Feature Re-weighting in Content-Based Image Retrieval

Gita Das[1], Sid Ray[1], and Campbell Wilson[2]

[1] Clayton School of Information Technology
Monash University
Victoria 3800, Australia
{Gita.Das, Sid.Ray}@csse.monash.edu.au
[2] Caulfield School of Information Technology
Monash University
Victoria 3800, Australia
Campbell.Wilson@csse.monash.edu.au

Abstract. Relevance Feedback (RF) is a useful technique in reducing semantic gap which is a bottleneck in Content-Based Image Retrieval (CBIR). One of the classical approaches to implement RF is feature re-weighting where weights in the similarity measure are modified using feedback samples as returned by the user. The main issues in RF are learning the system parameters from feedback samples and the high-dimensional feature space. We addressed the second problem in our previous work, here, we focus on the first problem. In this paper, we investigated different weight update schemes and compared the retrieval results. We proposed a new feature re-weighting method which we tested on three different image databases of size varying between 2000 and 8365, and having number of categories between 10 and 98. The experimental results with scope values of 20 and 100 demonstrated the superiority of our method in terms of retrieval accuracy.

1 Introduction

The selection of features e.g. colour, shape, colour-layout etc. and their proper representation e.g. colour histogram, statistical moments etc. are very important for good system retrieval. However, the low level features (e.g. colour, shape) used to represent an image do not necessarily represent the high level semantics and human perception of that image. A solution towards this problem is human intervention in terms of Relevance Feedback [1], [2], [3]. For a given query, the system first retrieves a set of ranked images according to a similarity metric, which represents the distance between the feature vectors of the query image and the database images. Then the user is asked to select the images that are relevant or irrelevant (or non-relevant) to his/her query. The system extracts information from these samples and uses that information to improve retrieval results. A revised ranked list of images is then presented to the user. This process continues until there is no further improvement in the result or the user is satisfied with the result. In classical approach, there are mainly two ways to implement RF

H. Sundaram et al. (Eds.): CIVR 2006, LNCS 4071, pp. 193–200, 2006.

namely, query updating and feature re-weighting. In query updating method, the components of the query vector are updated using the average of component values of all relevant samples so that the new query point moves towards the centre of relevant class. In feature re-weighting, the weight factors in the similarity measure are modified using relevant samples. The essence of feature re-weighting is to put more weights on the feature components that discriminate well between relevant and non-relevant images and thus enhances retrieval and to put less weights for the ones that do not help retrieval. Feature re-weighting is found to be very suitable for large size databases and high dimensional feature space [8]. Also, this method is simple to implement and produces fairly good retrieval. However, in order to improve retrieval accuracy we need to use the feedback samples carefully and intelligently. In MARS system [4], they used the inverse of standard deviation of the feature component values for the relevant samples. Most of the work reported in the literature used only relevant samples [5], [6], [7]. In [8], Wu and Zhang proposed a discriminant factor that determines the ability of a feature component in separating relevant images from the irrelevant ones. They showed improvement over the MARS system which used only relevant images. Inspired by their work, we propose a modified weight factor that demonstrated significant improvement over the method in [8]. Both of the query update and the feature re-weighting approaches are based on vector model which originally were used in text retrieval [1]. We used a combination of both methods. We experimented with several weight updating schemes to re-shape the similarity measure and compared the retrieval results.

Sections 2 and 3 describe the proposed approach and experimental results respectively. Section 4 contains conclusions and future directions.

2 Methodology

For rest of the paper, we used the following nomenclature:

N: Number of images in the database
C: Number of semantic categories in the database
N_r: Scope i.e. the number of top retrieved images returned to the user
Q, I: Query image and Database image respectively
k: Number of iterations in RF
M: Number of components in the feature vector
w_i^k: weight factor for i^{th} feature component in k^{th} iteration.

