Image Retrieval Based on User-Specified Features in Multi-Cluster Queries

Khanh Vu, Kien A. Hua, and Soontharee Koompairojn
Department of Computer Science, University of Central Florida
Orlando, Florida 32816-2362

Abstract

In a typical image retrieval system, all visual features of query images are used to determine image similarity. Thus, users are left to decide whether or not to include images that not only contain desirable features but also irrelevant ones. Fewer examples or a contaminated set of more could compromise the retrieval effectiveness of most similarity measures. In this paper, we extend our previous approach that allows users define queries by specifying relevant features present in image examples. The extended technique support queries decomposed in multiple clusters, each forming a subquery. Our experimental results have shown a remarkable improvement in retrieval performance.

1. Introduction

The need to effectively handle growing volumes of multimedia data has attracted much research attention addressing a variety of issues involving image indexing and retrieval. In previous content-based image retrieval (CBIR) systems, relevant images are determined based on their similarity to a single image. However, it has been shown that such a single-point query model lacks the flexibility to capture queries' intent [1]. Recent query models [2], [5], [6] allow queries to be defined with multiple example images. The extended model provides enormous flexibility in query formulation, able to overcome the limitations of traditional similarity measures [6].

In current CBIR systems, users are presented with a set of images (e.g., from the results of previous queries) to select relevant images that define the intent of their query. The system then extracts all pre-defined features such as color, texture and shape from these images that will be used for similarity computation. Thus, if an image is selected, all its features should be relevant in similarity determination. Otherwise, retrieval effectiveness is compromised. In reality, however, available images often contain some desirable features as well as irrelevant ones. In such cases, users face with the decision of either ignoring these images (thus fewer examples) or including partially irrelevant images (thereby risk contaminating the query set.) The end result of this dilemma is a frustrating experience for users and reduced system performance.

In our approach [9], similarity computation considers only relevant features specified by the users. As a result, they are able to define target results with a combination of features that may not be present in any single query image. Our study has shown that this capability improves effectiveness, reduces querying time and converges faster than other relevance feedback techniques. We extend this approach to handle multi-cluster queries. Recent research shows that decomposing complex queries into multiple clusters of (sub)queries yields higher precision than the traditional one-cluster methods [3].

Section 2 reviews our previous work. We describe our approach in detail in Section 3. Section 4 discusses the results of our performance study. Concluding remarks are given in Section 5.

2. Multipoint Query Retrieval

In this section, we briefly describe how to compute the center point for multipoint queries some of which might contain unwanted visual features, and a proposed similarity measure.

2.1. Computing the Query Point

In many existing approaches for multipoint queries, the center of retrieval sets is the centroid. When not all features are relevant, the ideal query may not be located at the centroid of the query points. We select the query point for a set of given query points as follows. Let $Q = \{Q_1, Q_2, \ldots, Q_k\}$ be a set of k distinct query images in any of which, there is at least one feature is relevant. That is, suppose Q_i is represented by a set of n predefined features q_{i1}, q_{i2}, …, and q_{in}, and $r_{ij} = 1$ if

1-4244-0367-7/06/$20.00 ©2006 IEEE

feature q_{ij} of Q_i is relevant and $r_{ij} = 0$ otherwise. Then $\sum_{i=1}^{n} r_{ij} \geq 1$. The query point $C = (c_1, c_2, \ldots, c_n)$ is defined as:

$$c_i = \frac{\sum_{j=1}^{n} r_{ij} q_{ij}}{\sum_{j=1}^{n} r_{ij}}. \qquad (1)$$

When $\sum_{j=1}^{n} r_{ij} = n$ for $1 \leq i \leq k$, C is the traditional centroid of Q. Observe that users select relevant image features in groups such as color, texture and shape groups, thus r_{ij} are specified in groups as well. Figure 1(a) shows the query point for a query with 3 image points: all features f_1 and f_2 in Q_1 and Q_2 are relevant, while only feature f_2 in Q_3 is relevant. Note that the traditional centroid point (the cross) sits closer to partially relevant image Q_3, which causes retrieval sets to skew toward less relevant images.

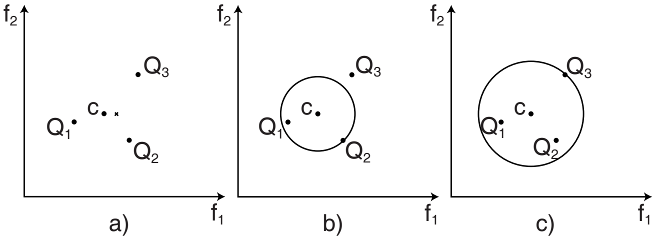

Figure 1. A 3-point query and retrieval sets based on $\mathcal{L}_2$ distance.

