Available online at www.sciencedirect.com
SCIENCE DIRECT®
The Journal of Systems and Software 78 (2005) 9–23
The Journal of Systems and Software
www.elsevier.com/locate/jss

Relevance feedback using adaptive clustering for image similarity retrieval

Deok-Hwan Kim [a,*], Chin-Wan Chung [b], Kobus Barnard [c]

[a] Department of Mobile Internet, Dongyang Technical College, 62-160 Kochuk-dong, Kuro-gu, Seoul 152-714, South Korea
[b] Division of Computer Science, Korea Advanced Institute of Science and Technology, 373-1, Kusong-dong, Yusong-gu, Taejon 305-701, South Korea
[c] Computer Science Department, University of Arizona, Gould-Simpson 730, Tucson, AZ 85721, USA

Received 2 September 2004; received in revised form 8 February 2005; accepted 8 February 2005
Available online 9 March 2005

Abstract

Research has been devoted in recent years to relevance feedback as an effective solution to improve performance of image similarity search. However, few methods using the relevance feedback are currently available to perform relatively complex queries on large image databases. In the case of complex image queries, images with relevant concepts are often scattered across several visual regions in the feature space. This leads to adapting multiple regions to represent a query in the feature space. Therefore, it is necessary to handle disjunctive queries in the feature space.

In this paper, we propose a new adaptive classification and cluster-merging method to find multiple regions and their arbitrary shapes of a complex image query. Our method achieves the same high retrieval quality regardless of the shapes of query regions since the measures used in our method are invariant under linear transformations. Extensive experiments show that the result of our method converges to the user's true information need fast, and the retrieval quality of our method is about 22% in recall and 20% in precision better than that of the query expansion approach, and about 35% in recall and about 31% in precision better than that of the query point movement approach, in MARS.

Keywords: Relevance feedback; Image database; Classification; Cluster-merging; Dimension reduction; Content-based image retrieval

1. Introduction

The relevance feedback based approach to content-based image retrieval (CBIR) has been an active research areas in the past few years. A good survey can be found in (Ishikawa et al., 1998; Rui et al., 1998; Rui et al., 1999). Most existing CBIR systems represent images as feature vectors using visual features, such as color, texture and shape. That is, the closer two vectors are, the more similar the corresponding images are. They search images via a query-by-example (QBE) interface. When the systems present a set of images considered to be similar to a given query, the user can pick up the ones most relevant to the given query, and the system refines the query using them, which allows the relevant images be the ones picked up by the user. Relevance feedback based CBIR techniques do not require a user to provide accurate initial queries, but rather estimate the user's ideal query by using relevant images feedback by the user.

Current approaches to CBIR assume that relevant images are physically near the query image in some feature space regardless of visual features. However, the similarity between images perceived by humans does not necessarily correlate with the distance between them

* Corresponding author. Tel.: +8222 6101914; fax: +8222 6101859.
E-mail addresses: dhkim@dongyang.ac.kr (D.-H. Kim), chungcw@islab.kaist.ac.kr (C.-W. Chung), kobus@cs.arizona.edu (K. Barnard).

0164-1212/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2005.02.005

in the feature space. That is, semantically relevant images might be spread out in the feature space and be scattered in several clusters rather than one. In this case, traditional relevant feedback approaches (Flickner et al., 1995; Ishikawa et al., 1998; Rui et al., 1997; Rocchio, 1971) do not work well when shifting the query center by linear combination of the relevant images.

Implementing the relevance feedback concerns the computation of a new query point (or points) in a feature space and the change of a distance function. As shown in Fig. 1(a), early studies (Ishikawa et al., 1998; Rui et al., 1997) represent a new query as a single point and change the weights of feature components to find an optimal query point and an optimal distance function. In this case, a single point is computed using the weighted average of all relevant images in the feature space. The contours represent equi-similarity lines. Meanwhile, a recent study (Porkaew and Chakrabarti, 1999) represents a new query as multiple points to determine the shape of the contour as shown in Fig. 1(b). This approach uses a clustering method (Charikar et al., 1997) to compute new query points using query results (relevant images) based on the user's relevance judgement. It is assumed that the relevant images are mapped to points close together according to the similarity measure. A single large contour is constructed to cover all query points and the system finds images similar to them. However, if the feature space and the distance function for the user's perception are quite different from those for the system, the relevant images are mapped to disjoint regions of arbitrary shapes in the feature space. That is, the relevant images may be ranked below other retrieved images for the given query. In order to converge rapidly to the user's information need at the higher semantic level, the system should find the images similar to any of the query points as in Fig. 1(c). A query that retrieves the images similar to any of the query points is called a disjunctive query. Especially, a complex image query is represented as disjoint multiple regions since semantically related images can be scattered in several visual regions rather than one.

In this paper, we propose a new adaptive classification and cluster-merging method (Qcluster: query clustering) to find multiple regions and their arbitrary shapes of contours for a given complex image query. Also we propose an approach to the relevance feedback using multiple query points to support disjunctive queries.

Fig. 2 shows the proposed relevance feedback mechanism. At the first stage, an example image submitted by the user is parsed to generate an initial query $Q = (q, d, k)$, where q is a query point in the feature space, k is the number of images in the query result returned by the system, and d is the distance function. The query point q is compared with images in the database using the distance function d. According to d, the result set consisting of k images close to q, $\text{Result}(Q) = \{p_1, \ldots, p_k\}$, is returned to the user.

At the next stage, the user evaluates the relevance of images in $\text{Result}(Q)$ by assigning a relevance score to each of them. Based on those scores, the relevant set, $\text{Relevant}(Q) = \{p'_1, \ldots, p'_m\}$, is obtained. In this paper, we present a new adaptive clustering method consisting of two processes: the classifying process and the cluster-merging process. The proposed classifying process places each element of the relevant set, $\text{Relevant}(Q)$, in one of the current clusters or a new cluster. Then, the proposed cluster-merging process finds the appropriate number of clusters by merging certain clusters to reduce the number of query points in the next iteration. Finally, representatives of clusters generated from relevant images in the classified set make up the set of new query points. A new query, $Q' = (q', d', k)$ with a set of new query points q' and a new distance function d', is computed and then used as an input for the second round.

After some iterations, the loop ends up with the final result set close to $\text{Result}(Q_{\text{opt}})$, where $Q_{\text{opt}} = (q_{\text{opt}}, d_{\text{opt}}, k)$ is the optimal query.

Our approach to the relevance feedback allows multiple objects to be a query. We refer to them as a multipoint query. When the user marks several points as relevant, we cluster sets of relevant points and choose the centroids of the clusters as their representatives. Then, we construct the multipoint query using a small number of good representative points. At the classifying process, Bayesian classification function (Fisher, 1938) is used. Statistics such as mean and covariance of each cluster, which were computed from the previous itera-

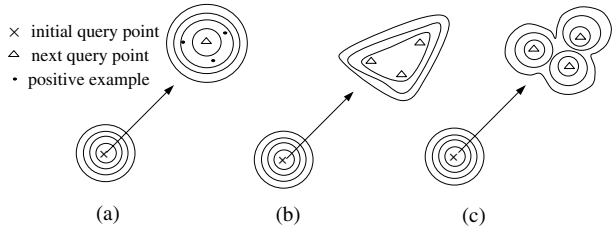

Fig. 1. Query shape. (a) Query point movement (b) Convex shape (multipoint) and (c) Concave shape (multipoint).

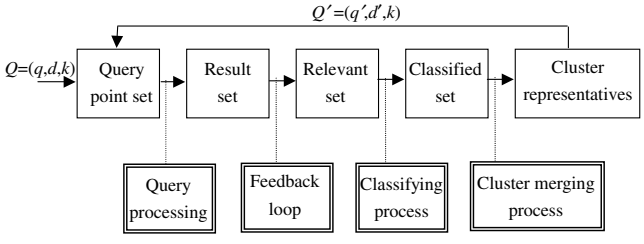

Fig. 2. Overall structure of the proposed method.

tion, are used as the prior information. At the cluster-merging process, Hotelling's T^2 (Johnson and Wichern, 1998) is used to merge any pair of clusters in arbitrary shapes and find the number of the appropriate clusters for a given query.

The contributions of this paper are as follows:

- The adaptive clustering generates contours consisting of multiple hyper-ellipsoids, and therefore our retrieval method can handle disjunctive queries.
- Our method constructs clusters and changes them without performing complete re-clustering. Its computing time is short since the same statistical measures are used at both the classification stage and the cluster-merging stage.
- The measures used in our method are invariant under linear transformations. Therefore, the retrieval quality is the same regardless of the shape of query clusters.
- Our experimental results show that the proposed method achieves about 22% improvement of recall and 20% improvement of precision against the query expansion approach (Porkaew and Chakrabarti, 1999), and about 35% improvement of recall and about 31% improvement of precision against the query point movement approach (Rui et al., 1997), in MARS.

The remainder of the paper is organized as follows: In Section 2, we discuss related work. In Section 3, we present some interesting motivating examples, the similarity measure, and the overall algorithm of the multipoint relevance feedback. We describe the classification and cluster-merging processes in Section 4. Section 5 contains the result of our experiments on a large set of 30,000 heterogeneous images and synthetic data. Finally, Section 6 summarizes our work.

