX approach outperforms the QPM approach both in terms of retrieval effectiveness (precision and recall) as well as the cost of query evaluation (which is somewhat counterintuitive given that QEX results in multiple queries per feature space per iteration in contrast to QPM which results in only a single query per feature space per iteration). Due to space limitations, many details of the techniques developed are missing from this paper. We refer interested readers to [3] for details.

References

[1] K. Chakrabarti and S. Mehrotra. The hybrid tree: An index structure for high dimensional feature spaces. In *ICDE*, 1999.

[2] Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: Querying databases through multiple examples. In *VLDB*, 1998.

[3] K. Porkaew, K. Chakrabarti, and S. Mehrotra. Query refinement for multimedia similarity retrieval in mars. Technical Report TR-MARS-99-05, Univ. of California at Irvine, 1999.

[4] K. Porkaew, S. Mehrotra, and M. Ortega. Query reformulation for content based multimedia retrieval in MARS. In *IEEE ICMCS*, 1999.

[5] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Trans. on Circuits and Video Tech.*, September 1998.