A Weighted Distance Approach to Relevance Feedback

Selim Aksoy, Robert M. Haralick
Intelligent Systems Laboratory
University of Washington
Seattle, WA 98195-2500 USA

Faouzi A. Cheikh, Moncef Gabbouj
Signal Processing Laboratory
Tampere University of Technology
P.O. Box 553, FIN-33101 Tampere, Finland

Abstract

Content-based image retrieval systems use low-level features like color and texture for image representation. Given these representations as feature vectors, similarity between images is measured by computing distances in the feature space. Unfortunately, these low-level features cannot always capture the high-level concept of similarity in human perception. Relevance feedback tries to improve the performance by allowing iterative retrievals where the feedback information from the user is incorporated into the database search. We present a weighted distance approach where the weights are the ratios of standard deviations of the feature values both for the whole database and also among the images selected as relevant by the user. The feedback is used for both independent and incremental updating of the weights and these weights are used to iteratively refine the effects of different features in the database search. Retrieval performance is evaluated using average precision *and* progress *that are computed on a database of approximately 10,000 images and an average performance improvement of 19% is obtained after the first iteration.*

1. Introduction

Image retrieval has received significant attention in recent years. Initial work on content-based retrieval focused on using low-level features like color and texture for image representation. After each image is associated with a feature vector, similarity between images is measured by computing distances between feature vectors in the feature space. It is generally assumed that the features are able to locate visually similar images close to each other in the feature space so that non-parametric approaches, like the k-nearest neighbor search, can be used for retrieval.

Unfortunately, none of the existing feature extraction algorithms can always map visually similar images to nearby locations. A common observation in retrieval results is that sometimes images that are quite irrelevant to the query image are also retrieved simply because they are close to it in the feature space. Therefore, post-processing methods [2] are usually required to improve the results.

The high-level concept of similarity and subjectivity in human perception cannot always be captured by the low-level features. To make the user have more control on the search criteria, some systems allowed the user to weight the features [9, 3]. This is not generally applicable because the ordinary user does not usually have a detailed understanding of the low-level features designed by an expert.

Recently, relevance feedback has been the most commonly applied post-processing technique. The main idea is to include the human user in the retrieval loop. An initial search is done in the database using the original query image input by the user. Upon being presented the results of this search, the user labels some of the these images as relevant and irrelevant according to his/her information needs. The goal is to incorporate this feedback information into the database search in terms of iterative retrievals.

A common approach for relevance feedback in image databases has been to use the ideas that were developed in the information retrieval literature [12]. A commonly used technique has been the vector space model where a new query feature vector is generated as a weighted linear combination of the original feature vector and the feature vectors of the images that were labeled as relevant or irrelevant by the user [12, 6]. Other approaches include using keywords [8], creating a probabilistic user model [7], modifying the distance measure [4], reorganizing the retrieval results [5], feature density estimation [10], and iterative weight updating [13, 11].

Retrieval algorithms depend on features directly computed from images. We want to use only the information fed back by the user instead of using artificial keywords or heuristic assumptions. We also cannot assume anything about the user's information need, neither can we assume any distributions for the relevancy and irrelevancy he/she is looking for. Therefore, we decided to weight the automatically computed features in the k-nearest neighbor search according to user's responses. The ratios of standard deviations of the feature values both for the whole database and also among the images selected as relevant by the user are used in both independent and incremental updating of the weights.

Experiments were done on a database of approximately 10,000 images and the retrieval performance was evaluated using average *precision* computed for a manually

groundtruthed data set. We also defined a new measure called *progress* to measure the performance. The following section describes the motivation and details of our weighted distance approach. Section 3 presents the experiments and discusses the results. Conclusions are given in Section 4.

2. The Weighted Distance Approach
2.1. Definitions

First, we present some definitions that will be used in the following sections.

K: Number of iterative searches.

Q: Number of features in a feature vector.

R^k= {retrieval set after the k'th search}, $k = 0, \ldots, K$, while R^0 being the whole database.

R_{rel}^k= {set of images in R^k that are marked as relevant}.

F_j^k= {values of the j'th feature components of the images in R^k}.

