가변적인 길이의 특성 정보를 지원하는 특성 가중치 조정 기법

(A Feature Re-weighting Approach for the Non-Metric Feature Space)

Robert Samuel Lee[†] 김 상 희[††] 박 호 현[†††]
(Robert Samuel Lee) (Sang-Hee Kim) (Ho-Hyun Park)

이 석 룡[††††] 정 진 완[†††††]
(Seok-Lyong Lee) (Chin-Wan Chung)

요 약 이미지 데이타베이스 분야에 대한 다양한 기법들 가운데, 내용 기반 영상 검색 기법(Content Based Image Retrieval)은 대용량의 영상을 효율적으로 검색하고 탐색할 수 있도록 한다. 기존의 내용 기반 영상 검색 시스템은 사용자가 입력한 질의 이미지에서 낮은 레벨의 특성(low-level feature)을 추출하고 그에 기반하여 데이타베이스로부터 유사한 영상을 검색한다. 하지만 컴퓨터에서 사용하는 낮은 레벨의 특성은 실제 인간이 영상을 인식하는 방법과 다르게 영상을 인식한다는 단점이 있다. 이러한 단점을 보완하기 위하여 각 특성에 대한 가중치를 적합성 피드백(relevance feedback)을 통하여 재조정하는 기법이 개발되었다. 기존의 특성 가중치 조정(feature re-weighting) 기법은 모든 영상에 대하여 특성은 항상 고정된 길이의 벡터 데이타로 표현된다고 가정한다. 이러한 가정을 전제로 하여 기존의 기법은 특성 표현(feature representation)의 각 부분을 n차원 공간의 각 축에 할당한다. 하지만 특성 표현 기법의 발전에 따라 가변적인 길이의 벡터로 표현되는 특성이 출현하였으며, 이로 인하여 기존의 제한된 길이의 벡터로 표현되는 특성 표현에 기반한 특성 가중치 조정 기법의 유효성은 감소하게 되었다.

본 논문에서는 가변적인 크기의 벡터로 표현되는 특성에 대해서도 특성 가중치를 효과적으로 조정할 수 있는 기법을 제안한다. 본 기법은 특성에 기반하여 계산된 질의 영상과 데이타베이스 내부의 영상 간의 거리와 양방향 신뢰구간을 이용하여 특성 가중치를 조정한다. 이 때 각 특성의 거리 계산 방법에 대해서는 제한을 두지 않는다. 또한 각 특성의 표현에 있어서도 고정적인 크기 뿐만이 아니라 가변적인 크기의 데이타 역시 사용할 수 있도록 한다. 본 논문에서는 실험을 통하여 제안한 기법의 유효성을 입증하였으며, 다른 연구 결과와의 비교를 통하여 제안한 기법의 성능이 보다 우수함을 보였다.

키워드 : 내용 기반 영상 검색, 적합성 피드백, 특성 가중치 조정, 모양 특성

Abstract Among the approaches to image database management, content-based image retrieval (CBIR) is viewed as having the best support for effective searching and browsing of large digital image libraries. Typical CBIR systems allow a user to provide a query image, from which low-level features are extracted and used to find "similar" images in a database. However, there exists the semantic gap between human visual perception and low-level representations. An effective methodology for overcoming this semantic gap involves relevance feedback to perform feature re-weighting. Current approaches to feature re-weighting require the number of components for a

· This research was supported by the Agency for Defense Development, Korea, through the Image Information Research Center at Korea Advanced Institute of Science and Technology.

† 학생회원 : 한국과학기술원 전산학전공
robbie@islab.kaist.ac.kr
†† 정 회 원 : 국방과학연구소 지형영상처리팀 책임연구원
falcon@add.re.kr
††† 정 회 원 : 중앙대학교 전자전기공학부 교수
hohyun@cau.ac.kr
†††† 정 회 원 : 한국외국어대학교 산업정보시스템공학과 교수
sllee@hufs.ac.kr
††††† 종신회원 : 한국과학기술원 전산학전공 교수
chungcw@cs.kaist.ac.kr

논문접수 : 2005년 12월 7일
심사완료 : 2006년 3월 27일

feature representation to be the same for every image in consideration. Following this assumption, they map each component to an axis in the n-dimensional space, which we call the metric space; likewise the feature representation is stored in a fixed-length vector. However, with the emergence of features that do not have a fixed number of components in their representation, existing feature re-weighting approaches are invalidated.

In this paper we propose a feature re-weighting technique that supports features regardless of whether or not they can be mapped into a metric space. Our approach analyses the feature distances calculated between the query image and the images in the database. Two-sided confidence intervals are used with the distances to obtain the information for feature re-weighting. There is no restriction on how the distances are calculated for each feature. This provides freedom for how feature representations are structured, i.e. there is no requirement for features to be represented in fixed-length vectors or metric space. Our experimental results show the effectiveness of our approach and in a comparison with other work, we can see how it outperforms previous work.

Key words : content-based image retrieval, relevance feedback, feature re-weighting, shape features

1. 서 론

The simplicity in generating digital images has led to the point where the management of the growing collections of images is no long a simple manual task. This is not limited to individuals, but includes businesses as well. This trend has led to the emergence of numerous repositories for the storage, organization, and distribution of such images.