2. Related work

Earlier approaches (Flickner et al., 1995; Smith and Chang, 1996) to the content-based image retrieval do not adapt the query and retrieval model based on the user's perception of the visual similarity. To overcome this problem, a number of relevance feedback techniques (Ashwin et al., 2002; Bartolini et al., 2001; Benitez et al., 1998; Brunelli and Mich, 2000; Ishikawa et al., 1998; Porkaew and Chakrabarti, 1999; Rui et al., 1997; Wu et al., 2000; Wang et al., 2001; Zhou and Huang, 2001) have been proposed. They try to establish the link between semantic concepts and low-level image features and model the user's subjective perception from the user's feedback. There are two components to learn for the relevance feedback: a distance function and a new query point. The distance function is changed by learning weights of feature components, and the new query point is obtained by learning the ideal point that the user looks for.

The query-point movement has been applied to the image retrieval systems such as MARS (Rui et al., 1997) and MindReader (Ishikawa et al., 1998). These systems represent the query as a single point in the feature space and try to move this point toward "good" matches, as well as to move it away from "bad" result points. This idea originated from the Rochio's formula (Rocchio, 1971), which has been successfully used in document retrieval. In this approach, the weighting technique assigns a weight to each dimension of the query point. It associates larger weights with more important dimensions and smaller weights with less important ones. MARS uses a weighted Euclidean distance, which handles ellipsoids whose major axis is aligned with the coordinate axis. On the other hand, MindReader uses a generalized Euclidean distance, which permits the rotation of the axes so that it works well for arbitrarily oriented ellipsoids.

Recently, other query refinement methods using the multipoint relevance feedback were introduced. The query expansion approach (Porkaew and Chakrabarti, 1999) of MARS constructs local clusters for relevant points. In this approach, all local clusters are merged to form a single large contour that covers all query points. On the other hand, the query-point movement approach (Rui et al., 1997; Ishikawa et al., 1998) ignores these clusters and treats all relevant points equivalently. These two approaches can generate a single hyper-ellipsoid or convex shapes using local clusters in some feature space to cover all query points for simple queries. However, both approaches fail to identify appropriate regions for complex queries. Wu et al. (2000) presented FALCON, the aggregate dissimilarity model, to facilitate learning disjunctive and concave query points in the vector space as well as in arbitrary metric spaces. However, the proposed aggregate dissimilarity function depends on ad hoc heuristics and this model assumes that all relevant points are query points.

3. Multipoint relevance feedback approach

This section proposes the overall mechanism of our approach to the multipoint relevance feedback. Table 1 shows some notations to be used.

3.1. Motivating examples

Example 1. The user wants to select bird images via query-by-example in the image data set of 30,000 color images. A pairwise distance metric relying primarily on color is used to compare images. As shown in Fig. 3, the set of retrieved relevant images includes bird images

Table 1
Symbols and their definitions

Symbol	Definition
p	Dimension of feature vector
$x_{ij} = [x_{ij1}, \ldots, x_{ijp}]'$	Feature vector of jth image of ith cluster
$C_1, \ldots, C_g$	g clusters
$\bar{x}_i = [\bar{x}_{i1}, \ldots, \bar{x}_{ip}]'$	Weighted centroid of ith cluster
$Q = \{\bar{x}_1, \ldots, \bar{x}_g\}$	Query set of g multiple points
$d^2()$	Generalized Euclidean distance function
q_{opt}	Ideal query point
n_i	The number of images for ith cluster
m_i	The sum of relevance score values of i cluster
w_i	Normalized weight of i cluster
α	Significance level
S^{-1}_{pooled}	Pooled inverse covariance matrix
S_i^{-1}	Inverse covariance matrix of i'th cluster
v_{ij}	Relevance score value for jth image of ith cluster
$\hat{d}_i()$	Classifier function for cluster i
$T^2()$	Cluster-merging measure
$d^2_{\text{aggregate}}()$	General aggregate distance function
$d^2_{\text{disjunctive}}()$	Disjunctive aggregate distance function
$p_1, \ldots, p_k$	Retrieved images
$p'_1, \ldots, p'_m$	Relevant images
r	Effective radius
c^2	Critical distance value

with a light-green background and ones with a dark-blue background. However, these may not be projected to points close together in the feature space. Instead, the points form two distinct clusters.

Finding similar images in this space is related to clustering. If the difference between the user's perception and the feature representation in the system gets large, there comes a necessity of expressing a query by several points in the feature space. MARS uses multiple point queries and every query point is supposed to be merged. All relevant images are merged to several clusters and a single large contour is made to cover all representatives of these clusters.

Example 2. Given the top-leftmost image as a query in Fig. 3, Fig. 4 shows the 3 dimensional plot of feature vectors of 10 retrieved relevant images. It shows that five

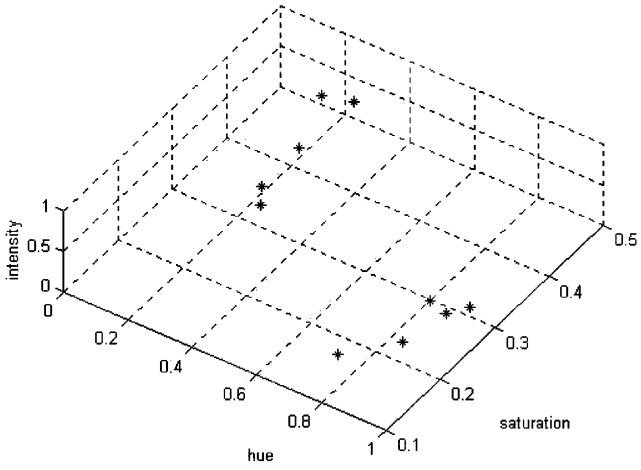

Fig. 4. 3 dimensional plot of 10 points.

images are similar, but the other five images are quite different, MARS makes a single contour for two clusters. However, it is better to make separate contours for two different clusters of data. Our method can determine the shapes of two local clusters.

For query expansion, Porkaew and Chakrabarti (1999) assumed that query images given by a user should be similar. Their method makes several clusters to include all relevant images and builds a large contour to cover them. So the single large contour of the query clusters is used as the basis for the search. However, the query clusters might be very distant from each other. So, the search might not yield fruitful results. For a successful search, the contours must be separated instead of being combined, as our method does.

A complex image query must be expressed as multiple query points so that multiple representatives of clusters are used. The basic method of clustering image feature vectors is as follows: Initially, assign n input points (feature vectors of images) to n distinct clusters. Among all clusters, pick up the two clusters with the smallest distance between them. Merge them to form a new cluster. Repeat these two steps until there are no clusters left to

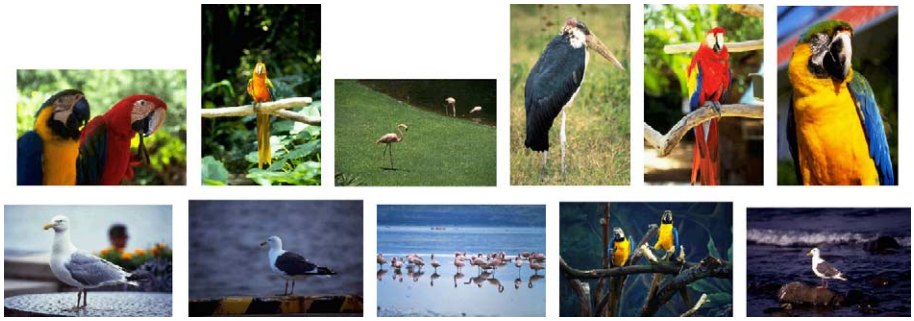

Fig. 3. Bird images.

be merged. The basic idea of our approach is to use an adaptive classification and cluster merging method that find clusters and their arbitrary shapes of multiple regions of a query. Our goal is as follows:

- Given: user-selected p-dimensional points from the result of a k-nearest neighbor query and their relevance scores.
- Goal: find a set of centroids $\bar{x}_1, \ldots, \bar{x}_g$ of clusters and their weights $w_1, \ldots, w_g$.

We use multiple points, which are the centroids of clusters $C_1, \ldots, C_g$ to guess the best query point q_{opt}, their covariance matrices, and weights to learn the hidden distance function.

3.2. Similarity measure

When a user marks several images as relevant ones at each iteration of the relevance feedback, we cluster a set of relevant points and choose the centroid of the cluster as its representative. For simplicity, let us use $x_{ij} = [x_{ij1}, \ldots, x_{ijp}]'$ to denote the feature vector of the jth image in ith cluster in $\Re^p$. Let $\bar{x}_i = [\bar{x}_{i1}, \ldots, \bar{x}_{ip}]'$ in $\Re^p$ be a centroid vector of the ith cluster. The distance d^2 between the two points $\bar{x}_i$ and x_{ij} is defined by:

$$d^2(\bar{x}_i, x_{ij}) = (\bar{x}_i - x_{ij})' S^{-1} (\bar{x}_i - x_{ij}) \tag{1}$$

$$= \sum_{l=1}^{p} \sum_{m=1}^{p} (\bar{x}_{il} - x_{ijl}) S^{-1} (\bar{x}_{im} - x_{ijm}), \tag{2}$$

where S^{-1} is a $p \times p$ distance matrix and denotes weights of dimensions and correlations in the feature space.