$F_{\text{rel},j}^k$= {values of the j'th feature components of the images in R_{rel}^k}.

In our retrieval scenario, similarity between images is measured by computing distances between feature vectors in the feature space. Given two feature vectors x and y and the weight vector w, we use the weighted L_1 distance

$$\rho(x, y; w) = \sum_{j=1}^{Q} |w_j(x_j - y_j)| \tag{1}$$

and the weighted L_2 distance

$$\rho(x, y; w) = \left(\sum_{j=1}^{Q} |w_j(x_j - y_j)|^2 \right)^{1/2}. \tag{2}$$

2.2. Motivation

From the pattern recognition point of view, for a feature to be good, its variance among all the images in the database should be large but its variance among the relevant images should be small. Any one of these is not enough alone but characterizes a good feature when combined with the other.

Given these observations, we decided to use $w_j^k = \sigma_j^0/\sigma_{\text{rel},j}^k$, where $\sigma_j^0 = \text{std}(F_j^0)$ and $\sigma_{\text{rel},j}^k = \text{std}(F_{\text{rel},j}^k)$, as the weight for the j'th feature in the $k + 1$'st iteration. For a given image, there is a small set of relevant images in the database; on the other hand, the rest of the images can be categorized as irrelevant. We preferred using only the relevant images because the small set of feedback images that are selected by the user for both relevancy and irrelevancy will probably provide a better estimate for the former case.

Depending on σ_j^0 and $\sigma_{\text{rel},j}^k$, four different situations can arise as shown in Table 1:

Table 1: Motivation for the weight selection. Moving upwards in the table represents a situation that is closer to ideal.

σ_j^0	$\sigma_{\text{rel},j}^k$	$w_j^k = \sigma_j^0/\sigma_{\text{rel},j}^k$
large	small	large
large	large	~ 1
small	small	~ 1
small	large	small

- When σ_j^0 is large and $\sigma_{\text{rel},j}^k$ is small, w_j^k becomes large. This means that the feature has a diverse set of values in the database but its values for relevant images are similar. This is a desired situation and shows that this feature is very effective in distinguishing this specific relevant image set so a large weight assigns more importance to this feature.
- When both σ_j^0 and $\sigma_{\text{rel},j}^k$ are large, w_j^k is close to 1. This means that the feature may have good discrimination characteristics in the database but is not effective for this specific relevant image group. The resulting weight does not give any particular importance to this feature.
- When both σ_j^0 and $\sigma_{\text{rel},j}^k$ are small, w_j^k is again close to 1. This is a similar but slightly worse situation than the previous one. The feature is not generally effective in the database and is not effective for this relevant set either. No importance is given to this feature.
- When σ_j^0 is small and $\sigma_{\text{rel},j}^k$ is large, w_j^k becomes small. This is the worst case among all the possibilities. The feature is not generally effective and even causes the distance between relevant images to increase. A small weight forces the distance measure to ignore the effect of this feature.

All of the resulting weights in these four cases are consistent with the desired situations in an ideal retrieval.

2.3. Iterative retrieval

The retrieval algorithm can be described as follows:

1. Initialize all weights uniformly as $w_j^0 = 1/Q$, $j = 1, \ldots, Q$. Compute σ_j^0, $j = 1, \ldots, Q$.
2. For $k = 1, \ldots, K$,

   (a) Search the database using w_j^{k-1} and obtain R^k.

   (b) Get feedback from the user as R_{rel}^k.

   (c) Compute $\sigma_{\text{rel},j}^k$, $j = 1, \ldots, Q$.

   (d) Compute

$$w_j^k = \frac{\sigma_j^0}{\sigma_{\text{rel},j}^k}, j = 1, \ldots, Q \tag{3}$$

and normalize as $w_j^k = w_j^k / \sum_{j=1}^{Q} w_j^k$.

3. Do the final search using $w_j^K, j = 1, \ldots, Q$.

To compute $\sigma_{\mathrm{rel},j}^k$ in 2c, we use two methods:

- Independent update: Standard deviations are estimated independently in every iteration using only that iteration's retrieval sets, i.e.

$$(\sigma_{\mathrm{rel},j}^k)^2 = E((F_{\mathrm{rel},j}^k)^2) - E(F_{\mathrm{rel},j}^k)^2, j = 1, \ldots, Q$$

where $E(F_{\mathrm{rel},j}^k)$ and $E((F_{\mathrm{rel},j}^k)^2)$ are the first and second moments of the sample $F_{\mathrm{rel},j}^k$ respectively.