2.2. The Similarity Measure

With C defined, the dissimilarity of query Q and an image O is quantified by the distance between them. The following formula is based on the weighted $\mathcal{L}_p$ distance:

$$\mathcal{D}(Q, O) = \mathcal{L}_p(C, O) = \left(\sum_{i=1}^{n} w_i (c_i - o_i)^p \right)^{1/p}$$

where w_i accounts for the significance of feature i of C. Similarly to r_{ij}, w_i are specified in groups. When $w_i = 1/n$, for $1 \leq i \leq n$, all visual image features are equally important in determining their similarity.

2.3. Retrieval of query points

It is expected that query images containing all relevant features should be retrieved. However, the inclusion of partially relevant query images is less clear. We consider two strategies for these images: conservative and liberal. These options are based on whether or not to extend the search radius up to partially relevant query images. Interested readers are referred to [9] for more detail.

3. Multi-Cluster Query Retrieval

In the previous section, we dealt with multipoint queries that can be represented by a single query point. When query examples are too diverse, it is more effective to divide them into groups and execute them as subqueries. The results are merged into one final set. In this section, we extend the idea to this general case.

3.1. Normalization of Query Points

Existing techniques such as [5] cluster query points considering all features. In order to use these techniques, we need to neutralize the effects of unwanted features of partially relevant points. One way to accomplish this is to modify the dimensions so that they are all relevant and the centroid of the new points is identical to the one computed using Eq. 1. Let $C = (c_1, c_2, \ldots, c_n)$ be the query point defined in Eq. 1. Point Q_j is mapped to Q_j^*, $1 \leq j \leq k$:

$$q_{ij}^* = \begin{cases} q_{ij} & \text{if } q_{ij} \text{ is relevant} \\ c_i & \text{otherwise} \end{cases} \qquad (2)$$

Thus, when Q_j contains all relevant features, $Q_j \equiv Q_j^*$ and only partially relevant query points get mapped to new points. Figure 2 illustrates the mapping of Q_3, whose f_1 is not relevant, to Q_3^*. Note that the centroid of $\{Q_1, Q_2, Q_3^*\}$ is identical to point C computed using Eq. 1. We prove that this holds true for the general case.

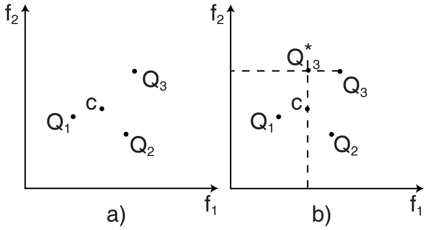

Figure 2. Mapping a partially relevant point.

Lemma 1 *The mapping using Eq. 2 of query set Q produces a centroid that is identical to point C of Q computed using Eq. 1.*

Proof We prove that the coordinates of the centroid of the mapped points are the same as C's. Obviously this is true if all features of the points are relevant. It remains to be proved that if Q_j contains irrelevant features q_{ij}, its mapped point will not change the coordinate c_j of C. This is guaranteed according to Eq. 2.

We will continue to use the notation Q_j, instead of Q_j^*, to denote its normalized point.

3.2. Clustering Query Points

Normalized query points can now be clustered using a variety of methods. We select the hierarchical clustering algorithm which groups data points into hyperspherical regions in $O(k^2)$ time [4]. The clusters are then optionally refined into hyper-ellipsoids that are described by their characteristics such as the means (determining the location) and covariance matrix (describing the shape and orientation) [5]. Algorithm 1 describes the basic steps of the hierarchical clustering approach. It returns the centroids and the number of query points that get mapped to the respective clusters.

Algorithm 1 Hierarchical Clustering

```
begin
Initialize the clusters, each containing one query image. $k$ is the initial number of clusters.
while $k >$ desired number of clusters do
    $k = k - 1$.
    for each pair of clusters do
        Calculate the centroid of each cluster.
        Compute the distance between the centroids.
    end for
    Sort the distances.
    Merge the pair of clusters whose distance is the shortest.
end while
Compute and return the number of query points and the centroid of each cluster.
end
```

Algorithm 1 is executed when the distance between query points exceeds a predetermined threshold. The desired number of query clusters in our experiments is typically from 3 to 5.

3.3. Similarity Measure for Disjunctive Sets

To rank retrieved images in multi-cluster queries, we need a generalized version of the similarity measure in single-centroid query retrieval. Let C be the set of the centroids of the clusters produced by Algorithm 1 and m_j the number of query points in cluster j. Equation 3 quantifies the similarity between image P and query set Q. Observe that retrieved images in larger clusters are ranked higher assuming an equal search distance from the centroids.

$$\mathcal{L}_p(P, Q) = \frac{1}{\sum_{C_j \in C} \frac{m_j}{\mathcal{L}_p(P, C_j)}} \qquad (3)$$

Figure 3 shows retrieval results using Eq. 3 of two subqueries. They belong to disconnected regions clustering around centroids C_1 and C_2 of two clusters. Had a single centroid been used for the query set, irrelevant points in-between could also have been retrieved and possibly ranked higher.