Although research on content-based image retrieval (CBIR) has been active since the early 1990s, the approach is still considered to be in its infancy. Nonetheless, it is considered to best support effective searching and browsing of large digital image libraries.

Searching for images with a CBIR system requires the user provide a query image. The system obtains a low-level description of the image, which it then uses to compare to the low-level descriptions of the images in the database. However, the performance of CBIR is still impeded by the semantic gap between the high-level user concept and low-level representation. For example, if a user provides an image of a green apple as a query, the system could return images of tennis balls, lily pads, turtles, etc., when using colour features.

One approach to overcome this shortcoming is to incorporate the use of numerous features; enough to ensure that every visual semantic of the image contents can be captured. Although a system may utilize all the features describing the images over a set of queries, the use of one feature may better identify similar images for a given query image, whereas another feature may better identify similar images for a different query image.

Relevance feedback has become an effective methodology for bridging the semantic gap. Incorporating relevance feedback into CBIR systems leads to an image search becoming an interactive session. The system utilizes user feedback to update the feature weights and retrieve a new set of results that is better than the previous result set. Following the scenario above, assume a user provides an image of a green apple as a query and the system uses both color and shape features. With all the features equally weighted, the system returns images of red and green apples, tennis balls, lily pads and frogs. The user then marks the images of the apples as being relevant and requests a new set of images. The system uses this feedback to update the feature weights and returns a new set of results that should now contain more apple images and fewer non-apple images.

Current techniques[1-6] have a shortcoming. They require the number of components for each feature representation to be the same among all images. The corresponding feature components of the images displayed to the user are analysed to obtain the feature re-weighting information. Also, these feature components must be mapped to an axis in the n-dimensional space (metric space) for distance calculation. Having done so, the weighted Euclidean distance is used to compute the distance between the features for each image.

Until recently, colour and texture features were most widely used to describe image content. Colour features are generally represented using histograms, where the components would be the frequency of each bin. Thus, using the colour feature is supported with existing feature re-weighting approaches since the number of bins would be the same for the colour histogram of each image. However, colour and texture features are not always successful in capturing visual concepts. Weather (e.g. snow coverage) and lighting are some factors in the effectiveness of colour and texture features. With the development of shape features and high-level features that incorporate the spatial relations of objects within an image, the assumption that the number of components for a feature representation is the same for all images is no longer valid. Examples of such features are Curvature Scale Space (CSS) [7], and 9D-SPA [8]. The CSS feature represents the maxima of the zero-crossing points of a contour. The number of maxima increases with the complexity of a contour, e.g., an ellipse would have 0 maxima and a star-like contour would have many maxima. 9D-SPA represents the spatial relations between objects in an image. Thus, the number of components in a 9D-SPA representation differs between images with different numbers of objects. Since the feature representations may not have corresponding components between images, it is no longer possible to compare images for each feature component to obtain feature re-weighting information. Likewise, the feature components can no longer be mapped to corresponding axes in the metric space, thus invalidating the use of the weighted Euclidean distance.

To support feature re-weighting for such features, we do not compare images for each feature component. Instead, we analyze the value of the distances that are calculated with each feature's own distance function. By analysing the distances, there is no restriction placed on the structure of the components of a feature. In addition, there is no requirement for feature components to be mapped to a metric space to calculate the distances; there is complete freedom in the distance function used for a given feature. The proposed feature re-weighting technique displays the following characteristics:

- *Support for features regardless of whether or not they can be mapped to an n-dimensional space (metric space).* The combination of features used for the experiments shows that it can effectively handle features regardless of whether or not they have the same number of components between images and can be mapped to a metric space.
- *Effective mapping of visual similarity to low-level features.* On the average, the recall of the proposed approach was better than an existing approach by 98% at the 2^{nd} iteration, 50% by the 5^{th} iteration, 45% for the 10^{th} iteration, and 42% for the 20^{th} iteration of retrieval.

The remainder of this paper is organized as follows. Section 2 provides an overview of the research background by providing a description of CBIR systems, the current view of feature re-weighting, and related work. In Section 3 we cover the proposed feature re-weighting technique. Then in Section 4 we show its effectiveness with the experimental results. A conclusion is provided in Section 5.

2. Research Background

2.4 CBIR Systems

The two main functions of a CBIR system are image storage and image search. Figure 1 shows a general architectural diagram for typical CBIR systems. For storage, images are first segmented. At this stage, regions or objects are identified within the image. Segmented images are then passed to the feature extractor. The feature extractor extracts the feature data for each object detected in the image. This data is then passed to a data manager which stores the feature data and the image into a database. For image search, the user provides an image for the query. This image is segmented and features are extracted. The distance calculator then compares the feature data of the query image to the feature data in the database. The closest matches are then displayed to the user.