A generalized Euclidean distance $d_i^2(x)$ between a new point x and the centroid of the ith cluster C_i, $\bar{x}_i$, is computed as follows:

$$d_i^2(x) = (x - \bar{x}_i)' S_{\text{pooled}}^{-1} (x - \bar{x}_i), \tag{3}$$

where

$$S_{\text{pooled}} = \frac{1}{m_1 + m_2 + \cdots + m_g - g} [(m_1 - 1)S_1 + (m_2 - 1)S_2 + \cdots + (m_g - 1)S_g],$$

m_i is the weight (sum of relevance score values and value is more than or equal to 1) of the ith cluster C_i, and S_i is the covariance matrix of C_i, for $i = 1, \ldots, g$.

This generalized Euclidean distance allows not only different weight of each dimension, but also correlation. It can express a user's high-level concept better than weighted Euclidean or ordinary Euclidean distances, since it is an ellipsoid whose major axes are not necessarily aligned with the coordinate axes. And, an ellipsoid can express a user's hidden distance function better than a circle. MindReader (Ishikawa et al., 1998) proved that this method is theoretically solid to handle similarity queries.

Conventionally, a similarity query is represented as a single point, while we insist that a complex image query be represented as multiple points. We compute multiple representatives or a single representative using the proposed adaptive classification and cluster-merging method. A general aggregate distance function between a point x and a set of multiple query points $Q = \{\bar{x}_i, \ldots, \bar{x}_g\}$ is defined by (Salton et al., 1983; Wu et al., 2000):

$$d^{\beta}_{\text{aggregate}}(Q, x) = \frac{1}{g} \sum_{i=1}^{g} d^{\beta}(\bar{x}_i, x). \tag{4}$$

The negative value of β mimic a fuzzy OR function since the smallest distance will have the largest impact on the aggregate distance function. We use $\beta = -2$. Then, we apply the following aggregate distance function to those representatives to find images similar to one of the representatives in the query point set.

$$d^2_{\text{disjunctive}}(Q, x) = \frac{\sum_{i=1}^{g} w_i}{\sum_{i=1}^{g} w_i / [(\bar{x}_i - x)' S^{-1}_{\text{pooled}} (\bar{x}_i - x)]}, \tag{5}$$

where Q is a set of multiple cluster representatives $\{\bar{x}_1, \ldots, \bar{x}_g\}$, x is the feature vector of a target image, and w_i is a normalized weight of the ith cluster.

3.3. A general algorithm

We propose a novel relevance feedback approach for multipoint queries using the adaptive classification and cluster-merging method. The algorithm of our relevance feedback for multipoint queries is as follows:

Algorithm 1. k-nearest neighbor search
input: a query example
output: k retrieved images
begin
Step1: Initialization
  (1) Set $Q = (q, d, k)$ using initial query image.
Step2: Retrieval of images
  (1) Retrieve k images Result$(Q) = \{p_1, \ldots, p_k\}$ such that $d^2_{\text{disjunctive}}(Q, p_1)$ is the lowest, $d^2_{\text{disjunctive}}(Q, p_2)$ is the next lowest, and so on.
Step3: User Interaction
  For each p_j in Result(Q),
    (1) recommend p_j to the user.
    (2) If p_j is marked as relevant, then add p_j to Relevant(Q).
  EndFor
Step4: Hierarchical Clustering or adaptive classification
  If initial iteration
    (1) perform hierarchical clustering using images in Relevant(Q)
  Else
    For each p'_j in Relevant(Q)

(2) Determine an appropriate cluster using Bayesian classification function.
(3) If p'_j is located within the determined cluster boundary, place it in the determined cluster.
EndFor
EndIf
Step5: Cluster-merging
(1) For each cluster C_i calculate a centroid, its covariance, and its weight.
(2) find the appropriate number of clusters $C_1, \ldots, C_g$ by using Hotelling's T^2.
Step6: Query refinement and distance function update
(1) Refine $Q' = (q', d', k)$ with a set of new query points q' and a new distance function d'
(2) Goto Step2 for the next iteration
end

4. Adaptive classification and merging clusters

The adaptive classification and merging clusters are the cores of our approach. They are used to accelerate query processing by considering only a small number of representatives of the clusters, rather than the entire set of relevant images. When all relevant images are included in a single cluster, it is the same as MindReader's. At each stage the clusters are modified according to the result of a query and a user's relevance feedback. Therefore, it is necessary to construct new clusters without complete re-clustering.

The proposed method is composed of two stages: the classification stage and cluster-merging stage. At the first stage, new points are classified into current clusters or new clusters using their statistical information. At the second stage, the number of current clusters is reduced. Its advantages are as follows:

- Two stages share statistics such as a distance matrix S^{-1}_{pooled} and the centroids of current clusters.
- It is easy to compute multiple query points since the method does not re-cluster completely at each iteration.
- The method can approximate any query shape to an arbitrarily oriented ellipsoid since the distance function is a quadratic form.

4.1. Initial hierarchical clustering

Initial clusters of the training data form the basis of the hierarchy. Among numerous methods, we use the hierarchical clustering algorithm that groups data into hyperspherical regions. Once initial clusters are obtained, we calculate a mean vector $\bar{x}$, a weighted covariance matrix S, and an effective radius r. The mean vector determines the location of the hyperellipsoid, while the covariance matrix characterizes its shape and orientation. The weight of each cluster compared with the others is determined by the sum of relevance score values of points in each cluster. The effective radius is a critical value to decide whether a new point x lies inside the given ellipsoid.

Lemma 1. *If x lies inside the ellipsoid, the following property is satisfied* (Bauer et al., 1999):

$$(x - \bar{x})' S^{-1} (x - \bar{x}) < r \tag{6}$$

Let us assume that the data follows a Gaussian distribution and takes α as a significance level. For the given significance level α, $100(1-\alpha)\%$ (typically 95–99%) of the data will fall inside the ellipsoid and the generalized Euclidean distance follows a χ^2_p distribution with p degrees of freedom. Then the effective radius r is $\chi^2_p(\alpha)$. As α decreases, a given effective radius increases. Any point outside of the ellipsoid is identified as an outlier and forms a new cluster.

4.2. Classification stage

Let $C_1, \ldots, C_g$ be g clusters. The classification algorithm places a new point in one of the g clusters or in a new cluster. The classifier is based on the Bayesian classification function (Duda et al., 2001) and it uses means, covariance matrices, and weights of clusters at the cluster-merging stage of the previous iteration as prior information. The classification rule is as follows:

$$\text{Allocate } x \text{ to } C_k \text{ if } w_k f_k(x) > w_i f_i(x) \quad \text{for all } i \neq k \tag{7}$$

where f_i is the probability density function of C_i.

At the feedback loop stage, the user specifies a score value v for each image x. Later, after the cluster-merging stage of the current iteration, if x has become the kth point of C_i, then v becomes v_{ik}. m_i is the weight of C_i, i.e., $m_i = \sum_{k=1}^{n_i} v_{ik}$. Then w_i is the normalized weight of the ith cluster, that is, $w_i = m_i / \sum_{k=1}^{g} m_k$. The classification rule in Eq. (7) is identical to one that maximizes the "posterior" probability $P(C_k|x) = P(x \text{ comes from } C_k \text{ given that } x \text{ was observed})$, where

$$P(C_k \mid x) = \frac{w_k f_k(x)}{\sum_{i=1}^{g} w_i f_i(x)} = \frac{(\text{prior}_k) \times (\text{likelihood}_k)}{\sum[(\text{prior}_i) \times (\text{likelihood}_i)]} \quad \text{for } i = 1, \ldots, g. \tag{8}$$

An important special case occurs when $f_i(x)$ is a multivariate normal density function with centroid vector $\bar{x}_i$ and covariance matrix S_i of cluster C_i for $i = 1, \ldots, g$. Then Eq. (7) becomes:

Allocate x to C_k if $\ln(w_k f_k(x)) \geqslant \ln(w_i f_i(x))$ for all $i \neq k$ where $\ln(w_i f_i(x)) = \ln(w_i) - \frac{p}{2}\ln(2\pi) - \frac{1}{2}\ln|S_i| - \frac{1}{2}(x-\bar{x}_i)'S_{\text{pooled}}^{-1}(x-\bar{x}_i)$.

The constant and common terms are ignored (see reference Duda et al. (2001) in finding the classification function for each cluster). Then the estimate of the classification function for C_i for $i = 1, \ldots, g$ is found to be:

$$\hat{d}_i(x) = -\frac{1}{2}(x-\bar{x}_i)'S_{\text{pooled}}^{-1}(x-\bar{x}_i) + \ln(w_i). \tag{9}$$

The basic idea of this classification algorithm is that a new point x should be classified to the nearest cluster. The details are as follows: For a given new point x, $\hat{d}_i(x)$ is calculated and x is assigned to kth cluster where $\hat{d}_k(x)$ is maximal. If the distance value is less than the effective radius of C_k, the point is placed to that cluster. Otherwise, it becomes the center of a new cluster. The effective radius and the distance are computed by using Eq. (6). The classification algorithm is as follows:

Algorithm 2. Bayesian Classification
begin
1. **For** a new point in the relevant result set,
2. Compute $\hat{d}_1(x), \hat{d}_2(x), \ldots, \hat{d}_g(x)$ using Eq. (9)
3. Determine the cluster k where $\hat{d}_k(x) = \max_{1 \leqslant i \leqslant g}\hat{d}_i(x)$
4. **If** $(x-\bar{x}_k)'S_k^{-1}(x-\bar{x}_k) < \chi^2(\alpha)$
5. place it in the cluster k
6. **Else** make it a separate cluster
7. **Endfor**

end

4.3. Cluster-merging stage

Our basic idea of the cluster-merging stage is as follows: (1) At the initial iteration, the hierarchical clustering is used. The level g at the hierarchy of clusters corresponds to g clusters. The initial clusters include only one point in each of them. Consider two clusters at a time. If they are not significantly different, then merge them to one cluster. We repeat this until the number of optimal clusters is found. Generally, the number of optimal clusters is not known (Duda et al., 2001). To estimate the number of optimal clusters, it is necessary to decide which level is more optimal than the other. At each clustering level, Hotelling's T^2 (Johnson and Wichern, 1998) is used to decide which pair of clusters is to be merged. If no merging occurs at g clustering level, then g is the estimated number of clusters.

