A Complete and Efficient Low-Dimensional Model for Content-Based Image Retrieval

Santosh Kulkarni Bala Srinivasan

School of Computer Science and Software Engineering
Monash University, Australia
srini@monash.edu.au

Abstract

Cotent-based image retrieval (CBIR) has become one of the most active research areas in the last ten years. This is because the number of applications which use image data is increasing. The main task of a CBIR system is to retrieve images relevant to the user's need. This is accomplished by comparing the content of the various images present in the database. Many visual representations have been explored by researchers and different systems have been built. Most of the CBIR systems use different image features such as colour and texture to retrieve similar images. Each feature is represented by a set of components in a feature vector. CBIR systems use a large number of components in order to improve the retrieval quality. As the number of components increase, the indexing process becomes complicated. This is because high dimensional structures have to be designed to store the large number of components associated with each image. Therefore, there is a tradeoff between the indexing complexity and the retrieval quality.

In this paper, we present an image model which reduces the size of the feature vectors and therefore makes it low dimensional. It is termed as the Low-Dimensional Image Model (L-DIM). The model uses the concept of eigen-vectors and eigen-values to identify the dominant components in the image feature vectors. This allows the database administrator to ignore the less significant components and therefore reduce the dimensionality of the image model. The low dimensionality simplifies the indexing process as the number of components to be searched and retrieved is very small. All this is achieved without decreasing the retrieval performance. Experimental results support the properties deduced from L-DIM. In order to account for user subjectivity, a relevance feedback mechanism is also incorporated with L-DIM.

Keywords: Content-Based Image Retrieval, Dimensionality reduction, Multimedia Databases

1 Introduction

Image retrieval is a growing area of research where the primary focus is on improving the retrieval quality. Many image retrieval systems use multiple image features such as colour, texture and shape to achieve this goal. The idea behind this approach is that if one of the image features fails to capture any important (relevant) information from the images, the other feature(s) are likely to capture that information, therefore improving the overall retrieval quality. A large number of components are required to represent image features effectively. Indexing these image features together causes the combined index to be very large in size (in the order of hundreds or thousands). Standard indexing structures such as B-trees and R-trees [3] are unable to handle such high dimensional data and thus the indexing performance degrades. Search and retrieval time increases exponentially when these indexing structures are used for indexing high dimensional data. High dimensional structures also suffer from the nearest neighbour problem [1]. It has been proved that as the dimensionality increases, the distance to the nearest neighbour approaches the distance to the farthest neighbour [1]. This is another drawback of high dimensional structures.

In this paper, the effect of reducing the dimensionality of the feature vectors is investigated. If the dimensionality is reduced by randomly eliminating components from the feature set, the retrieval performance would decrease as vital information will be lost with the elimination of certain components. This paper presents a solution to reduce the dimensionality of the feature set without affecting the retrieval performance. The approach is mathematically modelled and the image model is motivated by the latent semantic indexing model proposed for text retrieval [2]. The main objective of the model is to reduce the dimensionality of the feature vectors without affecting the retrieval performance. The low dimensionality makes the model very compact and expedites the indexing and retrieval process.

Unlike traditional database systems, the retrieval in image databases is based on similarity of feature, and not the actual image itself. As explained, the size of the feature vector is very large and consists of multiple feature components. We argue that the feature vectors contain some amount of redundant information which is directly proportional to the size of the feature vector. The problem therefore is to identify the important components which provide the maximum discrimination among the images in the database. A propery of SVD decomposition of matrices is exploited to identify the most discriminating components. The database administrator can therefore ignore the redundant components and reduce the size of the feature set. Eliminating the redundant components makes the model very compact and simplifies the indexing process. It also improves the search and retrieval time without decreasing the retrieval quality. A relevance feedback mechanism is incorporated into the L-DIM retrieval process in order to account for user subjectivity and make the process interactive in nature. The rest of the sections explain the working of the model in detail.