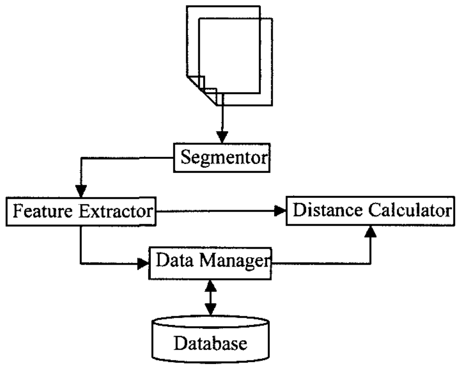


Fig. 1 Architecture of CBIR Systems

2.2 Relevance Feedback

Relevance feedback techniques eliminate from the user the responsibility of having to determine the important features for a given query by automatically updating the feature weights. This is required as a user should not be expected to have prior knowledge of the image collection. Updated feature weights are used for each of retrieval iteration. This cycle continues until the resulting set of images ceases to change (i.e. convergence) or the user is satisfied with the results. Take note, the relevance feedback mechanisms are used starting from the second retrieval iteration, once the system receives feedback from the user.

Existing feature re-weighting techniques are becoming invalidated with the development of new feature representations because they have the following view on feature re-weighting.

If a feature is made up of n components, then the feature component values of each image can be mapped to a metric space. Initially, the scale of each axis is equal. With each iteration of relevance feedback, the axes can be scaled by shrinking the more important feature component axes and expanding the less important feature component axes. Take for example Figure 2. In (a) we see the initial plot of the feature components for some feature of image A and image B. From the user feedback, the system determines that the y-component of the feature is less important than the x-component. The system uses this feedback to update the feature space. Since the x-component is more important, its axis is compressed. The opposite occurs for the axis corresponding to y-component. It is found to be less important in determining similarity and is thus stretched. The result of the changes to the feature space is shown in (b). With the scaling of the axes, the point corresponding to image B is brought closer to the query point.

Notice that the effect illustrated with Figure 2 assumes that the feature in consideration has exactly 2 components for both image A and B. However, considering features such as CSS, the number of feature components extracted from each image may differ. Thus, approaches that assume feature data can be mapped to a metric space do not support such features.

2.3 Related Work

Although research in CBIR is still considered to be in its infancy, there exists a great deal of prior work. The first commercial CBIR system to be developed is IBM's QBIC [9] which allows a user to search for images using colour, texture, shape and text. VisualSEEk [10] introduced the identifi-

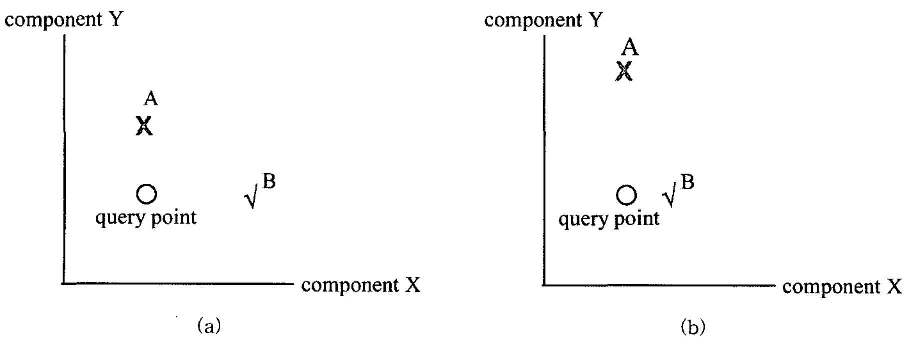


Figure 2 Effect of Feature Re-weighting

cation of regions and objects within an image, which provided significant gains in similarity retrieval compared to using global features. Since then, systems that demonstrate improved segmentation, feature extraction, distance calculation, and indexing techniques have been developed. However, none of those systems can overcome the semantic gap without the use of relevance feedback.

MARS [1,2] proposes a method where the top k results are returned to the user and the feedback is used to refine the feature weights using both the variance in feature values and set intersection. The most cited approach to re-weighting features is proposed in MARS and has been applied in numerous other CBIR systems. In their model they assume a feature representation is comprised of multiple components. Thus, a feature representation can be thought of as a vector of real values (e.g. color histogram). Their model is illustrated in Figure 3.

The W_i correspond to the weight for a feature and the w_{ij} are the weights corresponding to the feature components that comprise feature i. For each of level in the model, a different re-weighting technique is performed. At the representation level, the weight for the j^{th} feature component of the i^{th} feature representation is updated to the inverse of the standard deviation of the component values from the relevant images, i.e., $w_{ij} = \frac{1}{\sigma_{ij}}$.

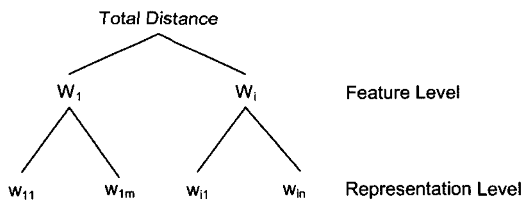


Figure 3 Feature Weighting Model of MARS

When each feature component weight has been updated, they are normalized by total weight to make them sum to 1. These weights are then placed in a diagonal matrix W so that the distance for feature i can be calculated using the weighted Euclidean distance as follows:

$$\text{feature distance}_i = (\vec{q_i} - \vec{f_i})^T W (\vec{q_i} - \vec{f_i})$$

where $\vec{q_i}$ contains the feature component values for the query image and $\vec{f_i}$ contains the feature component values for an image in the database. Note that this level of re-weighting assumes that the feature representation can be mapped to a metric space. More specifically, this approach assumes that the number of components j for each feature representation, is the same among all images. However, as mentioned previously, with features such as CSS, this is not valid.