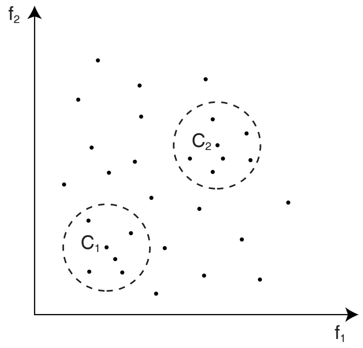

Figure 3. Retrieval of subqueries.

3.4. Multi-Cluster Search

Algorithm 2 summarizes the search procedure for a set of query images containing desirable and undesirable features.

Algorithm 2 Multi-Cluster Search

```
begin
Compute $C$ using Eq. 1.
Compute the distance between each pair of the query images.
if the largest distance is smaller than a predetermined threshold then
    Retrieve and rank relevant images from the database based on their distance to $C$.
else
    Set $Q^* \leftarrow \emptyset$.
    for all $Q_j \in Q$ do
        Compute $Q_j^*$ using Eq. 2.
        $Q^* \leftarrow Q^* \cup Q_j^*$.
    end for
    Cluster $Q^*$ according to Algorithm 1
    Retrieve and rank relevant images from the database using to Eq. 3.
end if
end
```

4. Performance Study

We have conducted a variety of experiments to evaluate the performance of our approach (SFME) using $\mathcal{L}_2$ distance. Our dataset consists of about 20,197 Corel images from various categories. For each image, we extracted 3 groups of visual features as suggested in recent methods, including color features (9 components) [8], texture (10 components) [7], and edge-based structure (18 components) [10]. We compared our technique with ImageGrouper (IG), Principal Component Analysis (PCA) and Adaptive distance computation (ADC) techniques. Thus, when the color group is selected as relevant, all its 9 components are utilized, and so on for the other two groups. We implemented these techniques on a prototype based on IG. Our prototype allows users to specify any combination of relevant groups for each by checking the three boxes below the images. We can set a specific number of decomposed clusters or let it be determined automatically by the system. We used the manual setting to evaluate the effects of query decomposition. In our experiments, we decomposed each set of query examples into one, three, and five clusters.

Tables 1, 2 and 3 summarize the precision performance of the compared techniques for various database sizes. Querying with 1-cluster is our previous approach. As seen, SFME outperforms the other techniques because it is able to ignore the effects of irrelevant features in the examples.

When query examples with diverse characteristics are decomposed into multiple (sub)queries, higher precision can be achieved than executing it as a single query. The results show that our extended approach performs very well.

This highlights the advantages of retrieval based on relevant features in images, not only in traditional one-cluster techniques but also in more recent multi-cluster approaches.

	1 cluster	3 clusters	5 cluster
SFMC	0.83	0.85	0.87
IG	0.66	0.70	0.72
ADC	0.55	0.56	0.58
PCA	0.66	0.70	0.71

Table 1. Retrieval precision, dataset = 5K

	1 cluster	3 clusters	5 cluster
SFMC	0.67	0.70	0.73
IG	0.55	0.58	0.60
ADC	0.52	0.56	0.59
PCA	0.64	0.68	0.71

Table 2. Retrieval precision, dataset = 10K

	1 cluster	3 clusters	5 cluster
SFMC	0.59	0.62	0.65
IG	0.50	0.54	0.57
ADC	0.34	0.41	0.44
PCA	0.39	0.45	0.49

Table 3. Retrieval precision, dataset = 20K

References

[1] T. E. Bjoerge and E. Y. Chang. Why one example is not enough for an image query. In *In the Proc. IEEE International Conference on Multimedia, Taipei*, June 2004.

[2] K. Chakrabarti, M. O.-B., S. Mehrotra, and K. Porkaew. Evaluating refined queries in top-k retrieval systems. *IEEE Transactions on Knowledge and Data Engineering*, 16:256–270, 2004.

[3] K. A. Hua, N. Yu, and D. Liu. Query decomposition: A multiple neighborhood approach to relevance feedback processing in content-based image retrieval. In *Proceedings of the 22nd International Conference of Data Engineering (ICDE2006)*, 2006.

[4] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 2:241–254, 1967.

[5] D.-H. Kim and C.-W. Chung. Qcluster: relevance feedback using adaptive clustering for content-based image retrieval. In *Proceedings of the ACM SIGMOD Conference*, pages 599–610, 2003.

[6] M. Ortega-Binderberger and S. Mehrotra. Relevance feedback techniques in the mars image retrieval system. *Multimedia Systems*, 9:535 – 547, June 2004.

[7] J. R. Smith and S.-F. Chang. Transform features for texture classification and discrimination in large image databases. In *Proceedings of the International Conference on Image Processing*, pages 407–411, 1994.

[8] M. A. Stricker and M. Orengo. Similarity of color images. In *Proceedings of Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392, 1995.

[9] K. Vu, K. A. Hua, and S. Koompairojn. Image retrieval based on user-specified features in queries with multiple examples. In *The Twelfth International Multimedia Modeling Conference (MMM2006)*, pages 430–433, 2006.

[10] X. S. Zhou and T. S. Huang. Edge-based structural features for content-based image retrieval. *Pattern Recognition Letters*, 22(5):457–468, 2001.