- Incremental update: We assume that user's notion of similarity does not change as the iterations progress and he/she is consistent in consecutive iterations. Therefore, standard deviations are incrementally updated in every iteration, i.e.

$$(\sigma_{\mathrm{rel},j}^k)^2 = \left(\frac{|R_{\mathrm{rel}}^{k-1}|E((F_{\mathrm{rel},j}^{k-1})^2) + |R_{\mathrm{rel}}^k|E((F_{\mathrm{rel},j}^k)^2)}{|R_{\mathrm{rel}}^{k-1}| + |R_{\mathrm{rel}}^k|} \right) - \left(\frac{|R_{\mathrm{rel}}^{k-1}|E(F_{\mathrm{rel},j}^{k-1}) + |R_{\mathrm{rel}}^k|E(F_{\mathrm{rel},j}^k)}{|R_{\mathrm{rel}}^{k-1}| + |R_{\mathrm{rel}}^k|} \right)^2,$$

j=1,...,Q, where the retrieval sets are updated as $R^k = R^k \cup R^{k-1}$ and $R_{\mathrm{rel}}^k = R_{\mathrm{rel}}^k \cup R_{\mathrm{rel}}^{k-1}$ after every iteration.

When all the values in $F_{\mathrm{rel},j}^k$ are the same, i.e. all images have the same value for that feature, we assign a large constant value to w_j^k.

3. Experiments and Results
3.1. Database population

Our database contains 10,410 256×256 images that include aerial images (Fort Hood Data of the RADIUS Project) and remote sensing images (LANDSAT). For performance evaluation, we randomly selected 340 images and formed a groundtruth of 7 categories; parking lots, roads, residential areas, landscapes, LANDSAT USA, DMSP North Pole and LANDSAT Chernobyl. Textural features described in [1] were used for image representation. The first set of features are the line-angle-ratio statistics that use spatial relationships and properties of lines, and the second set of features are the variances of co-occurrence statistics of pixels in particular spatial relationships.

3.2. Retrieval performance

We measure the retrieval performance by *precision* which is defined as the percentage of retrieved images that are actually relevant. Retrieval results in terms of precision averaged over the groundtruth images are given in Figure 1. The search engine performs a new search in the database and retrieves 12 images in every iteration. Independent updating was used for both L_1 and L_2 distances. The results
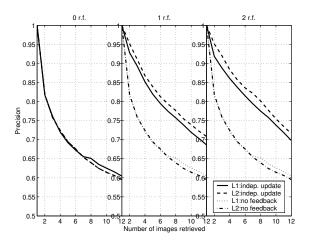

Figure 1: Average *precision* for the first two iterations.

Table 2: Average *precision* when 12 images are retrieved. Improvements for each iteration over the previous iteration are given in paranthesis.

Distance	0 rf	1 rf	2 rf
L_1	0.60	0.69 (13.53%)	0.70 (1.71%)
L_2	0.60	0.71 (19.03%)	0.72 (1.06%)

are summarized in Table 2. We can see that slightly better results could be obtained on the average when the L_2 distance was used. We tried up to five iterations and the largest average improvement was obtained as 19% after the first iteration. This is a desirable situation that shows a fast convergence.

When the whole database is searched in every iteration, the improvement is usually a few additional relevant images and this can also be achieved by showing a new set of images from the retrieval set of the original query instead of waiting for the computation of getting feedback and doing one more search. Another way of investigating how well the relevance feedback performs is to compare the performance of iterative retrieval with that of the original search in terms of the *progress* made towards retrieving a specific number of images. After obtaining the feedback, the search engine performs a new search in the database but ignores all the images that were retrieved in previous iterations. Therefore, a new set of 12 images are retrieved in every iteration. This performance is compared with the case where the next set of 12 images from the retrieval set of the original search are presented to the user (showing the next page in the user interface). Given n as a specific number of images retrieved, we define *progress* as the ratio of two precisions,