(2) At other iterations, the clusters after the classification stage can be further merged into bigger clusters. Given g clusters, our cluster-merging algorithm finds candidate pairs of clusters to be merged. Two clusters most likely to be merged should be "close" enough. The algorithm selects the next pair of clusters to be merged until the number of optimal clusters is found. For this purpose, we compare their mean vectors. We infer the merge of the two clusters statistically from the closeness of two mean vectors $\bar{x}_i$ and $\bar{x}_j$. We use Hotelling's T^2 statistics to test the equivalence of two mean vectors of a given pair of clusters.

For the statistical test, let us define:

- the points of ith cluster, $x_{i1}, x_{i2}, \ldots, x_{in_i}$, to be a random sample of size n_i from a population with a mean vector μ_i and a covariance matrix Σ_i.
- the points of jth cluster, $x_{j1}, x_{j2}, \ldots, x_{jn_j}$, to be a random sample of size n_j from a population with the mean vector μ_j and a covariance matrix Σ_j.
- $x_{i1}, x_{i2}, \ldots, x_{in_i}$ to be independent of $x_{j1}, x_{j2}, \ldots, x_{jn_j}$.

Especially, when n_i and n_j are small, we need the following assumptions:

- The populations of the two clusters follow multivariate normal distributions.
- The populations have the same covariance.

We use a pooled covariance to estimate the common covariance since we assume that the population covariances for the two clusters are nearly equal.

Definition 1. Hotelling's T^2 is defined by

$$T^2 = \frac{m_i m_j}{m_i + m_j}(\bar{x}_i - \bar{x}_j)'S_{\text{pooled}}^{-1}(\bar{x}_i - \bar{x}_j), \tag{10}$$

where

$$S_{\text{pooled}} = \frac{1}{m_i + m_j}\left(\sum_{k=1}^{n_i} v_{ik}(x_{ik}-\bar{x}_i)(x_{ik}-\bar{x}_i)' + \sum_{k=1}^{n_j} v_{jk}(x_{jk}-\bar{x}_j)(x_{jk}-\bar{x}_j)'\right). \tag{11}$$

The usual hypothesis to test the location difference is as follows:

$$H_0 : \mu_i = \mu_j \text{ and } H_1 : \mu_i \neq \mu_j,$$

where μ_i is the unknown true center of C_i for $i = 1, \ldots, g$. If T^2 is too big which happens when $\bar{x}_i$ is "too far" from $\bar{x}_j$, then the null hypothesis H_0 is rejected. Note that $T^2 \approx \frac{p(m_i+m_j-2)}{m_i+m_j-p-1}F_{p,m_i+m_j-p-1}(\alpha)$ if H_0 is true. Here $F_{p,m_i+m_j-p-1}(\alpha)$ is the upper $(100(1-\alpha))$th percentile of F-distribution with p and $m_i + m_j - p - 1$ degrees of freedom. Therefore

$$\text{Reject } H_0 \text{ if } T^2 = (\bar{x}_i - \bar{x}_j)'\left[\left(\frac{1}{m_i} + \frac{1}{m_j}\right)S_{\text{pooled}}\right]^{-1} \times (\bar{x}_i - \bar{x}_j) > c^2 \tag{12}$$

where $c^2 = \frac{(m_i+m_j-2)p}{m_i+m_j-p-1}F_{p,m_i+m_j-p-1}(\alpha)$.

In other words, if T^2 is larger than c^2, we conclude that the two clusters are separated.

For example, Fig. 5 shows the cluster-merging procedure for 6 clusters. In order to test closeness of any pair of clusters, given a proper α (99%), the minimum value of T^2 values for all pairs of clusters and c^2 value for corresponding clusters are calculated. If minimum T^2 value is less than c^2 value, then two clusters are close enough and merge them. When 6 clusters are given at certain iteration, among all possible pairs of clusters, we find a pair of clusters with minimum T^2. The pair of clusters with minimum T^2 is the candidate to be merged and their c^2 is calculated for the given α. When T^2 is less than c^2, they are merged into one. Then the number of clusters is reduced by one. This process will be applied until minimum T^2 and c^2 are reversed. Then the number of adjusted clusters becomes 3. That is, we can adjust the number of clusters to be merged by selecting a proper significance level α.

For efficient clustering, we determine the parameters of the merged clusters from those of existing clusters instead of those of points in existing clusters. When clusters are characterized by the mean vector, $\bar{x}_i$, covariance matrix, S_i, the number of elements in the cluster, n_i, and the weight of the cluster, m_i, we characterize a new cluster created by combining clusters i and j with the following statistics (Johnson and Wichern, 1998):

$$m_{\text{new}} = m_i + m_j \tag{13}$$

$$\bar{x}_{\text{new}} = \frac{m_i}{m_{\text{new}}}\bar{x}_i + \frac{m_j}{m_{\text{new}}}\bar{x}_j \tag{14}$$

$$S_{\text{new}} = \frac{m_i - 1}{m_{\text{new}-1}}S_i + \frac{m_j - 1}{m_{\text{new}-1}}S_j + \frac{m_i m_j}{m_{\text{new}}(m_{\text{new}} - 1)} \times [(\bar{x}_i - \bar{x}_j)(\bar{x}_i - \bar{x}_j)'] \tag{15}$$

Algorithm 3. Cluster Merging
begin
1. Compute T^2 values and their c^2 values for all pairs of clusters
2. Select candidate pair of clusters having minimum value of T^2 values
3. **While** minimum T^2 value $\leqslant$ their c^2
4. merge candidate pair of clusters
5. calculate m_{new}, $\bar{x}_{\text{new}}$, S_{new} using Eqs. (13)–(15)
6. select next candidate pair of clusters
7. **EndWhile**

end

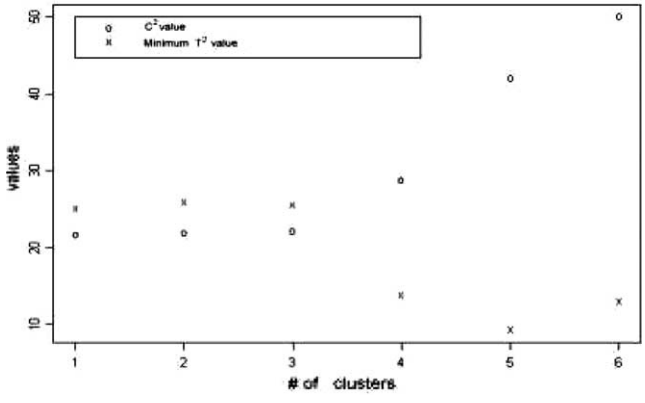

Fig. 5. Scatter plot.

The advantages of using T^2 are as follows:

- T^2 has been verified through various simulations in statistics, and its theoretical properties are well known.
- Our cluster-merging method using T^2 can combine clusters of any shape. Especially, it can be well applied to elliptical clusters.
- To compute T^2, we can use the previous information from the earlier classification stage such as mean vectors, covariance matrices, etc.

Definition 2. Let $\vec{x} = (x_1, \ldots, x_p) \in \Re^p$. An algorithm is invariant under linear transformations if the statistic $U(\vec{x})$ is invariant under linear transformations, that is

$$U(A\vec{x}) = U(\vec{x}),$$

where A is a $p \times p$ matrix with a proper inverse.

Theorem 1. *Algorithm* 2 *and Algorithm* 3 *are invariant under linear transformations.*

Proof 1. It is enough to show that T^2, d^2, and $\hat{d}$ are invariant under the linear transformations. First, let us consider T^2.

$$\begin{aligned} T^2(A\vec{x}) &= \frac{m_i m_j}{m_i + m_j}(A\bar{x}_i - A\bar{x}_j)' S_{\text{pooled}}^{-1}(A\vec{x})(A\bar{x}_i - A\bar{x}_j) \\ &= \frac{m_i m_j}{m_i + m_j}(\bar{x}_i - \bar{x}_j)' A' (AS_{\text{pooled}}A')^{-1} A(\bar{x}_i - \bar{x}_j) \\ &= \frac{m_i m_j}{m_i + m_j}(\bar{x}_i - \bar{x}_j)' A' (A')^{-1} S_{\text{pooled}}^{-1} A^{-1} A(\bar{x}_i - \bar{x}_j) \\ &= T^2(\vec{x}). \end{aligned}$$

The proofs are similar for d^2 and $\hat{d}$. □

Because of this property, the efficiency and quality of the proposed algorithms are almost the same for any linear transformations of circles, which include ellipsoids.