Updating the weight for each feature is considered inter-feature re-weighting. These weights reflect the emphasis of a feature representation in the overall distance comparisons. For each iteration of retrieval, the set of returned images is compared to the sets of similar images with respect to each individual feature.

Let S and S_i, where i = 1, ..., # of features, be the sets of k most similar images determined using the all features combined, and distance based on only feature i, respectively:

$$S = [img_1, \cdots, img_k]$$
$$S_i = [img_1^i, \cdots, img_k^i]$$

Scores are assigned to the retrieved images, those in set S, by the user. The scoring scheme is set arbitrarily, where the score is based on the amount of relevance. To update the weight for feature i, W_i is first initialized to 0. The following procedure is then performed.

```
for x = 1 to k
    if (img_x^i exists in S)
        W_i = W_i + score
```

The inter-feature re-weighting process involves performing an intersection of set S with each set S_i. The scores assigned to the images in the intersection are then summed. Having done so, the resulting weights (W_i's) are normalized by total weight to sum to 1. The feature distances are multiplied by the respective W_i's to obtain the overall distance.

One inherent problem with the MARS approach is that a large number of relevant images are required in the retrieved set for there to be any substantial amount of refinement in the feature weighting. In addition, the scoring scheme for inter-feature re-weighting is determined arbitrarily. Also, the information obtained to re-weight the features is limited to the size of the intersections in the set intersection procedure. As mentioned previously, the intra-feature re-weighting also assumes the feature data can be mapped to metric space and that each image has corresponding feature components.

The approach proposed in [3] uses the same feature weighting model and distance calculation as MARS. However, its process of intra-feature includes the use of non-relevant images. A discriminant ratio (dr_{ij}) is used to determine the ability of component j of feature i in separating non-relevant and relevant images and is defined as follows:

$$dr_{ij} = 1 - \frac{\sum_{l=1}^{m} or(f_{ij}^{non-rel,l})}{\sum_{l=1}^{m} f_{ij}^{non-rel,l}}$$

where m is the number of non-relevant images and $or(f_{ij}^{non-rel,l})$ is the value of the j^{th} component of the l^{th} non-relevant image that is outside the range of values for the j^{th} component of relevant images, and 0 otherwise. The weight w_{ij} for the j^{th} component of the i^{th} feature representation is then determined as follows:

$$w_{ij} = \frac{dr_{ij}}{\sigma_{ij}^{rel}}$$

where σ_{ij}^{rel} is the standard deviation of the j^{th} component of the i^{th} feature among the relevant images. The inter-feature re-weighting, or determination of the weights for each feature, is performed using the following equation.

$$Wi = \sum_{k=1}^{K} \sqrt{\frac{\partial_k}{\partial_i}}$$

where K is the number of features and ∂_k is the total distance between the k^{th} feature of the query image and those of the relevant images.

Other recent work, including iPure [4], Mind-Reader [5], and [6] also propose methods of feature re-weighting. However, their approaches are fully dependent on the assumption that the feature vector for all images must be of the same length. These approaches are invalidated by features that do not have a fixed number of feature components between images.

The distance calculation for CSS cannot be calculated by mapping its components to the metric space and then using a function such as the weighted Euclidean distance since the number of CSS peaks may differ between images. Instead, the set of peaks for the two images being compared are obtained, then a matching algorithm is performed to find the differences between the set of peaks. For more details please refer to [7]. We mention CSS as an example of a feature that cannot be mapped to the metric space because it is part of the MPEG-7 standard [11].

3. Proposed Feature Re-weighting Approach

In this section, the proposed relevance feedback mechanism for improving image retrieval is presented. But first, the distance model will be described to provide insight into how the images are evaluated to determine their similarity to the query image.

3.1 Distance Model

Figure 4 illustrates the model used to determine the distance value between a query object and target object. *d(query, DBimage$_i$)* represents the distance function for a given feature.

Each of these distances is associated with a corresponding weight. As mentioned before, the better the feature is at identifying the visual similarity between images, the higher the weight. These features weights are normalized to sum to one. The distances calculated for each feature are multiplied by their associated weight, and then they are all summed to obtain the overall distance value. In effect, the distance between the query image and an image in the database is defined as

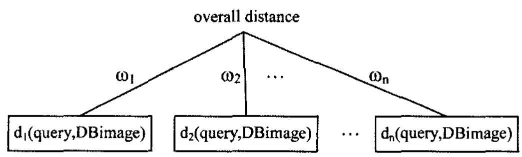


Figure 4 Distance Model

$$\text{overall distance} = \sum_{i=1}^{n} \omega_i \cdot d_i(query, DBimage)$$

where n is the number of features and ω_i and $d_i(query,\ DBimage)$ are the feature weight and distance function for feature i.

Since the distance values for different feature representations are not of comparable magnitudes, the distances, with respect to each feature, are divided by the maximum distance calculated between the query image and the images in the database. As a result, the distance values will be in the range [0, 1] and the overall distance will be in the range of [0, n], where n is the number of features.