$$Progress = \frac{|\text{relevant images among } n \text{ after } \lceil n/12 \rceil \text{ iterations}|}{|\text{relevant images among } n \text{ retrieved without feedback}|}. \tag{4}$$
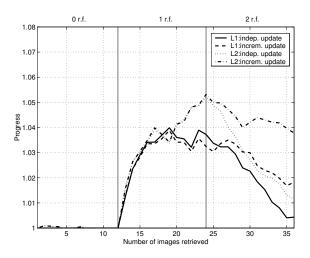
Figure 2: Average *progress* for the first two iterations.

Table 3: Average *progress* when 12, 24 and 36 images are retrieved.

Distance	0 rf	1 rf	2 rf
L_1:Indep. update	1	1.037	1.004
L_1:Increm. update	1	1.033	1.018
L_2:Indep. update	1	1.052	1.012
L_2:Increm. update	1	1.053	1.038

When progress is greater than 1, it means the feedback algorithm is effective and converges faster. Average progress is given in Figure 2. The results are summarized in Table 3. We can see that incremental updating performed better than independent updating. The largest improvement was obtained as 5.3% greater progress over the no feedback case.

4. Conclusions and Future Work

We presented a weighted distance approach where the weights were the ratios of standard deviations of the feature values both for the whole database and also among the images selected as relevant by the user. We discussed the effects of independent weight updating where the weights were estimated independently in every iteration using only the feedback information in that iteration, and incremental weight updating where the weights were incrementally updated in every iteration using the feedback information in that iteration as well as the previous iterations.

Retrieval performance was evaluated in terms of average *precision* and *progress* for a groundtruth database and an average performance improvement of 19% was obtained after the first iteration, which shows a fast convergence.

One issue to be addressed further is to investigate other functions of the standard deviations as weight updating methods. The results may also be improved if the feedback for irrelevant images is also incorporated into the database search.

References

[1] S. Aksoy and R. M. Haralick. Textural features for image database retrieval. In *Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries, in conjunction with CVPR'98*, pages 45–49, Santa Barbara, CA, June 1998.

[2] S. Aksoy and R. M. Haralick. Graph–theoretic clustering for image grouping and retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 63–68, Colorado, June 1999.

[3] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C.-F. Shu. The virage search engine: An open framework for image management. In *SPIE Storage and Retrieval of Image and Video Databases*, February 1996.

[4] J.-Y. Chen, C. Bouman, and J. Dalton. Active browsing using similarity pyramids. In *Proceedings of the VII IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases*, volume 3656, pages 144–154, San Jose, CA,, January 1999.

[5] J.-Y. Chen, C. Taskiran, E. Delp, and C. Bouman. Vibe: A new paradigm for video database browsing and search. In *Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 96 –100, 1998.

[6] T.-S. Chua, W.-C. Low, and C.-X. Chu. Relevance feedback techniques for color-based image retrieval. In *Proceedings of Multimedia Modeling*, pages 24–31, 1998.

[7] I. Cox, M. Miller, S. Omohundro, and P. Yianilos. Pichunter: Bayesian relevance feedback for image retrieval. In *Proceedings of the 13th IAPR International Conference on Pattern Recognition*, volume 3, pages 361–369, 1996.

[8] I. Cox, T. Papathomas, J. Ghosn, P. Yianilos, and M. Miller. Hidden annotation in content based image retrieval. In *Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 76–81, 1997.

[9] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. The QBIC project: Querying images by content using color, texture and shape. In *SPIE Storage and Retrieval of Image and Video Databases*, pages 173–181, 1993.

[10] C. Nastar, M. Mitschke, and C. Meilhac. Efficient query refinement for image retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 547–552, June 1998.

[11] J. Peng, B. Bhanu, and S. Qing. Probabilistic feature relevance learning for content-based image retrieval. *Computer Vision and Image Understanding, Special Issue on Content-Vased Access of Image and Video Libraries*, 75(1/2):150–164, July/August 1999.

[12] Y. Rui, T. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in MARS. In *Proceedings of IEEE International Conference on Image Processing*, 1997.

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