2 The Image Model

2.1 Mathematical Notions

Given any $m \times n$ matrix X having a rank p, we can decompose into three matrices U, Σ, V, such that, $X = U \Sigma V^*$. Here Σ is a $p \times p$ positive diagonal matrix(the diagonal elements are all positive and all other are zero). U and V are isometric matrices of order $m \times p$ and $n \times p$ respectively. V^* indicates the adjoint of the matrix V. This decomposition is known as Singular Value Decomposition (SVD) and finds a wide application. There are standard algorithms to obtain SVD of a given matrix. In particular, if every element of X is a real number then V is also real. This means that the transpose of V, V^T, is same as the adjoint matrix V^*. The first p columns of the matrices U and V are called the left and right singular vectors respectively.

SVD is closely related to the standard eigenvalue-eigenvector or spectral decomposition of a square matrix, Y, into $VLV^{'}$, where V is orthonormal and L is diagonal [2]. In fact U and V of SVD represent the eigen vectors for $XX^{'}$ and $X^{'}X$ respectively.

2.2 The Image Model

The main motivation in designing the image model is to reduce the dimensionality of the feature vectors. Our hypothesis is that the size of the feature set is very large, in the order of hundreds. Only a small subset of the components are significant for a given data set. The rest of the components do not discriminate between the images in the collection. These features can be ignored as their contribution in the retrieval process is minimal. The model therefore tries to identify the dominant components from the entire image feature vector. In order to achieve this, the entire feature set needs to be pre-processed and the relationship between the components has to be examined. The technique to achiev this is explained below.

2.2.1 Matrix formation and decomposition

All the attributes to be used for retrieval are computed and stored in a feature set for each image. If the size of the feature set is K and the total number of images is N, then a matrix of size $K \times N$ is contructed. All the feature components are normalised to unit length. This is because, in order to use the eigen decomposition, the eigen vectors have to be of unit length. The normalised matrix is decomposed using Singular Value Decomposition into three matrices. The three matrices obtained are $X = T_0 S_0 D_0^{'}$. T_0 and D_0 are the orthonormal matrices and S_0 is the diagonal matrix of size $m \times m$, where m is the rank of the original matrix X. These matrices decompose the original information into linearly independent components or factors and enables us to identify the dominant factors. Matrices T_0 and D_0 represent the eigenvectors of XX^T and X^TX respectively and the square of the values of the diagonal matrix (S_0) are the eigenvalues. The matrix S_0 therefore provides the scaling factor which determines the magnitude of the vectors, depending on its elements. The matrix contains many elements which are non-zero but so small that their contribution to the discrimination of the images is insignificant. Such elements can be ignored in the retrieval process. Hence the significant values from the matrix S_0 are identified and the rest of the values are set to zero. The produce of the resulting reduced matrices is denoted by $\hat{X}$. Matrix $\hat{X}$ is used as a base in the retrieval process. The rank k of the matrix $\hat{X}$ depends on the amount of elimination done to the original matrix. The amount of reduction (the choice of k) is very important and depends on the different attributes used and the natire of the images in the database. Ideally the value of k should be large enough to incorporate all the important information (dominant values) in the image data but it should be small enough to ignore all the unimportant details. Once the dominant attributes have been identified and the image data has been reduced in size, the resultant images are represented as vectors in a multidimensional space. The dimensionality of the vector space depends on the reduction factor (the value of k). As the number of dominant values retained is low, the resultant model is low dimensional.

2.3 Geometric representation of the model

The resultant matrix $\hat{X}$ can be represented geometrically in a k-dimensional spatial configuration (the value of k depends on the number of components retained in the diagonal matrix). As the matrix $\hat{X}$ stores the approximated features along the rows, the rows of matrix D, represent the unit vectors of all the images in the database. The matrix S is the matrix of the reduced eigenvalues which provides the scaling factor for the vectors. The rows of the matrix DS would therefore provide the resultant vectors for the images. If the diagonal matrix S is of size $k \times k$ (k non-zero elements along the diagonal), the matrix DS would have k rows. As the rows of the matrix DS represent the image vectors, that each image vector has k elements and is therefore k-dimensional. Hence we need a k-dimensional space to represent the images. It should be noted that the relation between taking D as the coordinates for the images and DS as the coordinates is that since S is a diagonal matrix of eigen values, the direction of the vectors is not affected except that the vectors are stretched or shrunk depending on the elements of S.