4.4. Dimension reduction

A general problem of similarity retrieval in large image databases is that image/video descriptors are represented by high dimensional vectors. Since most data are from a very high dimension, the singularity of

covariance is troublesome. To reduce the dimension, we use the popular principal components (Duda et al., 2001) instead of the original data.

4.4.1. Principal component analysis

If x is a p dimensional random vector with mean μ and covariance matrix Σ and Γ is the eigenvector matrix of Σ, the principal component transformation is given by

$$z = (x - \mu)'\Gamma$$

where Γ is orthogonal, $\Gamma'\Sigma\Gamma = \Lambda$ is diagonal and $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_p \geqslant 0$. The strict positivity of the eigenvalues λ_i is guaranteed if Σ is positive definite. Let γ_i be ith column vector of Γ. Then $z_i = (x - \mu)'\gamma_i$ and z_i is the ith principal component of x. The variance of z_i is λ_i and the expected value of z_i is 0.

4.4.2. Sample principal components

Let $X = (x_1, \ldots, x_n)'$, S be the sample covariance matrix of X, G be the $p \times p$ eigenvector matrix of S and L be the eigenvalue matrix of S where x_i's are column vectors in R^p and $g_{(i)}$'s are column vectors of G. Then the sample principal component is defined by direct analogy with 4.4.1 as

$$z_{(i)} = (X - 1\bar{x}')g_{(i)}$$

where $S = GLG'$. Putting the sample principal components together we get

$$Z = (X - 1\bar{x}')G$$

G transformed one $(n \times p)$ matrix to another of the same order. L is the covariance matrix of Z.

4.4.3. Hotelling's T^2 with principal components

Recall that

$$T^2(\bar{x}, \bar{y}) = C(\bar{x} - \bar{y})'S^{-1}_{\text{pooled}}(\bar{x} - \bar{y}),$$

where C is a constant and $\bar{x}$ and $\bar{y}$ are used in place of $\bar{x}_1$ and $\bar{x}_2$, respectively. Let $COV_{\text{pooled}}(x,y)$ denote the pooled covariance of x and y. Then

$$\begin{aligned} COV_{\text{pooled}}(G'x, G'y) &= G'S_{\text{pooled}}(G')' = G'S_{\text{pooled}}G \\ &= G'(GLG')G = L. \end{aligned}$$

So,

$$\begin{aligned} T^2(G'\bar{x}, G'\bar{y}) &= C(G'\bar{x} - G'\bar{y})'(G'S_{\text{pooled}}G)^{-1}(G'\bar{x} - G'\bar{y}) \\ &= C(\bar{x} - \bar{y})'G(G'S_{\text{pooled}}G)^{-1}G'(\bar{x} - \bar{y})' \\ &= T^2(\bar{x}, \bar{y}). \end{aligned} \tag{16}$$

By using Theorem 1, $T^2(G'\bar{x}, G'\bar{y}) = T^2(\bar{x}, \bar{y})$, $\hat{d}_i(G'x) = \hat{d}_i(x)$ and $d^2(G'x, G'\bar{x}_i) = d^2(x, \bar{x}_i)$ holds. Let $\bar{z}_x = G'\bar{x}$ and $\bar{z}_y = G'\bar{y}$. We have a simpler form of T^2 with principal components as follows:

$$\begin{aligned} T^2(G'\bar{x}, G'\bar{y}) &= C(\bar{z}_x - \bar{z}_y)'(G'S_{\text{pooled}}G)^{-1}(\bar{z}_x - \bar{z}_y) \\ &= C(\bar{z}_x - \bar{z}_y)'(G'GLG'G)^{-1}(\bar{z}_x - \bar{z}_y) \\ &= C\sum_{j=1}^{p}(\bar{z}_{xj} - \bar{z}_{yj})^2 \Big/ \lambda_j. \end{aligned} \tag{17}$$

Note that T^2 becomes a quadratic form which saves a lot of computing efforts. Likewise, we have a simpler form of $\hat{d}_i$, d^2 with principal components.

4.4.4. Dimension reduction in Hotelling's T^2

The proposed measures such as Eqs. (5), (9) and (10) make use of the sample covariance matrix and its inverse. To resolve the singularity problem, we adopt a new scheme using the diagonal matrix instead of the inverse covariance matrix.

Let us take the first $k \leqslant p$ principal components such that

$$\frac{\lambda_1 + \cdots + \lambda_k}{\lambda_1 + \cdots + \lambda_k + \cdots + \lambda_p} \geqslant 1 - \epsilon,$$

where $\epsilon \leqslant 0.15$. $1 - \epsilon$ is the proportion of total variation covered by the first k principal components. Let G_k be a $(p \times k)$ matrix, where columns are the first k columns of G. Let $\bar{z}_{xk} = G_k'\bar{x}$ and $\bar{z}_{yk} = G_k'\bar{y}$.

$$\begin{aligned} T^2_k(G_k'\bar{x}, G_k'\bar{y}) &= C(\bar{z}_{xk} - \bar{z}_{yk})'(G_k'S_{\text{pooled}}G_k)^{-1}(\bar{z}_{xk} - \bar{z}_{yk}) \\ &= C(\bar{z}_{xk} - \bar{z}_{yk})'(G_k'GLG'G_k)^{-1}(\bar{z}_{xk} - \bar{z}_{yk}) \\ &= C\sum_{j=1}^{k}(\bar{z}_{xkj} - \bar{z}_{ykj})^2 \Big/ l_j. \end{aligned} \tag{18}$$

In this case, Hotelling's T^2 becomes a simple quadratic form. Likewise, we have a similar form of d^2, $\hat{d}_i$.

4.5. Quality of clustering

A good way of measuring the quality of the proposed classification method is to calculate its "classification error rates," or misclassification probabilities. Our method of measuring the clustering quality is as follows:

After the number of clusters is fixed at the final iteration, take out one element of a cluster. Check if the element is classified into the previous cluster again according to the classification procedure. Let C be the number of elements classified correctly to its own cluster and N be the total number of elements in all clusters. The error-rate becomes $1 - C/N$. This method can be applied even though numbers of elements of the cluster are small.

5. Experimental evaluation

For the experiments, the Corel image collection and the synthetic data are used as the test set of data. The original image collection was obtained from Corel

Corporation (http://corel.digitalriver.com). It includes 30,000 color images. Its images have been classified into distinct categories by domain experts, and there are about 100 images in each category. In the experiments, we use high-level category information as the ground truth to obtain the relevance feedback since the user wants to retrieve the images based on high-level concepts, not low-level feature representations (Wang et al., 2001). That is, images from the same category are considered most relevant, and images from related categories (such as flowers and plants) are considered relevant.

For the purpose of the performance evaluation of the query clustering approach (Qcluster), we have conducted this experiment with two goals. First, evaluate Qcluster to answer multipoint k-NN queries and compare it to the query point movement (QPM) and the query expansion approach (QEX). Second, test that the proposed Qcluster algorithm converges quickly to the user's true information needs. Precision-recall curve is used to measure the retrieval performance. Precision is defined as the number of retrieved relevant images over the number of total retrieved images. Recall is defined as the number of retrieved images over the total number of relevant images. We have implemented Qcluster in C++ on a Sun Ultra II. First, we consider the real data. In our system, we use two visual features: color moments and co-occurrence matrix texture. For color moments, we use the HSV color space because of its perceptual uniformity of color. For each of three color channels, we extract the mean, standard deviation, and skewness, and reduce the length of the feature vector to three using the principal component analysis. Then, we use the three dimensional feature vector as the color feature.

For the co-occurrence matrix texture, the (i,j)th element of co-occurrence matrix is built by counting the number of pixels, the gray-level (usually 0–255) of which is i and the gray-level of its adjacent pixel is j, in the image. Texture feature values are derived by weighting each of the co-occurrence matrix elements and then summing these weighted values to form the feature value. We extract a vector of the texture feature whose 16 elements are energy, inertia, entropy, homogeneity, etc (Porkaew and Chakrabarti, 1999) and reduce the length of feature vector to four using the principal component analysis.

In the experiments, we generate 100 random initial queries and evaluate the retrieval quality for a sequence of iterations starting with these initial queries. We perform five feedback iterations in addition to the initial query. All the measurements are averaged over 100 queries. When the cluster merging process is performed, $\alpha = 0.01$, as the given significance level (99%), is used to test closeness of any pair of clusters. The k-NN query is used to accomplish the similarity-based match, and we set k to 100. We use the hybrid tree (Chakrabarti and Mehrotra, 1999) to index feature vectors of the whole data and fix the node size to 4 KB.

Fig. 6 compares the CPU cost of an inverse matrix scheme and a diagonal matrix scheme for the Qcluster approach when color moments are used as a feature. The diagonal matrix scheme of the Qcluster approach significantly outperforms the inverse matrix scheme in terms of CPU time. Therefore, we use a diagonal matrix scheme in our method.