2.1 Feature Representation

In [9], we proposed a compact feature representation based on the elements of Colour Co-occurrence Matrices (CCM) in Hue, Saturation, Value (H,S,V=16,3,3) colour space. We chose HSV colour model as it is known to be perceptually uniform. A Colour Co-occurrence Matrix represents how the spatial correlation of colour changes with distance i.e. pixel positions [10]. So, unlike colour histogram, colour co-occurrence matrix provides spatial information of the image.

We observed that diagonal elements of CCMs are much more in number (about 80%) compared to the non-diagonal elements (about 20%). This observation is in line with that reported in [11]. Also, we have noticed that most of the non-diagonal elements are zero. From the original 148-dimensional feature vector, we constructed a 25-dimensional feature vector with all diagonal elements and Sum-Average [12] of all non-diagonal elements from H,S,V matrices. An increase in feature dimension essentially means an exponential growth in the number of training samples. This limitation, called the Curse of Dimensionality, is a well known fact in Pattern Classification [13]. The reduction in dimension from original higher dimension reduced online computation time and enhanced retrieval accuracy. For more details, see our previous work reported in [9].

As different feature components have different ranges (or values), we normalized them so that they lie within [0,1] and each component contributes equally in the similarity measure. The i^{th} normalized feature component, $f_i^{'}$ is given by [9],

$$f_i^{'} = \frac{f_{i,org} - \mu_i}{3\sigma_i},\ i = 1, 2, ...M\ . \tag{1}$$

where $f_{i,org}$ is the original i^{th} feature component, μ_i is the mean and σ_i is the Standard Deviation (SD) of $f_{i,org}$. These values are calculated over the entire database of N samples. Under the assumption of Gaussian distribution of values, the term $3\sigma_i$ ensures that at least 99% of the samples are within the range $[-1, 1]$. Any value that is < -1 is set to -1 and > 1 is set to 1. In order to map the normalized values from $[-1, 1]$ to $[0, 1]$, we used the following formula:

$$f_i = \frac{f_i^{'} + 1}{2}\ . \tag{2}$$

2.2 Feature Re-weighting

In CBIR research, a number of distance measures have been used in the past in order to measure the similarity (or dissimilarity) between the query image and the database images. Each one of them has its own merits and demerits. Minkowski distance, of which Manhattan (or City-block) and Euclidean distances are special cases, is probably the most widely used. We chose Manhattan distance because it is computationally very simple and produces fairly good results. Also, as our main focus of this article is on the RF strategy, whatever strength or weakness the similarity measure has got, we assume that it will affect the retrieval of different data sets more or less the same way.

The similarity between I and Q is given by the following weighted Minkowski distance measure:

$$D(I, Q) = \sum_{i=1}^{M} w_i * |f_{iI} - f_{iQ}|\ . \tag{3}$$

where f_{iI}, f_{iQ} are i^{th} feature component of I and of Q respectively and w_i is weight factor. When there is no RF, equal weight values are used for each feature

component. With RF, these weights are updated using feedback samples. First, we used the following weight value:

$$weight - type1 : w_i^{k+1} = \frac{\epsilon + \sigma_{N_r,i}^k}{\epsilon + \sigma_{rel,i}^k}, \epsilon = 0.0001 \ . \tag{4}$$

Here, $\sigma_{N_r,i}^k$ is standard deviation over the N_r retrieved images and $\sigma_{rel,i}^k$ is the standard deviation over the relevant images in k^{th} iteration. If a feature component has smaller variation over the relevant samples then it should get higher weight as this represents the relevant samples better [4] in the feature space. Similar weight factor is used in [5] and [6], however, there the numerator represented standard deviation over the entire database. In the numerator of eqn (4), we used standard deviation over N_r as the variation over the entire database remains unchanged with iteration and thus does not provide any extra information. However, with each iteration a new set of images is likely to be retrieved and a new $\sigma_{N_r,i}^k$ obtained. A small value of ϵ is used to avoid computational problem of $\sigma_{rel,i}^k$ being zero when no similar image (other than the query itself is retrieved) is retrieved. The value of ϵ is chosen to be 0.0001 so that it does not affect the weight values significantly.

