A Compact Feature Representation and Image Indexing in Content-Based Image Retrieval

G. Das and S. Ray

Clayton School of Information Technology
Monash University, Australia
email: {Gita.Das, Sid.Ray}@csse.monash.edu.au

Abstract

Proper selection of features and their representation are essential for a good Content-Based Image Retrieval (CBIR) system. In this paper, we propose a compact feature representation based on the elements of Colour Co-occurrence Matrices (CCM) in Hue, Saturation, Value (HSV) colour space. To represent each image in the database, we constructed a feature vector considering all diagonal elements and Sum-Average [1] of all non-diagonal elements of the CCM. We demonstrate that our feature representation is superior in terms of system accuracy and online computation time which are attributed to dimension reduction. We present experimental results with a database of 2000 images from 10 categories. With our method, we achieved up to 3% improvement in precision. With relevance feedback, we obtained a further 12% improvement as opposed to 9% with original higher dimension.

Keywords: compact feature vector, dimension reduction, feature re-weighting, CBIR, relevance feedback

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. The concept of co-occurrence matrix in texture has been known for long [1], however, its use in colour has been reported very recently [2], [3], [4]. A CCM represents how the spatial correlation of colour changes with distance i.e. pixel positions. In [3], a Modified Colour Co-occurrence Matrix (MCCM) is used where a CCM of Hue is simplified to represent the number of colour pairs between adjacent pixels (4-neighbourhood). They did not consider the adverse effect of ignoring Saturation and Value components of colour. The diagonal elements convey the colour information of the entire image whereas the non-diagonal elements represent the shape information in an indirect way. In [4], Ojala et. al demonstrated the effect of quantization of H,S,V levels on retrieval accuracy.

In this paper, we studied the effect of H-only and H,S,V together on the system performance. Although, addition of S and V increases feature dimension, it provides extra information about images and thus improves retrieval. The use of all matrix elements in the feature vector as in [2], [3] and [4], will contribute to online computation significantly and this can be worse in real situations where database size is very large. Also, 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 [5], [6]. Hence, dimension reduction has become a critical issue in feature representation and image indexing of CBIR systems [7]. If there is no information loss due to dimension reduction then system performance would be the same in both spaces. However, that is an ideal situation. 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 [3]. Also, we have noticed that most of the non-diagonal elements are zero. Thus representing all the non-diagonal elements with a single Sum-Average [1] value (for details, see section 2) attributes to the following benefits:

1. the Sum-Average of non-diagonal elements would be less sensitive to noise and thus enhance retrieval performance,
2. the dimension is reduced significantly, thus reducing online computation and retrieval time,
3. compared to other methods of dimension reduction e.g. Principal Component Analysis (PCA)[7], computing Sum-Average is very simple.

We used all the diagonal elements in the feature vector as they are the majority and manipulating them in any way would contribute to loss of information content of images significantly. Thus, we constructed a feature vector with all diagonal elements and Sum-Average of non-diagonal elements from H,S,V matrices. For rest of the paper, we refer the original feature space as original dimension and our one as reduced dimension. With our method, we achieved improved performance and faster retrieval.

In CBIR systems, relevance feedback is found to be very useful in reducing semantic gap, which is a bottleneck to achieve a successful CBIR system [8], [9]. 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 to his/her query. The system uses this information to improve retrieval results. We used a feature reweighting approach to implement relevance feedback, where weight of each component in the similarity measure is updated according to information extracted from the previous iteration. The small number of training samples can cause problem especially in high dimension. We have shown that precision is better with lower dimension.

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

2 Methodology

2.1 Feature Representation

We used the following nomenclature:

N: Number of images in the database

N_r: Scope i.e. the number of retrieved images

Q, I: Query image and Database image respectively

M: Number of components in the feature vector

L: Number of quantization levels in H,S,V

k: Iteration number in relevance feedback

w_i^k: weight factor for i^{th} feature component in k^{th} iteration

Let P be the $L \times L$ co-occurrence matrix whose element p_{xy} indicates the number of times a pixel with colour level x occurs, at a distance d, relative to pixels with colour level y. To incorporate the Sum-Average as described in [1] into our case, we introduced the following formula:

$$Sum\_ndiag = \sum_{x=1}^{L-1} \sum_{y=x+1}^{L} (x+y)p_{xy} \qquad (1)$$

where Sum_ndiag are Sum-Average of non-diagonal elements of P. We chose HSV colour model as it is known to be perceptually uniform [8]. We tried to make the spatial correlation more sensitive to Hue and less sensitive to Value and Saturation. We experimented with different levels of quantization and found H,S,V=16,3,3 to be a good choice. As we increased the level of quantization of H,S,V, we obtained better precision but after a certain range the change in precision was insignificant. So we decided to use 16,3,3 to keep computation low while still maintaining good precision. We chose co-occurrence distance d=3 and used pixel pairs in both vertical and horizontal directions. Thus we obtained symmetric matrices and needed only upper diagonal elements to consider. For a 16x16 matrix, the number of diagonal elements is 16 and the number of non-diagonal elements is 120. For a 3x3 matrix, this number is 3 for both diagonal and non-diagonal elements. In our method, we represented all non-diagonal elements by a single value. Thus, for H,S,V=16,3,3 the feature dimension is 148 and it is reduced to 25 in our approach.