The benefit of this distance model is the freedom it provides in how distances are calculated; there is no restriction on how the distances for each feature are calculated. As a result, there is freedom in the data structures used to represent the features and their associated similarity measures. Features such as CSS, 9D-SPA, and high-level features incorporating spatial relations are all supported.

3.2 Feature Re-weighting via Two-sided Confidence Intervals

The sets of relevant images and non-relevant images obtained from user feedback provide the information to update the weights of features by means of two-sided confidence intervals [12]. Confidence intervals are used to approximate the difference between two population proportions, $p(A)$ and $p(B)$, based on two independent samples of size n and m with sample proportions $p'(A)$ and $p'(B)$. The bounds of a two-sided confidence are calculated as follows:

Equation 1: $p(A)-p(B) \in (lb, ub)$

Equation 2:

$$lb = p'(A)-p'(B)-z_{\alpha/2}\sqrt{\frac{p'(A)\cdot(1-p'(A))}{n}+\frac{p'(B)\cdot(1-p'(B))}{m}}$$

Equation 3:

$$ub = p'(A)-p'(B)+z_{\alpha/2}\sqrt{\frac{p'(A)\cdot(1-p'(A))}{n}+\frac{p'(B)\cdot(1-p'(B))}{m}}$$

In Equations 2 and 3, $z\alpha/2$ is the confidence coefficient that is dependent on the desired confidence. The choice of the confidence coefficient is arbitrary.

The proposed re-weighting technique is based on the use of two-sided confidence intervals to approximate the difference between the feature distances for the sets of *relevant* and *non-relevant* images.

The user provides a query image for which the system must retrieve the top-k similar images in the database. However, the images considered most similar with regards to the calculated distances may not respect the user's perspective of visual similarity. With respect to each feature i individually, it is not possible to determine the average distance for relevant images for all the images in the database, which we will denote as $d_i(rel)$, and the same for non-relevant images, which we will denote as $d_i(non\text{-}rel)$, since the user cannot be expected to check every image contained in the database. By means of user feedback on the returned images, the system can calculate the average feature i distance for relevant and non-relevant images in the returned set, which will be represented by $d'_i(rel)$ and $d'_i(non\text{-}rel)$. Using the values $d'_i(rel)$ and $d'_i(non\text{-}rel)$, the range for the difference between $d_i(rel)$ and $d_i(non\text{-}rel)$ can be approximated using two-sided confidence intervals.

The bounds of a two-sided confidence for feature re-weighting are calculated as follows:

Equation 4: $d_i(rel)-d_i(non-rel) \in (lb, ub)$

Equation 5: $lb = d'_i(rel)-d'_i(non-rel)-$

$$z_{\alpha/2}\sqrt{\frac{d'_i(rel)\cdot(1-d'_i(rel))}{n}+\frac{d'_i(non-rel)\cdot(1-d'_i(non-rel))}{m}}$$

Equation 6: $ub = d'_i(rel)-d'_i(non-rel)+$

$$z_{\alpha/2}\sqrt{\frac{d'_i(rel)\cdot(1-d'_i(rel))}{n}+\frac{d'_i(non-rel)\cdot(1-d'_i(non-rel))}{m}}$$

As mentioned previously, $z\alpha/2$ is the confidence coefficient corresponding to the confidence interval desired. n is the number of images marked relevant and m is the number of images considered non-relevant (i.e. m=k-n). Again, for the difference between the average feature distance for relevant images and non-relevant images, the confidence interval must lie somewhere in the range [-1, 1] since the distances have been normalized to a maximum value of 1.

Confidence intervals are calculated for each feature. The location of the confidence interval determines how the feature will be re-weighted. Observing the location of the upper and lower bounds, one of the following cases will arise:

- If both the upper bound and lower bound are greater than 0, then the approximate average feature distance for all relevant objects in the database, for the feature under consideration, is greater than that for the non-relevant images. As a result, one can infer that this feature does not appropriately capture visual similarities for this query and its weight is set to 0.
- If the bounds straddle 0, that is, if the upper bound is positive and the lower bound is negative, one can infer that the feature is somewhat good, but cannot fully distinguish relevant images from those that are non-relevant. In this case, the feature weight is set to the ratio of the length of the negative portion of the interval and total length of the interval. The further the confidence interval slides into the negative range, the better the feature must be at distinguishing relevant images from non-relevant images.
- If both the upper bound and lower bound are less than 0, than the approximate average feature distance for all relevant images in the database is smaller that that for the non-relevant images. Thus, one can infer this feature successfully distinguishes visual similarities and the feature weight is determined using Equation 7. As in the previous case, the further negative the confidence interval is, the better the feature can distinguish the visual similarity. Thus, the upper bound is placed in the numerator to reflect this characteristic. Likewise, the closer the lower bound approaches -1, it the better the feature. Thus, $|lb|$ is placed in the denominator. Finally, the boundary condition where the upper bound is 0 must be considered. The feature weight must be equal to that when the upper bound is 0 for the previous case. In the previous case, when the upper bound is 0, the feature weight is set to 1. Thus, we add the constant 1.

Equation 7: feature weight = $1+\frac{|ub|}{1-|lb|}$

Having obtained the updated weight for each feature, the feature weights are normalized by total weight.