Fig. 7 compares the execution cost for the three approaches. The proposed Qcluster shows the similar performance with the multipoint approach (Chakrabarti et al., 2000) and outperforms the centroid-based approach such as MARS (Chakrabarti and Mehrotra, 1999) and FALCON (Wu et al., 2000). This is because our k-NN search is based on the multipoint approach

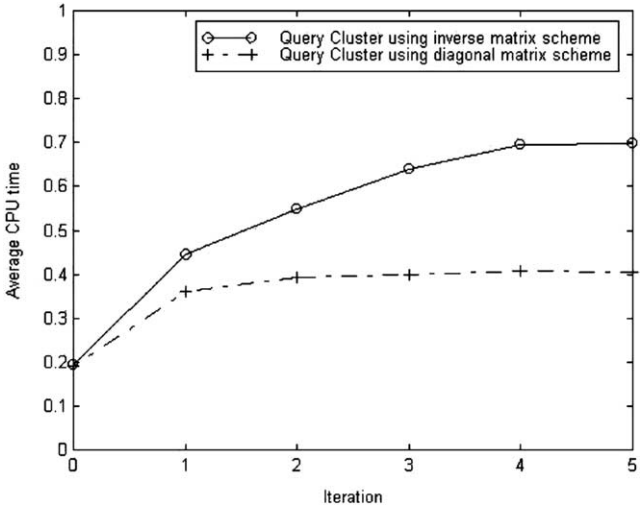

Fig. 6. CPU time for inverse and diagonal matrix schemes in query cluster approach.

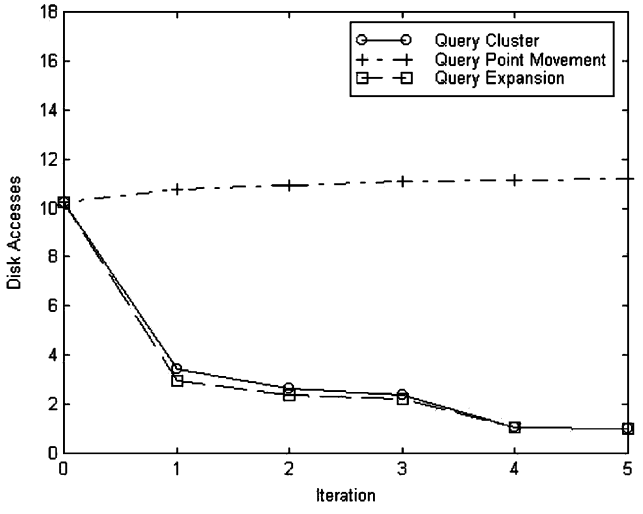

Fig. 7. Comparison of execution cost for the three approaches.

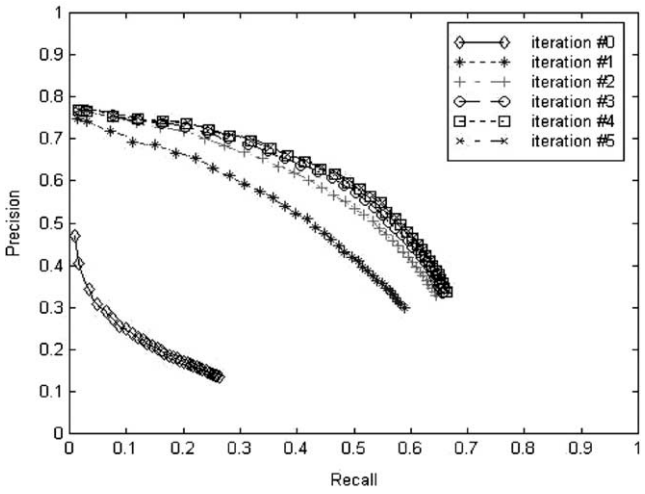

Fig. 8. Precision recall graph for query clustering when color moments is used.

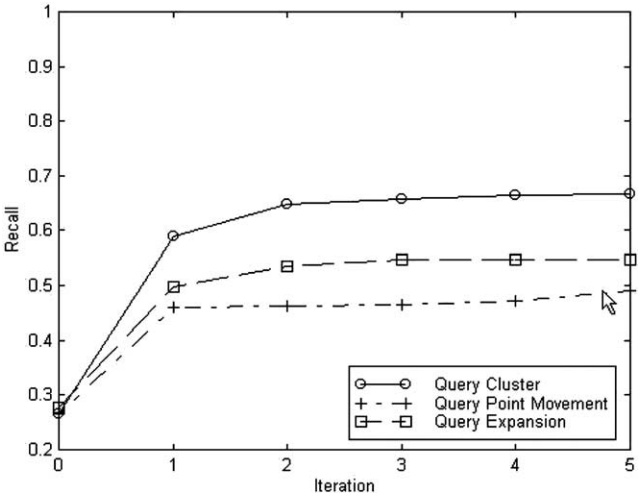

Fig. 10. Comparison of recall for the three approaches when color moments is used.

that saves the execution cost of an iteration by caching the information of index nodes generated during the previous iterations of the query.

Figs. 8 and 9 show the precision-recall graphs for our method when color moments and co-ocurrence matrix texture are used, respectively. In these graphs, one line is plotted per iteration. Each line is drawn with 100 points, each of which shows precision and recall as the number of retrieved images increases from 1 to 100. Based on these figures, we make two observations as follows:

- The retrieval quality improves at each iteration.
- The retrieval quality increases most at the first iteration. At the following iterations there are minor increases in the retrieval quality. This ensures that our method converges quickly to the user's true information need.

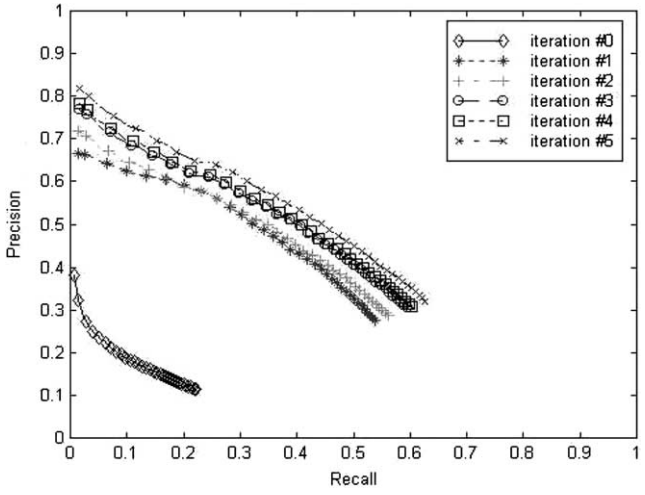

Fig. 9. Precision recall graph for query clustering when co-occurrence matrix texture is used.

Figs. 10 and 11 compare the recall for query clustering, query point movement, and query expansion at each iteration. Figs. 12 and 13 compare the precision for the three approaches. They produce the same precision and the same recall for the initial query. These figures show that the precision and the recall of our method increase at each iteration and outperform those of the query point movement and the query expansion approach.

Next, we performed extensive experiments to measure the accuracy of the adaptive classification algorithm and that of the cluster-merging algorithm using the synthetic data. Let $z = (z_1, \ldots, z_p)$ in $\Re^p$ where $z_1, \ldots, z_p$ are independent and identically distributed with $N(0,1)$. Then z is a multi-variate normal with a mean vector 0 and a covariance matrix I, and the data shape of z is a

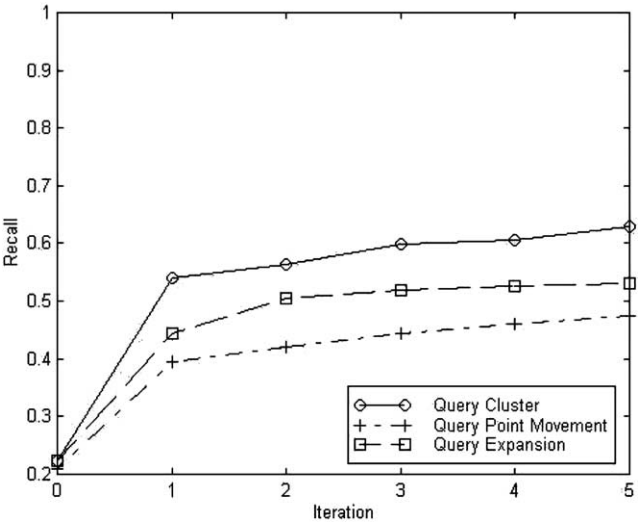

Fig. 11. Comparison of recall for the three approaches when co-occurrence matrix texture is used.

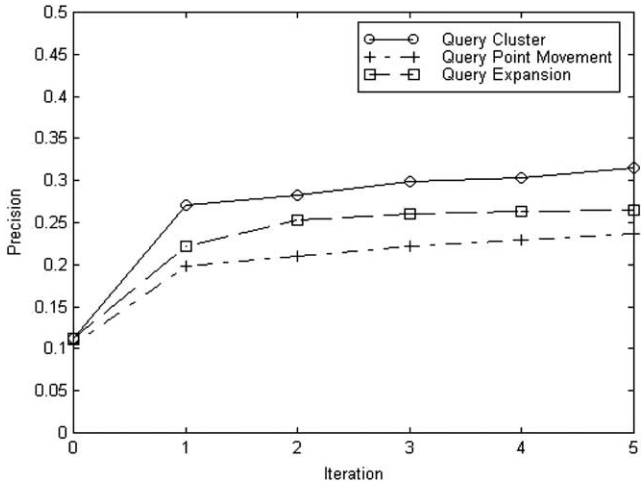

Fig. 12. Comparison of precision for the three approaches when color moments is used.