In[8], Wu and Zhang used both relevant and non-relevant images to update weights. They used a discriminant ratio to determine the ability of a feature component in separating relevant images from the non-relevant ones:

$$\delta_i^k = 1 - \frac{\sum_{l=1}^k |\psi_i^{l,U}|}{\sum_{l=1}^k |F_i^{l,U}|} \ . \tag{5}$$

where $\sum_{l=1}^k |\psi_i^{l,U}|$ is the no. of non-relevant images located inside the dominant range of relevant samples and $\sum_{l=1}^k |F_i^{l,U}|$ is the total no. non-relevant images among the retrieved images, for the i^{th} feature component. The dominant range of a feature component is found by the minimum and maximum values from the set of relevant samples. The value of δ_i^k lies between 0 and 1. It is 0 when all non-relevant images are within the dominant range and thus, no weight should be given for that feature component. On the other hand, when there is not a single non-relevant image lying within the dominant range, maximum weight should be given to that feature component. They used the following weight factor, for details see [8]:

$$weight - type2 : w_i^{k+1} = \frac{\delta_i^k}{\epsilon + \sigma_{rel,i}^k} \ . \tag{6}$$

In order to maximize the benefits in separating relevant images from the non-relevant ones, we introduced weight-type3 where we combined the above discriminant ratio with the weight factor in eqn (4). This resulted in eqn (7) and our experimental results also demonstrated the synergy of the weight-type 1 and weight-type 2.

$$weight - type3 : w_i^{k+1} = \delta_i^k * \frac{\epsilon + \sigma_{N_r,i}^k}{\epsilon + \sigma_{rel,i}^k} \ . \tag{7}$$

3 Experiment

To demonstrate the goodness of our weight factor we experimented with three databases of different sizes and different number of semantic categories. All images are of 256×256 pixels size. An image in the retrieved list is considered to be relevant if that image comes from the same category as the query image, otherwise, non-relevant.

We used precision as a measure of system performance which is given by the following formula:

$$precision = \frac{No.\ of\ relevant\ retrieved\ images}{No.\ of\ retrieved\ images}\ . \tag{8}$$

3.1 Image Database and Ground Truth

1. ImageDB2000: This consists of 2000 images from 10 different categories (Flowers, Vegetables and Fruits, Nature, Leaves, Ships, Faces, Fishes, Cars, Animals, Aeroplanes). Each category contains 200 images. We used all 2000 images as query images and calculated performance in terms of Precision after averaging over all query images.
2. ImageDBCaltech: From http://www.vision.caltech.edu website, we obtained the Caltech-101 image database. This consists of 9144 images from 102 categories with 34 to 800 images per category. We omitted all grey scale images including the categories Binoculars, Car-side, Camera and Yin-yang that had mostly grey scale images. Finally, we experimented with 8365 images from 98 categories. As query images, we chose 4 images randomly from each category, thus a total of 392 query images. Results presented are obtained by averaging results over 392 query images.
3. ImageDB2020: This consists of 2020 images from 12 categories. The number of images per category varies from 96 to 376. We measured performance by averaging performance over all 2020 images used as query.

3.2 Result Analysis

Table 1 shows the results for all three data sets for scope 20 and table 2 for scope 100. In figure 1, figure 2 and figure 3, the improvement in precision from 0rf to 1rf are almost the same, however, the improvement in the next iterations is much higher with our weight factor. For scope 20, the improvement in precision with weight-type3 compared to weight-type2 is 5.625 % for ImageDB2000, 1.365% for ImageDBCaltech and 4.52% for ImageDB2020. For scope 100, the precision improvement with weight-type3 compared to weight-type2 is 10.426 % for ImageDB2000, 1.089% for ImageDBCaltech and 5.23% for ImageDB2020.