From the first iteration of retrieval, the system has only the single set of feedback to update the weights. When the feature weights are updated, the next iteration of retrieval may return a different set of images. However, for the following iterations, the information from the images returned in the previous iterations can be used as well. The following is a description of the notation that will be used to describe the images that are used to obtain the distance information for feature re- weighting.

k : the number of iterations of retrieval

R_k : the set of images returned in the k^{th} iteration

R_k^{rel} : the relevant images from the k^{th} iteration

$R_k^{non-rel}$: the non-relevant images from the k^{th} iteration

R_k^{unique} : the set of unique images that have been returned up to the k^{th} iteration

$R_k^{unique,rel}$: the set of unique images identified as relevant in the k iterations

$R_k^{unique,non-rel}$: the set of unique images identified as non-relevant in the k iterations

Take note : $R_k = R_k^{rel} \cup R_k^{non-rel}$,
$R_k^{unique} = R_k^{unique,rel} \cup R_k^{unique,non-rel}$,
$R_1^{unique} = R_1$,
$R_1^{unique,rel} = R_1^{rel}$, and
$R_1^{unique,non-rel} = R_1^{non-rel}$

Then for iterations k+1, where k > 0, R_{k+1}^{unique} and $R_{k+1}^{unique,non-rel}$ are defined as follows.

$R_{k+1}^{unique} = R_{k+1} \cup R_k^{unique}$

$R_{k+1}^{unique,rel} = R_{k+1}^{rel} \cup R_k^{unique,rel}$

$R_{k+1}^{unique,non-rel} = R_{k+1}^{non-rel} \cup R_k^{unique,non-rel}$

Thus, to incorporate the information obtained from the previous retrieval iterations for feature re-weighting, the images in $R_{k+1}^{unique,rel}$ and $R_{k+1}^{unique,non-rel}$ are used to calculate the confidence intervals at the $(k+1)^{th}$ iteration. More specifically, when using Equations 5 and 6 to update the weights at the $(k+1)^{th}$ iteration, $d'_i(rel)$ and $d'_i(non-rel)$ are now obtained using the feature distances for the images in $R_{k+1}^{unique,rel}$ and $R_{k+1}^{unique,non-rel}$, respectively. Also, n now corresponds to the number of images in $R_{k+1}^{unique,rel}$ and m corresponds to the number of images in $R_{k+1}^{unique,non-rel}$.

4. Experimentation

In this section, the experimental results are presented to demonstrate the effectiveness of the proposed feature re-weighting technique. In addition, the retrieval performance of our approach is compared to that of MARS and [3] (DD) since their inter-feature re-weighting techniques can be applied to features that cannot be mapped into the metric space.

4.1 Experimental Environment & Prototype System

The experimentation was performed on the Windows platform powered by a Pentium4 2.6GHz CPU using 512MB of RAM. The prototype system is implemented using C++ and the .NET framework. Images and their associated feature data are stored to an Oracle 10g database located remotely. The features used to describe the images include Polar Projections [13], CSS [7], eccentricity, compactness, perimeter, and circularity. A screenshot of the experimental system is provided in Figure 5. The query image is displayed in the top left corner of the window. The images below are the top-k images retrieved from the database. The images are ranked based on distance, where the top-left image has the smallest distance and increase from left to right, top to bottom. The user identifies the relevant images among the result images by clicking on them, and then clicks the button to the right of the query image to update the feature weights and obtain the next set of results.

4.2 Experimental Results

In each of the experimental comparisons, any images not identified by the user as being relevant are automatically excluded from being candidates for retrieval in the following iterations. This benefit of this approach is intuitive as there is no reason for the user to see a non-relevant image more than once. The image database contains 8400 images obtained from a modification of the MPEG-7 Shape Silhouette image set [14]. The results presented are averaged over 30 queries. For each query image, there are 120 relevant images in the dataset, thus we retrieve the top 120 images per iteration.

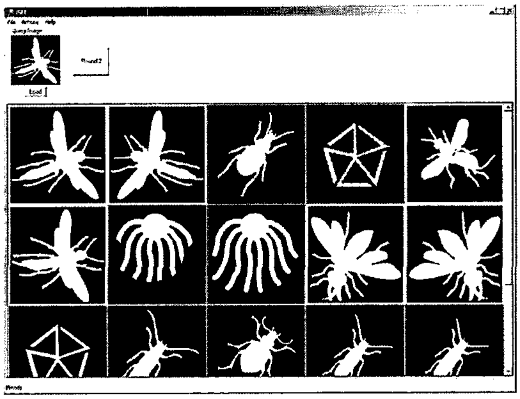


Figure 5 Screenshot of Experimental System

The measure used to present the retrieval accuracy in the experimental results is recall and is defined as follows:

$$recall = \frac{relevant_{retrieved}}{relevant_{total}}$$

where $relevant_{retrieved}$ represents the number of relevant images retrieved for the given iteration and $relevant_{total}$ represents the number of images in the database that are relevant to the query.

Table 1 shows a comparison of the retrieval recall performance using the confidence coefficients corresponding to 60%, 70%, 80%, 90%, 95%, 99% and 99.9% *two-sided confidence intervals*. The boldface values identify the best retrieval for the given iteration.