2.4 Comparing the Image Vectors

The cosine of the angle between the image vectors is used to compare the image vectors. If $\vec{a}$ and $\vec{b}$ are two image vectors and θ is the angle between them, we know from the definition of dot product of two vectors that:

$$\cos\theta = \frac{\vec{a}.\vec{b}}{|a||b|} \tag{1}$$

As all the image vectors are normalised to unit length, we have:

$$\cos\theta = \vec{a}.\vec{b} \tag{2}$$

Therefore the dot product of the corresponding columns of $\hat{X}$ would reflect the extent to which two images are similar. The matrix $\hat{X}'\hat{X}$ is the square symmetric matrix which contains all the image to image dot products. As we know that S is a diagonal matrix and T and D are orthonormal matrices, we have:

$$\begin{aligned}
\hat{X} &= TSD' && (3)\\
\hat{X}'\hat{X} &= (TSD')'TSD' && (4)\\
&= D(TT')S^2D' && (5)\\
&= D(I)S^2D' && (6)\\
\hat{X}'\hat{X} &= DS^2D' && (7)
\end{aligned}$$

Therefore the i,j cell of $\hat{X}'\hat{X}$ can be obtained by taking the dot product between the i and j rows of the matrix DS. We can hence consider the rows of the matrix DS are coordinates of points in space and take their dot products in this space (as we know that DS is the stretched version of D space).

3 Relevance Feedback

Content-based image retrieval provides an automated method to retrieve similar images based on the image's content. Inspite of the extensive research effort in this field, the retrieval techniques used in content-based image retrieval systems lag behind the corresponding techniques in today's text search engines such as Alta Vista, Yahoo and Lycos that are currently available. The performance is not satisfactory due to the following reasons:

- *Difference between high level concepts and low level features:* Content-based image retrieval (CBIR) systems focus on finding different representations for the low level image features. For example, there are different ways of representing colour such as colour histogram, colour moments and colour sets. Texture can be represented by Gabor filters, cooccurence matrices, fractals, etc. Researchers try to use the best possible representation for each feature in order to achieve good results.

  A major problem with this approach is that high level concepts are not considered in the retrieval process. Users of an image retrieval system are not aware of the low level features and are interested in retrieving images based on higher level image semantics. Some systems map the high level semantics to low level image features using certain object recognition techniques. This works to a certain extent with simpler objects (eg. an apple could be identified as a round object with red colour) but it is hard to map complex images into low level features (eg. an image with multiple overlapping objects). This gap between the two levels degrades the performance of image retrieval systems.

- *Subjective human perception:* Human vision is subjective in nature. Different users could perceive the same image differently depending on their requirements. Even the same user may have a different perception about an image at a different time if the user requirements change. For example, a particular person might be interested in the texture property of the images whereas another person could be interested in the colour property for the same set of images. In Figure 1, a person who is interested in the colour of the flowers would consider images (a) and (c) to be closer than the images (a) and (b), whereas a person who is keen on the shape of the flowers would consider the images (a) and (b) as closer compared to the images (a) and (c). Even if a group of users are interested in retrieving images based on texture, the way they perceive the similarity of texture may vary from one user to another. This is illustrated in Figure 2. Among the three texture images, some users may say that images (a) and (b) are more similar if they do not care about the intensity but some users may say that images (a) and (c) are more similar if they are interested in the intensity of the images. This difference in perception therefore affects the performance of image retrieval systems as the systems fail to capture human perception.

Content-based image retrieval systems have a fixed representation for each feature which cannot be changed to model user subjectivity. Some systems allow the user to assign weights to the low level features before the retrieval process. This imposes a burden on the user as it requires the user to have a comprehensive understanding of the low level feature representations used in the retrieval process. This is normally not the case with the average user. Due to these limitations, image retrieval systems started involving the user in the retrieval process making it interactive in nature.

The different interactive image retrieval techniques are modelling user subjectivity [6, 7], supervised learning similarity measures [4] and interactive relevance feedback [9, 10, 8, 11]. In this section, a relevance feedback mechanism is proposed which is incorporated with L-DIM to improve its performance. The main advantage of the model is that it is universal and can be used by any standard image retrieval system The following subsection explains the feedback model.