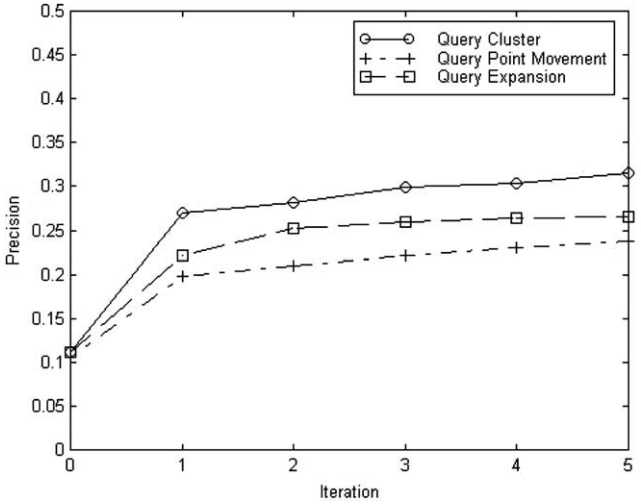

Fig. 13. Comparison of precision for the three approaches when co-occurrence matrix texture is used.

sphere. Let $y = Az$. Then $COV(y) = AA'$ and the data shape of y is an ellipsoid. The synthetic data in $\Re^{16}$ are generated. The data consist of 3 clusters and their inter-cluster distance values vary from 0.5 to 2.5. Then the principal component analysis is used to reduce the dimension of them from 16 to 12, 9, 6, 3, respectively.

We calculate error rates of the classification algorithm (*Algorithm* 2) with respect to 12, 9, 6, 3 dimensional data. Fig. 14(a) shows those for spherical data, and Fig. 14(b) shows those for elliptical data when we use an inverse matrix in the Bayesian classifier. Fig. 15(a) shows those for spherical data and Fig. 15(b) shows those for elliptical data when we use a diagonal matrix instead. The result shows that the error rate decreases as the inter-cluster distance value increases and the error rate increases as the dimension decreases for the same inter-cluster distance value. The reason is that the information loss increases as the proportion of total variation covered by the k principal components ($k = 12, 9, 6, 3$) decreases. Importantly, figures show that the quality of the classification algorithm stays almost the same regardless of the data shape. This result confirms the linear transformation invariant property of the proposed classification algorithm.

Next, we compute the error-ratios of the T^2-statistic with an inverse matrix and those with a diagonal matrix in order to measure the accuracy of cluster-merging algorithm with respect to 12, 9, 6, 3 dimensional data. Given 100 pairs of clusters of size 30, 100 T^2 values and corresponding critical distance (c^2) values are computed. Quantile-F values in Tables 2 and 3 are the critical distance values given by the 95th percentile $F_{p,n-p}(0.05)$ where p is a dimension and n is the number of objects. If T^2 value is larger than corresponding c^2 value, reject H_0. That is, we decide that a pair of clusters must be separated. If a pair of clusters is close, then the error ratio increases in case of separating them. Tables 2 and 3 show the average T^2 and the average error ratio

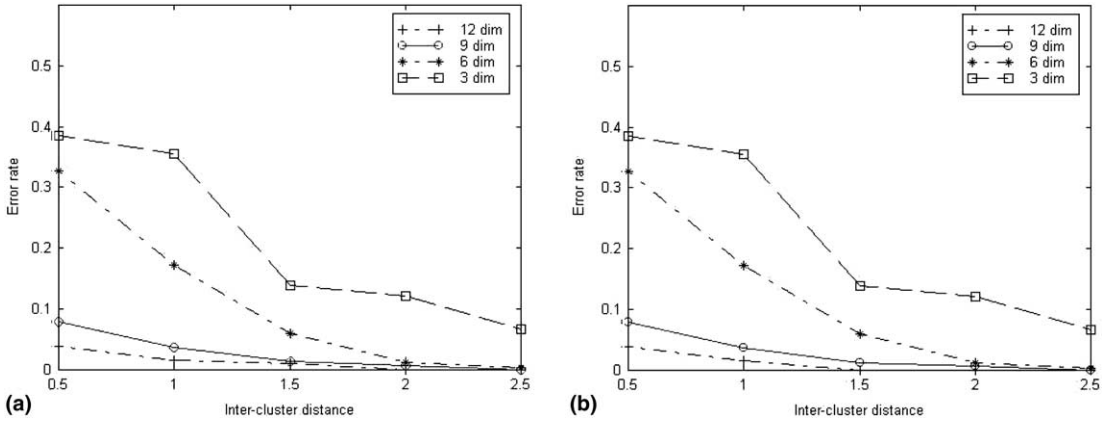

Fig. 14. Error rate of the classification algorithm using an inverse matrix: (a) for 3 clusters of the spherical shape; (b) for 3 clusters of the elliptical shape.

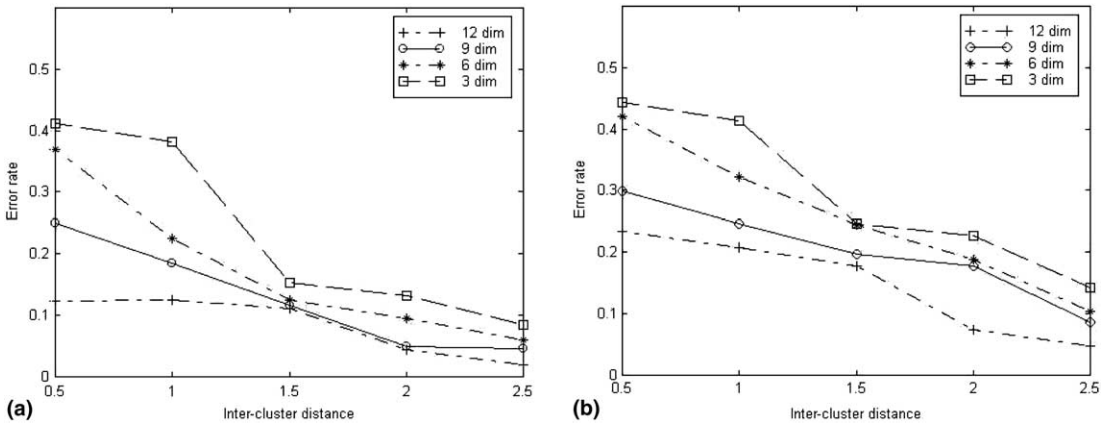

Fig. 15. Error rate of the classification algorithm using a diagonal matrix: (a) for 3 clusters of the spherical shape; (b) for 3 clusters of the elliptical shape.

Table 2
Comparison of T^2 with inverse matrix and T^2 with diagonal matrix when each pair of clusters have same means

Dim	Variation ratio	T^2	Quantile-F	Error-ratio (%)
		T^2 with inverse matrix		
12	0.996	0.77	1.96	0
9	0.97	1.02	2.07	1
6	0.96	0.79	2.28	2
3	0.94	0.44	2.77	2
		T^2 with diagonal matrix		
12	0.996	0.70	1.96	2
9	0.97	0.87	2.07	4
6	0.96	0.68	2.28	6
3	0.94	0.44	2.77	6

Table 3
Comparison of T^2 with inverse matrix and T^2 with diagonal matrix when each pair of clusters have different means

Dim	Variation ratio	T^2	Quantile-F	Error-ratio(%)
		T^2 with inverse matrix		
12	0.996	20.54	1.96	0
9	0.97	24.17	2.07	0
6	0.96	31.01	2.28	0
3	0.94	38.29	2.77	6
		T^2 with diagonal matrix		
12	0.996	28.37	1.96	0
9	0.97	25.03	2.07	1
6	0.96	31.27	2.28	2
3	0.94	41.20	2.77	8

(%) with respect to 12, 9, 6, 3 dimensional data for T^2 with an inverse matrix and T^2 with a diagonal matrix.

Figs. 16–18 show the accuracy of the cluster-merging algorithm. For simulation, the random seven clusters of size 30 were generated from bivariate distributions in $\Re^2$. Initially, the hierarchical clustering is used to build nine clusters from the synthetic data. Left figures show scatter diagrams when their inter-cluster distance values vary from 2 to 6. In right figures, x-axis denotes the number of clusters and y-axis denotes critical distance values (marked as 'o') and T^2 value (marked as 'x') for the nearest pair of clusters at each clustering level. From clustering level 9 to 1, the cluster-merging algorithm decides whether a pair of nearest clusters are merged. Fig. 16 shows the result that our cluster-merging algorithm finds three clusters. Figs. 17 and 18 show our algorithm finds seven clusters, respectively. That is, the T^2 and the c^2 used in our cluster-merging algorithm (*Algorithm* 3) are useful in deciding whether to merge a pair of close clusters.

6. Conclusion

We have focused on the problem of finding multiple clusters of a complex image query, based on the relevance feedback, to guess the distance function and the ideal query point that the user has in mind. Our approach consists of two steps: (1) an adaptive classification that attempts to place relevant images in the current clusters or new clusters, and (2) cluster-merging that reduces the number of clusters by merging certain clusters to reduce the number of query points in the next iteration.

The major contribution of this approach is the introduction of unified quadratic forms for the distance function, the adaptive classifier, and the cluster-merging measure. Their benefit is to achieve the same high retrieval quality regardless of shapes of clusters of a query since they are invariant under linear transformations in the feature space.