From the results, one can see that as the number of iterations increases, the higher confidence coefficients provide better recall. As 95% is the most commonly used percentage for two-sided confidence intervals, such confidence intervals will be used for the remaining experimental results.

The precision-recall graph of Figure 6 shows the effectiveness of our proposed feature re-weighting

Table 1 Retrieval Comparison of Confidence Interval Percentages

	60%	70%	80%	90%	95%	99%	99.9%
Iteration 5	0.5978	**0.5992**	0.5922	0.5975	0.5975	0.5969	0.5972
Iteration 10	0.6981	0.7050	0.6931	**0.7147**	0.7119	0.7117	0.7136
Iteration 20	0.8050	0.8139	0.8042	0.8222	0.8228	**0.8239**	**0.8239**

technique. Notice that as the feedback iterations increase, the retrieval performance continues to improve.

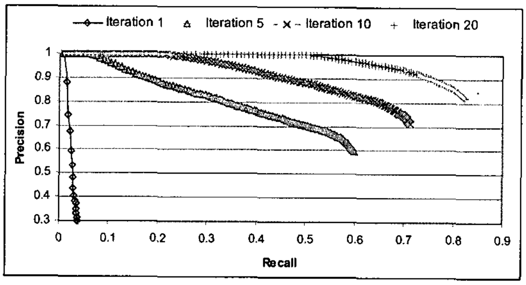


Figure 6 Precision-Recall Graph

To show the effectiveness of our proposed approach to that of MARS and the approach of DD, we provide a quantitative performance comparison. We set the scoring scheme of MARS' relevance feedback technique so that relevant images are assigned a score of 1 and non-relevant images are assigned a score of 0. Figure 7 compares the final recall of each approach after retrieving the 120 images for each iteration. Each approach starts with the same recall as the feature re-weighting is performed starting from the first set of user feedback. Note, that since CSS cannot be represented in the metric space, the intra-feature weighting approach of MARS and DD cannot be applied. The eccentricity, compactness, perimeter, and circularity representations are each comprised of just a single component, which further invalidates the use of intra-feature re-weighting. In addition, because the values for the components of the Polar Projection representation are not always of comparable magnitudes, a lower variation for a component's value between images does not necessarily indicate higher importance of the component. For these reasons, intra-feature re-weighting approaches are not applied. In Figure 7, note how the proposed approach shows a significant improvement between the first and second iteration. This is desirable as the user would like to obtain optimal results as quickly as possible. The lower performance of MARS and DD can be attributed to the fact that only their inter-feature re-weighting is applicable.

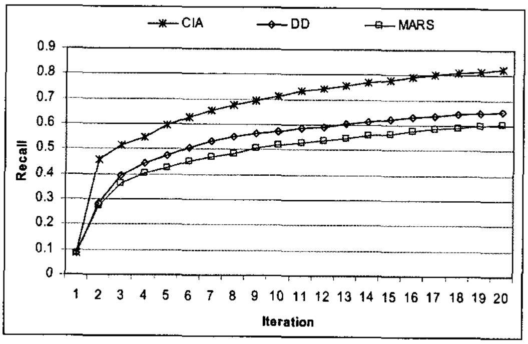


Figure 7 Comparison of Retrieval Effectiveness

Table 2 provides more detail of the recall performance of the three approaches. The values in the *Confidence Intervals, DD,* and *MARS* columns correspond to the improvement in recall compared to the first iteration of retrieval. The *Improvement*$_X$ column, where X is MARS or DD, displays the improvement of the proposed approach over MARS and DD as calculated using:

$$\frac{improvement_{ConfidenceIntervals} - improvement_X}{improvement_X}$$

5. Conclusion

In this paper, we proposed a feature re-weighting technique for content-based image retrieval systems that incorporate relevance feedback. Unlike existing approaches, feature representations are not required to have a fixed number of components for each image. Instead, the feature distances are used with the statistical technique of two-sided confidence intervals to update the feature weights. As a result,

Table 2 Improvement in Recall

	Confidence Intervals	DD	MARS	ImprovementDD	ImprovementMARS
Iteration 2	0.3708	0.1947	0.1869	0.9045	0.9839
Iteration 5	0.5108	0.3889	0.3403	0.3134	0.5010
Iteration 10	0.6253	0.4867	0.4317	0.2848	0.4485
Iteration 20	0.7361	0.5664	0.5175	0.2996	0.4224

this allows for the approach to support features irrespective of their structure. This is of great benefit as existing approaches cannot handle features that cannot be mapped to metric spaces.

Another advantage of the proposed approach is the simplicity of the user feedback. More specifically, in the approach of MARS, obtaining multi-class (relevant and non-relevant) feedback is dependent on the scoring scheme used. In addition, more responsibility is placed on the user in MARS since they must judge the degree of relevance for each result. In the proposed approach, the user simply identifies the relevant images to provide feedback.

Our experimental results show that our feature re-weighting approach provides effective feature re-weighting regardless of the structure of the feature representations. One desirable characteristic that is evident from the results is that the proposed approach provides a significant improvement in the early iterations of retrieval, which is evident with the 98% improvement over MARS for the 2^{nd} iteration of retrieval.