The significant improvement in precision, for all three weight types, from 0rf to 1rf shows the impact of the use of relevance feedback mechanism. From 1rf to 5rf the improvement is much higher for weight-type3. In other words, the convergence towards the highest possible precision value is higher for this weight

Table 1. Improvement in Precision (%) from 0rf to 5rf, Scope = 20

	weight factor	0rf	1rf	2rf	3rf	4rf	5rf
ImageDB2000	weight-type1	56.002	73.613	75.740	78.488	78.745	79.923
	weight-type2	56.002	71.408	76.605	77.575	78.050	78.235
	weight-type3	56.002	72.455	78.567	82.048	82.980	83.860
ImageDBCaltech	weight-type1	11.811	13.801	14.732	15.242	15.523	15.689
	weight-type2	11.811	13.724	14.592	14.707	14.758	14.834
	weight-type3	11.811	13.801	15.089	15.714	16.110	16.199
ImageDB2020	weight-type1	50.280	63.777	66.381	68.545	68.765	69.639
	weight-type2	50.280	61.399	65.567	66.653	67.037	67.252
	weight-type3	50.280	61.990	67.757	70.250	71.002	71.772

Table 2. Improvement in Precision (%) from 0rf to 5rf, Scope = 100

	weight factor	0rf	1rf	2rf	3rf	4rf	5rf
ImageDB2000	weight-type1	37.264	55.131	56.919	60.060	60.004	61.280
	weight-type2	37.264	53.614	55.054	54.138	53.445	52.877
	weight-type3	37.264	56.259	59.984	62.160	62.465	63.303
ImageDBCaltech	weight-type1	5.526	7.640	8.033	8.607	8.556	8.755
	weight-type2	5.526	7.571	7.982	8.133	8.066	8.077
	weight-type3	5.526	7.793	8.406	8.936	9.074	9.166
ImageDB2020	weight-type1	34.315	48.811	50.579	52.686	52.990	53.611
	weight-type2	34.315	47.467	49.717	49.967	49.787	49.363
	weight-type3	34.315	48.735	52.299	53.862	54.126	54.593

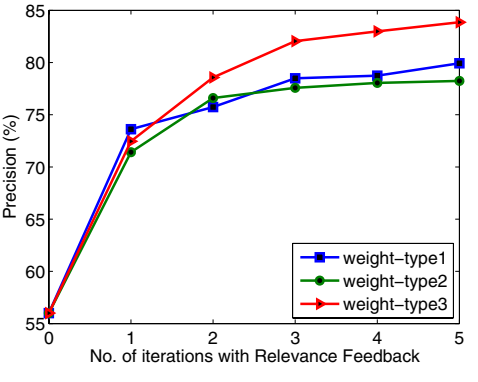

Fig. 1. ImageDB2000: Improvement in precision with RF at scope 20

factor. We reported results up to 5 iterations as after that there is no significant improvement in precision.

It may be worth noting that the precision values for ImageDB2000 and ImageDB2020 are in the range of 50% to 84% whereas for ImageDBCaltech, they

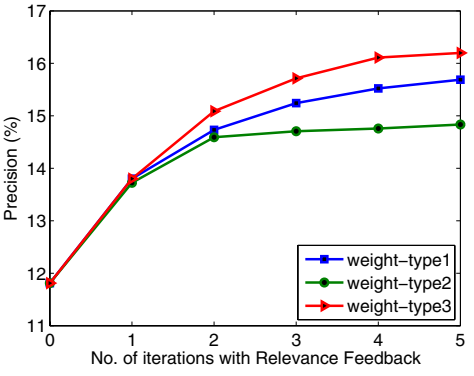

Fig. 2. ImageDBCaltech: Improvement in precision with RF at scope 20

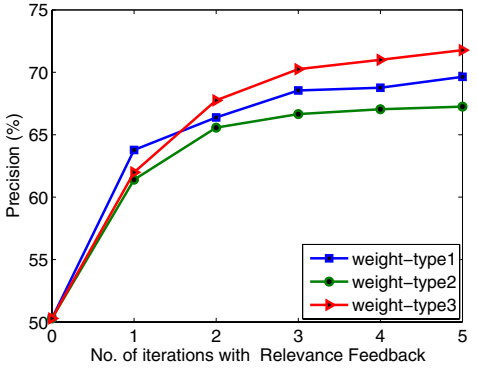

Fig. 3. ImageDB2020: Improvement in precision with RF at scope 20

are in the range of 11% to 16%. This huge difference in precision is clearly explained by the well known result that in the C-category classification problem the classification accuracy is of the order of $\frac{1}{C}$ for random allocation [13]. The effectiveness of feature vector i.e the discriminatory power of feature vector and the inherent separability of classes in the feature space considered are also very important in determining overall precision.