3.1 Relevance feedback model

A standard image model is first defined as a 3 tuple, $I_m =< R, F, S >$ where R is the raw image data, F is the set of feature components and S is the similarity measure(s) used to compute how similar or dissimilar two images are. There is also a weight associated with each feature component. Therefore if there are k components in the feature vector F= $(f_1, f_2, \ldots, f_i, \ldots, f_k)$ then there are k different corresponding weights, $W_1, W_2, \ldots, W_k$. Based on this image model, the relevance feedback method is described below:

1. All the weights are initialised to a fixed value of 1. $W_1 = W_2 = \ldots = W_k = 1$.

2. The feature components are updated by multiplying them with their corresponding weights as follows:
$$\forall_{i=1,k} \quad f_i = W_i f_i \tag{8}$$

3. The query image is represented as follows: $I_q = [q_1, q_2, \ldots, q_k]$ where $q_1, q_2, \ldots, q_k$ are the feature components for the query image. The query feature vector is also updated by multiplying it with the corresponding weights:
$$\forall_{i=1,k} \quad q_i = W_i q_i \tag{9}$$

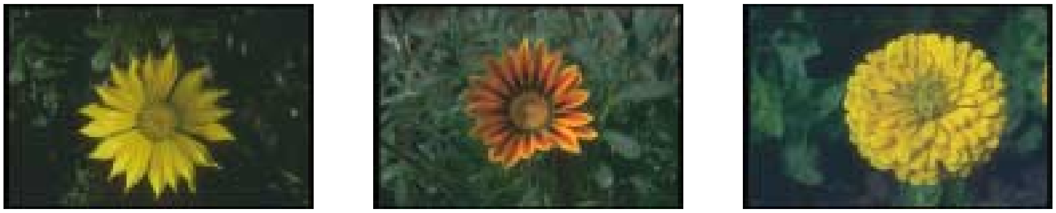
4. The similarity between the query image (I_q) and the database images ($I_1, I_2, .., I_n$) is computed using the image retrieval systems appropriate similarity measure (M), $SimM(I_q,I_i)$, where I_q is the query image and I_i is the i^{th} image in the database.

5. The database images are then ordered according to their proximity to the query image I_q. The system displays the top L images to the user. The value of L can vary according to the user's need.

6. The user marks the highly relevant images from the set of displayed images. The weights associated with each feature are then updated based on the user's feedback. The features that better represents user's perception will be assigned higher weights. The procedure for updating the weights is explained in the next section.

7. The image feature components are updated using equation 8. The similarity of the images is recomputed using the updated values of the feature components and displayed to the user.