As different feature components have different range (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,

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

where $f_{i,org}$ is the original i^{th} feature component, μ_i is the mean and σ_i is the standard deviation 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:

$$f_i = \frac{f_i^{'} + 1}{2} \qquad (3)$$

2.2 Image Indexing

To find the similarity between I and Q, we used a weighted Minkowski distance measure, a commonly used metric in CBIR,

$$D(I,Q) = \sum_{i=1}^{M} w_i * |f_{iI} - f_{iQ}| \quad (4)$$

where w_i is the weight of i^{th} feature component. When there is no relevance feedback, equal weight values are used for each feature component. When relevance feedback is used the weights can be updated (this is the topic of another paper by the authors to be published elsewhere) according to the following formula:

$$w_i^{k+1} = \frac{\sigma^k_{N_r,i}}{\sigma^k_{rel,i}} \quad (5)$$

Here, $\sigma^k_{N_r,i}$ is the standard deviation of i^{th} feature component over N_r retrieved images and $\sigma^k_{rel,i}$ that over the relevant retrieved images in k^{th} iteration. Please note that in [10] and [11], they have used similar weight factors, however, there the numerator represented standard deviation over the entire database. A good feature component should have a large variation over the entire database images and a small variation over the relevant images. As the feature values are normalized, the standard deviation over the entire database remains unchanged with iteration and thus does not provide any extra information. However, a feature component that has small variation over relevant images and a large variation over the retrieved images is considered to be good and should be given more weight. Also, with each iteration a new set of images is likely to be retrieved and a new $\sigma^k_{N_r,i}$ obtained. This is why we opted to use eqn (5) in weight calculation. In a few cases, we had only one image retrieved meaning $\sigma^k_{rel,i}$ equals to zero. To avoid this situation of infinite weight value, we used a modified formula as below:

$$w_i^{k+1} = \frac{1+\sigma^k_{N_r,i}}{1+\sigma^k_{rel,i}} \quad (6)$$

3 Experimental Results and Analysis

The data set comprised 2000 images from 10 different categories, namely, Flowers, Fruits and Vegetables, Nature, Leaves, Ships, Faces, Fishes, Cars, Animals, Aeroplanes. Each category contains 200 images. All images have same pixel size of 256×256. We used precision as a measure of system performance which is given by the following formula:

$$precision = \frac{No.\ of\ relevant\ retrieved\ images}{No.\ of\ retrieved\ images} \quad (7)$$

We used all 2000 images as query images and calculated precision after averaging over all query images. This way it ensures the true representation of system performance [12], [13]. Results are reported for different scope values, that is, the number of retrieved images. First, we investigated the effect of H-only space and H,S,V together. As expected we obtained much improvement in precision for H,S,V together. At scope = 200, it is 2.409% in original dimension and 5.849% in reduced dimension (see figure 1). Then, we experimented with

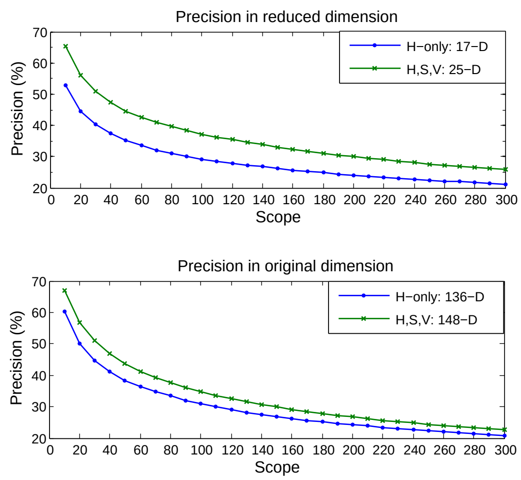

Figure 1: Effect of H-only and H,S,V together on precision at different feature dimensions

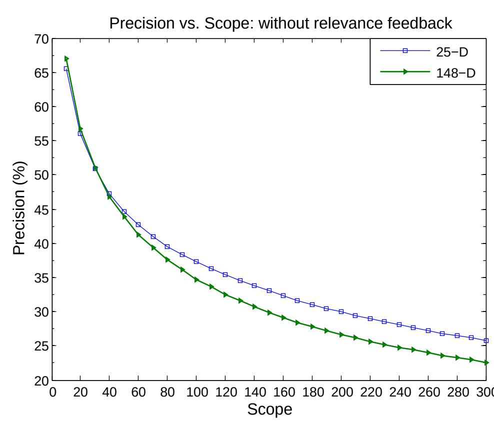

Figure 2: Performance comparison in original dimension and reduced dimension

25-D ("D" stands for dimension) feature vector and 148-D feature vector for scope = 300 (see figure 2).