4 Conclusions and Future Directions

To improve system performance with relevance feedback, we used the synergy of weight-type1 and weight-type2 intuitively to obtain weight-type3. For all 3 image databases, weight-type3 performed the best. The experimental results conformed to our expectation. We have also discussed that the convergence towards the highest possible precision value is higher with our weight factor compared to the other two weight factors considered.

Also, the number of semantic categories in the database plays an important role in the retrieval results. Precision usually worsens as the number of categories increases.

In reality, the number of samples per category as well as the number of feedback samples can be very small. The situation becomes worse in high dimensional feature space. In feature re-weighting method, the calculation of standard deviation used in the weight factor becomes inaccurate and thus, the class representation becomes poor. In our current research we are addressing this issue in order to have better class representation and hence, more reliable retrieval results.

References

1. Zhang, H.: Relevance Feedback in Content-Based Image Retrieval, Multimedia Information Retrieval and Management-Technological Fundamentals and Applications, Feng, D.D., Siu, W. C., Zhang, H. (Eds), Chap 3, Springer-Verlog, Germany (2003)
2. Rui, Y., Huang, T. S., Chang, S.: Image Retrieval: Current Techniques, Promising Directions and Open Issues, Journal of Visual Communication and Image Presentation, volume 10, no. 4 (April 1999)
3. Ortega-Binderberger, M., Mehrotra, S.: Relevance Feedback in Multimedia Databases, Handbook of Video Databases: Design and Applications, CRC Press, Chap 23 (2003) 1-28
4. Rui Y., Huang, T.S., Ortega, M., Mehrotra, S.: Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval, IEEE Transactions on Circuits and Video Technology, Special issue on Segmentation, Description, and Retrieval of Video Content (September 1998) 644-655
5. Aksoy, S., Haralick, R.M.: A Weighted Distance Approach to Relevance Feedback, International Conference on Pattern Recognition, Barcelona Spain (September 2000)
6. Hore, E.S, Ray, S.: A Sum-result Indexing Algorithm for Feature Combining in Content-Based Image Retrieval, Proceedings of the Fourth IASTED International Conference Signal and Image Processing, Hawaii USA (August 2002) 283-287
7. Ishikawa, Y., Subramanya, R., Faloutsos, C.: MindReader: Querying Databases through Multiple Examples, Proceedings 24th International Conference on Very Large Data Bases (VLDB)(1998)
8. Wu, Y., Zhang, A.: A Feature Re-weighting Approach for Relevance Feedback in Image Retrieval, Special issue on Segmentation, Description, and Retrieval of Video Content, Rochester NewYork (September 2002)
9. Das, G., Ray, S.: A Compact Feature Representation and Image Indexing in Content-Based Image Retrieval, Proceedings of Image and Vision Computing New Zealand 2005 Conference (IVCNZ 2005), Dunedin New Zealand (28-29 November 2005) 387-391
10. Huang, J.: Color-spatial Image Indexing and Applications, PhD Dissertation, Cornell University (1998)
11. Shim, S., Choi, T.: Image Indexing by Modified Color Co-occurrence Matrix, International Conference on Image Processing (September 2003)
12. Haralick, R.M., Shanmugam,K., Dinstein, I.: Textural Features for Image Classification, IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, No.6 (November 1973) 610-621
13. Duda, R. O., Hart, P. E., Stork, D. G.: Pattern Classification, 2nd ed, Wiley-Interscience, New York (2000).