Our experiment shows that the proposed techniques provide a significant improvement over the query point movement and the query expansion in terms of the retrieval quality.

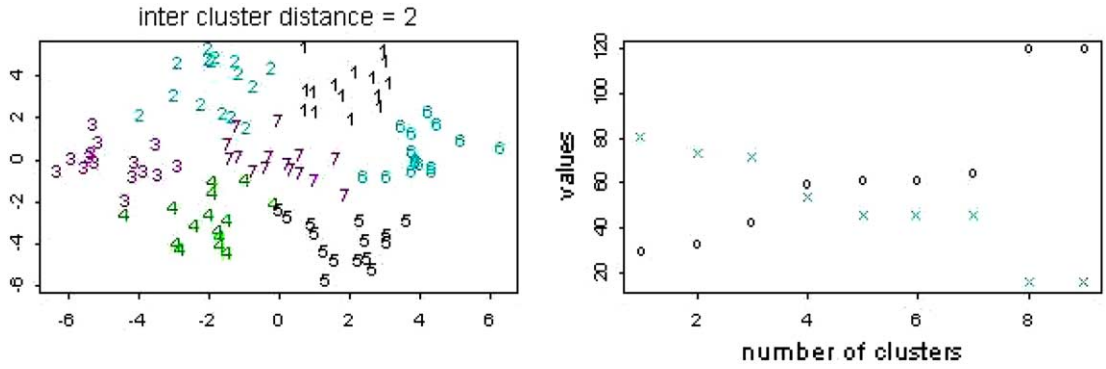

Fig. 16. Scatter-diagram and effect of cluster-merging algorithm when inter-cluster distance is 2.

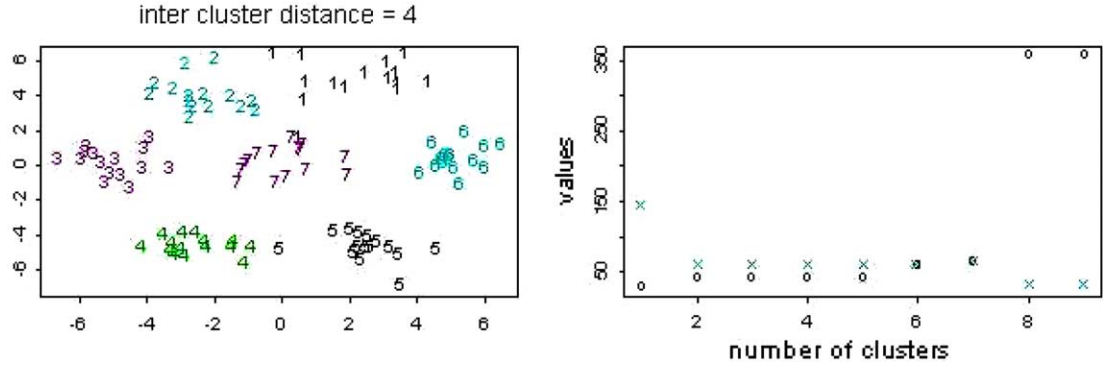

Fig. 17. Scatter-diagram and effect of cluster-merging algorithm when inter-cluster distance is 4.

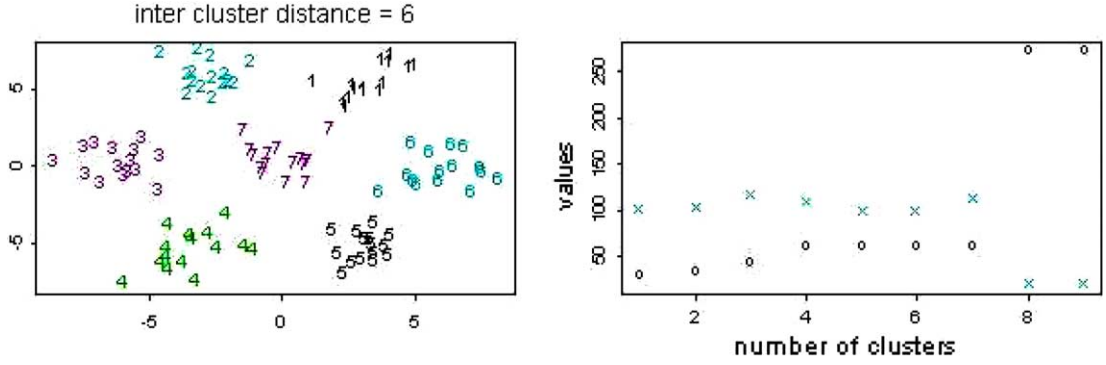

Fig. 18. Scatter-diagram and effect of cluster-merging algorithm when inter-cluster distance is 6.

Acknowledgement

This work was supported by the Korea Research Foundation Grant (M01-2003-000-20068-0). Dr. Deok-Hwan Kim is currently visiting Department of Computer Science at University of Arizona, USA. This research was partly supported by the Agency for Defense Development, Korea, through the Image Information Research Center at Korea Advanced Institute of Science & Technology.

References

Ashwin, T.V., Gupta, R., Ghosal, S., 2002. Adaptable similarity search using non-relevant information. In: Proceedings of the 28th VLDB Conference, August, Hong Kong, China.

Bartolini, I., Ciacci, P., Waas, F., 2001. Feedbackbypass: A new approach to interactive similarity query processing. In: Proceedings of the 27th VLDB Conference, Roma, Italy, pp. 201–210.

Bauer, M., Gather, U., Imhoff, M., 1999. The Identification of Multiple Outliers in Online Monitoring Data. Technical Report 29, SFB 475, University of Dortmund.

Benitez, A.B., Beigi, M., Chang, S.-F., 1998. A content-based image meta-search engine using relevance feedback. IEEE Internet Computing 2 (4), 59–69, July/August.

Brunelli, R., Mich, O., 2000. Image retrieval by examples. IEEE Transactions on Multimedia 2 (3), 164–171, September.

Chakrabarti, K., Mehrotra, S., 1999. The hybrid tree: An index structure for high dimensional feature spaces. In: Proceedings of 5th Int'l Conference on Data Engineering, Sydnet, Australia, pp. 440–447.

Chakrabarti, K., Porkaew, K., Mehrotra, S., 2000. Efficient query refinement in multimedia databases. In: Proceedings of the IEEE Int'l Conference on Data Engineering, February, San Diego, USA, p. 196.

Charikar, M., Chekuri, C., Feder, T., Motwani, R., 1997. Incremental clustering and dynamic information retrieval. In: Proceedings of the ACM STOC Conference, pp. 626–635.

Duda, R.D., Hart, P.E., Stork, D.G., 2001. Pattern Classification. John Wiley & Sons Inc., New York.

Fisher, R.A., 1938. The statistical utilization of multiple measurements. Annals of Eugenics 8, 376–386.

Flickner, M., Sawhney, H., Niblack, W., et al., 1995. Query by image and video content: The QBIC system. IEEE Computer Magazine 28 (9), 23–32, Sep.

Ishikawa, Y., Subramanya, R., Faloutsos, C., 1998. Mind Reader: Querying databases through multiple examples. In: Proceedings of the 24th VLDB Conference, New York, USA, pp. 218–227.

Johnson, R.A., Wichern, D.W., 1998. Applied Multivariate Statistical Analysis. Prentice-Hall, NJ.

Porkaew, K., Chakrabarti, K., 1999. Query refinement for multimedia similarity retrieval in MARS. In: Proceedings of the 7th ACM Multimedia Conference, Orlando, Florida, pp. 235–238.

Rocchio, J.J., 1971. Relevance feedback in information retrieval. In: Salton, G. (Ed.), The SMART Retrieval System—Experiments in Automatic Document Processing. Prentice Hall, Englewood Cliffs, NJ, pp. 313–323.

Rui, Y., Huang, T., Mehrotra, S., 1997. Content-based image retrieval with relevance feedback in MARS. In: Proceedings of IEEE International Conference on Image Processing '97, October, Santa Barbara, CA.

Rui, Y., Huang, T., Ortega, M., Mehrotra, S., 1998. Relevance feedback: A power tool for interactive content-based image retrieval. IEEE Transactions on Circuits and Systems for Video Technology 8 (5), 644–655.

Rui, Y., Huang, T., Chang, S.F., 1999. Image Retrieval: current techniques, promising directions and open issues. Journal of Visual Communication and Image Representation 10, 39–62, March.

Salton, G., Fox, E.A., Wu, H., 1983. Extended boolean information retrieval. CACM 26 (11), 1022–1036, November.

Smith, J.R., Chang, S.F., 1996. VisualSEEk: A fully automated content-based image query system. In: Proceedings of the ACM Int'l Multimedia Conference, November, pp. 87–98.

Wang, T., Rui, Y., Hu, S.-M., 2001. Optimal adaptive learning for image retrieval. In: Proceedings of IEEE CVPR 2001, Kauai, Hawaii, pp. 1140–1147.

Wu, L., Faloutsos, C., Sycara, K., Payne, T.R., 2000. FALCON: Feedback adaptive loop for content-based retrieval. In: Proceedings of the 26th VLDB Conference, Cairo, Egypt, pp. 297–306.

Zhou, X.S., Huang, T.S., 2001. Comparing discriminating transformations and SVM for learning during multimedia retrieval. In: Proceedings of the 9th ACM Multimedia Conference, Orlando, Florida, pp. 137–146.