References

[1] Rui, Y., Huang, T. S., Ortega, M., and Mehrotra, S., "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," IEEE Transactions on Circuits and Video Technology, Vol. 8, No. 5, pp. 644-655, 1998.

[2] Porkaew, K., Mehrotra, S. and Ortega, M., "Query Reformulation for Content Based Multimedia Retrieval in MARS," Technical Report TR-DB-99-03, University of California at Irvine, 1999.

[3] Wu, Y., and Zhang, A., "A Feature Re-weighting Approach for Relevance Feedback in Image Retrieval," Proceedings of the IEEE 2002 International Conference on Image Processing, pp. 581-584, 2002.

[4] Aggarwal, G., Ashwin, T. V., and Ghosal, S., "An Image Retrieval System with Automatic Query Modification," IEEE Transactions on Multimedia, Vol. 4, No. 2, pp. 201-214, 2002.

[5] Ishikawa, Y., Subramanya, R., and Faloutsos, C., "MindReader: Querying Databases Through Multiple Examples," Proceedings of the 24th International Conference on Very Large Data Bases, pp. 218-227, 1998.

[6] Kherfi, M.L., Ziou, D., and Bernardi, A., "Combining Positive and Negative Examples in Relevance Feedback for Content-Based Image Retrieval," Journal of Visual Communication and Image Representation, Vol. 14, No. 4, pp. 428-457, 2003.

[7] Mokhtarian, F., Abbasi, S., and Kittler, J., "Robust and Efficient Shape Indexing through Curvature Scale Space," Proceedings of the British Machine Vision Conference, pp. 53-62, 1996.

[8] Huang, P.W. and Lee, C.H., "Image Database Design Based on 9D-SPA Representation for Spatial Relations," IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 12, pp. 1486-1496, 2004.

[9] Faloutsos, C., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., and Equitz, W., "Efficient and Effective Querying by Image Content," Journal of Intelligent Information Systems, Vol. 3, No 3/4, pp 133-150, 1994.

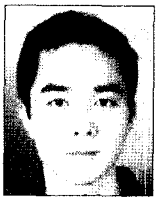
[10] Smith, J. R. and Chang, S. F., "VisualSEEk: A Fully Automated Content-Based Image Query System," Proceedings of the 4th ACM International Conference on Multimedia, pp. 87-98, 1997.

[11] Manjunath, B.S., Salembier, P., and Sikora, T., "Introduction to MPEG-7: Multimedia Content Description Interface," Wiley, 2002.

[12] Hayter, A. J., "Probability and Statistics for Engineers and Scientists," Duxbury Press, 2002.

[13] Kwon, Y. I., Park, H. H., Lee, S. L., and Chung, C. W., "A Shape Feature Extraction for Complex Topographical Images," Proceedings of the International Symposium on Remote Sensing, pp. 575-578, 2005.

[14] Latecki, L. J., "Shape Data for the MPEG-7 Core Experiment CE-Shape-1," http://www.cis.temple.edu/~latecki/TestData/mpeg7shapeB.tar.gz, 2002.



Robert S. Lee

2002년 University of British Columbia, Computer Science (BS). 2006년 한국과학기술원 전자전산학과 전산학전공(MS) 2006년~현재 Business Objects (Software Developer). 관심분야는 멀티미디어 데이타베이스

김 상 희

1985년 서강대학교 전자계산학과(공학사) 1987년 서강대학교 전자계산학과(공학석사). 2004년 한국과학기술원 전산학과(공학박사). 1994년~1995년 미국 Hughes Training Inc.에서 항공기 시뮬레이터 개발(교환 연구원). 1987년~현재 국방과학연구소 책임연구원. 관심분야는 실시간 3차원 렌더링, 가상현실, 영상데이타베이스, GIS, 컴퓨터 그래픽스

박 호 현

1987년 2월 서울대학교 계산통계학과 학사. 1995년 8월 한국과학기술원 컴퓨터공학과 석사. 2001년 2월 한국과학기술원 전산학과 박사. 1987년 1월~2003년 2월 삼성전자 수석연구원. 2003년 3월~현재 중앙대학교 조교수. 관심분야는 멀티미디어 스트리밍, 멀티미디어 정보검색, 시공간 데이타베이스

이 석 룡

1984년 연세대학교 기계공학과 학사. 1993년 연세대학교 산업공학과 전자계산전공 석사. 2001년 한국과학기술원 정보및통신공학과 컴퓨터공학전공 박사. 1984년~1995년 한국IBM 소프트웨어 연구소 선임연구원. 1995년~2001년 안산1대학 조교수. 2002년~현재 한국외국어대학교 산업정보시스템공학부 부교수. 관심분야는 멀티미디어 데이타베이스, 데이타마이닝, 정보검색 등

정 진 완

1973년 서울대학교 공과대학 전기공학과(학사). 1983년 University of Michigan 컴퓨터공학과(박사). 1983년~1993년 미국 GM 연구소 전산과학과 선임연구원 및 책임연구원. 1993년~현재 한국과학기술원 전산학전공 부교수 및 교수. 관심분야는 XML, 시맨틱웹, 멀티미디어 데이타베이스, 스트림 데이타 및 센서 네트워크 데이타베이스