As, the number of samples per category is 200, we should be using a maximum scope of 200. However, this is for an ideal system where all the relevant images are retrieved as top 200 images. We wanted to look at the behaviour of both methods at a scope

Table 1: Improvement of Precision (%) from 0rf to 5rf

Scope	**148-D**	**25-D**
20	16.093	16.272
200	9.12	12.583

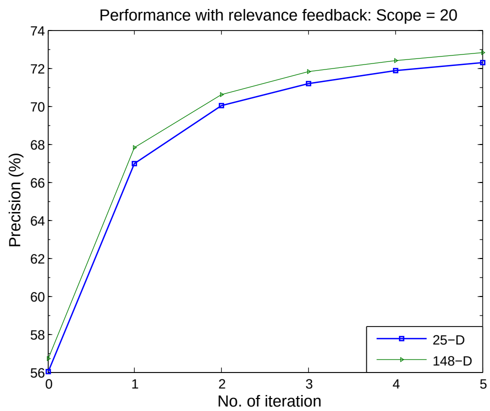

Figure 3: Improvement in precision with relevance feedback at scope 20

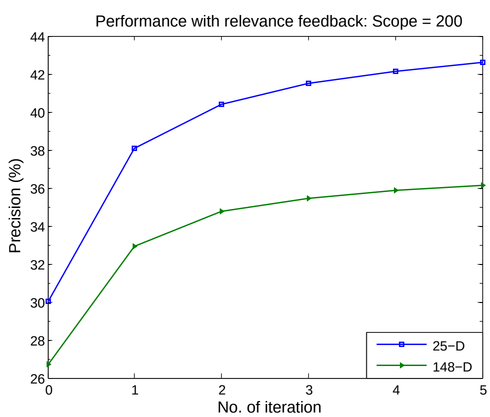

Figure 4: Improvement in precision with relevance feedback at scope 200

beyond 200 where relevant images will be all over the space.

At scope = 20, precision with 25-D is marginally worse by 0.7% but at scope = 200, it is improved by 3.297%. The effect of dimension reduction is more prominent at higher scope.

In figure 3 and figure 4, we have shown results up to 5 iterations as after that performance does not improve significantly, which is good indicating that the algorithm converges fast. The improvement with relevance feedback (from no iteration to fifth iteration) is more with 25-D compared to 148-D and also, it is more prominent at higher scope, see table 1. It is well known that more training samples are required as feature dimension increases. Thus with iteration, improvement is more with reduced dimension compared to original dimension.

4 Conclusion and Future Direction

The addition of S and V space with H space contributes to the information content of images and thus enhances system performance significantly. The online computation with original dimension is $O(3L^2)$ whereas it is $O(3(L+1)$ in our method. This reduction in number of computations will be very significant in today's real situation where image database size is already very big and is increasing day by day. At low scope, the precision with our method is marginally worse which can be attributed to the information loss due to dimension reduction. At higher scope, the effect of low dimension becomes more prominent and as a result, precision with our method is much better.

However, we need to perform further experiments with a larger dataset and with different datasets to get more confidence in experimental findings and generality in system performance.

Also, we need to investigate other existing methods (e.g. PCA) of dimension reduction and compare performance of each method.

References

[1] R. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 6, vol. SMC-3, pp. 610–621, November 1973.

[2] J. Huang, *Color-spatial Image Indexing and Applications*. Cornell University: PhD Dissertation, 1998.

[3] S. Shim and T. Choi, "Image indexing by modified color co-occurrence matrix," in *International Conference on Image Processing*, September 2003.

[4] T. Ojala, M. Rautiainen, E. Matinmikko, and M. Aittola, "Semantic image retrieval with hsv correlograms," in *Proceeding of 12th Scandinavian Conference on Image Analysis*, (Bergen, Norway), pp. 621–627, 2001.

[5] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, Vol. 19, pp. 153–158, February 1997.

[6] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification, 2nd ed.* New York, Chichester, John Wiley and Sons, 2001.

[7] P. Wu, B. Manjunath, and H. Shin, "Dimensionality reduction for image retrieval," in *Proceeding IEEE International Conference on Image Processing (ICIP 2000)*, (Vancouver, Canada), pp. 726–729, Vol. 3, September 2000.

[8] Y. Rui, T. S. Huang, and S.-F. Chang, "Image retrieval: Current techniques, promising directions and open issues," *Journal of Visual Communication and Image Representation*, no. 4, Vol. 10, April 1999.

[9] M. Ortega-Binderberger and S. Mehrotra, *Relevance Feedback in Multimedia Databases*, pp. 1–28. CRC Press, Chap 23, 2003.

[10] S. Aksoy and R. Haralick, "A weighted distance approach to relevance feedback," in *International Conference on Pattern Recognition*, (Barcelona, Spain), September 2000.

[11] E. Hore and S. Ray, "A sum-result indexing algorithm for feature combining in content-based image retrieval," in *Proceedings of the Fourth IASTED International Conference Signal and Image Processing*, (Hawaii,USA), pp. 283–287, August 2002.

[12] D. P. Huijsmans and N. Sebe, "How to complete performance graphs in content-based image retrieval: Add generality and normalize scope," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, Vol. 27, February 2005.

[13] H. Muller, W. Muller, D. M. Squire, and T. Pun, "Performance evaluation in content-based image retrieval: Overview and proposals," in *Technical Report, Computer Vision Group, University of Geneva*, December 1999.