8. This procedure (steps 2 - 8) is repeated until the user is satisfied with the results.

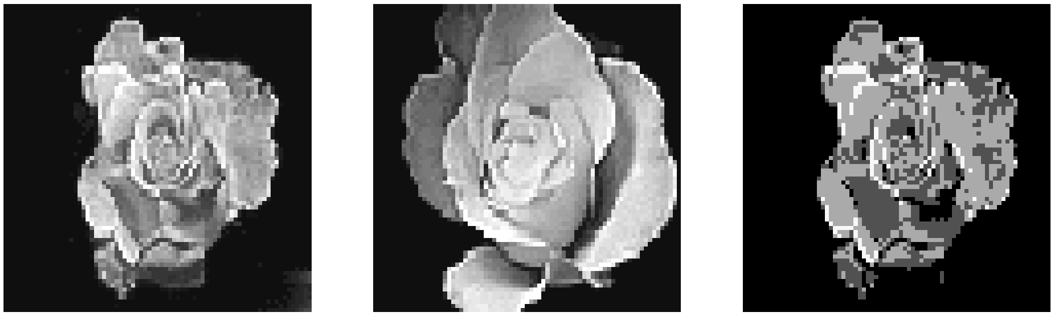


(a) a

(b) b

(c) c

Figure 1: Similar images



(a) a

(b) b

(c) c

Figure 2: Similar images based on texture

3.2 Computing the Feature Component Weights

The users of image retrieval system can have different perception about the components in the feature vector. For example, a particular user may perceive blue colour more significantly than red colour and hence when computing the weights a higher value should be assigned to the blue colour pixels as compared to the red colour pixels (in the RGB spectrum). The relevance feedback mechanism should be able to capture the user's perception and retrieve images accordingly. Images are normally retrieved based on low level image features whereas the user's thoughts are in terms of high level semantic objects. A mechanism is designed to compute the weights for the different features which is based on a standard deviation approach. The feature vectors of all the images that are returned by the user as relevant are examined. If the user returns $R (R \leq L)$ relevant images, a matrix of size $R \times K$ is formed using the feature vectors which have K components each. If a particular component in this matrix has very similar values, it implies that it is an important component from the user's perspective. This component should therefore be assigned a higher weightage. On the contrary, dissimilar values for a particular component imply that the component is not relevant to the user's information need and should be assigned a lower weight. The matrix M has K columns each representing one feature component. Each column has R elements. If the R values are similar then a higher value is assigned to the weight corresponding to this component and vice versa. The weights are computed as follows:

- The inverse of the standard deviation of each column is used to update the weight for that column. Standard deviation shows the variation of the feature values over the entire set. Therefore if the standard deviation is high for a particular feature component, the feature values are spread apart and not very similar. This particular feature component is therefore not important and should be assigned a lower weight. The inverse of the standard deviation ($W_i = 1/\sigma_i$) would be a small value in this case as σ_i is high. The same concept is applicable to the feature components where the standard deviation is very low.

These components have very similar values and are important components as far as the user is concerned. They should therefore be assigned a higher weight. The inverse of the standard deviation ($W_i = 1/\sigma_i$) would be low in this case and is therefore assigned as the weight for this particular feature component.

- After computing the weight for all the feature components, they are normalised over the entire set of weights.

$$W_i = \frac{W_i}{\sum_{i=1}^{K} W_i} \tag{10}$$

4 Experimental Results

The performance of the model is measured by building an experimental prototype system. The image features used in the experiments are colour and texture. Colour is represented by computing the colour histograms and texture is represented using Gabor filters [5]. The objective of the experiments is to show that when the model is used to reduce the dimensionality of the image vectors, the retrieval performance does not decrease drastically. The image collection used contains 109 images which are in 24-bit colour. The image feature vectors are reduced to a dimensionality of 125 using the image model and the performance is measured. The dimensionality is further reduced steadily and the performance is measured at each level. Figure 3 shows the precision-recall values at varying dimensionalities. From the graph, it can be observed that the performance does not degrade significantly even when the dimensionality is reduced to a very low value. This is because the image model identifies the dominant components from the feature set and eliminates the less significant components during the dimensionality reduction process. The low dimensionality makes the image retrieval system very compact and therefore simplifies the indexing and retrieval process. It also improves the response time of the system as the search and retrieval time is reduced. The relevance feedback technique proposed allows the user to submit a query and then refine his/her information need via relevance feedback. This technique greatly reduces the user's effort of composing a complex query and captures the user's information more precisely. This further improves the performance of the image model as observed in Figure 4.

5 Summary

This paper presented a mathematical model to reduce the dimensionality of image vectors in an image retrieval system. The model uses the concept of

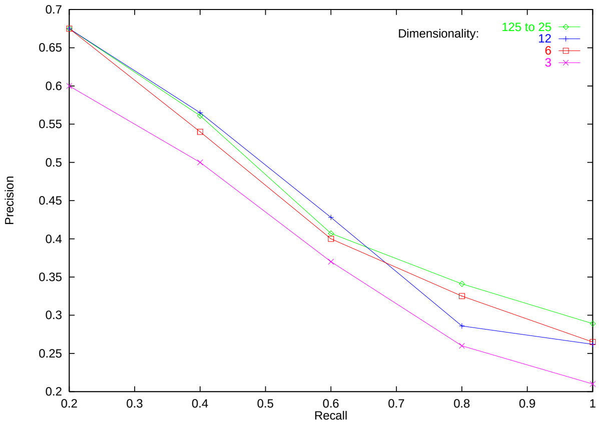


Figure 3: Precisio-recall values at varying dimensionalities

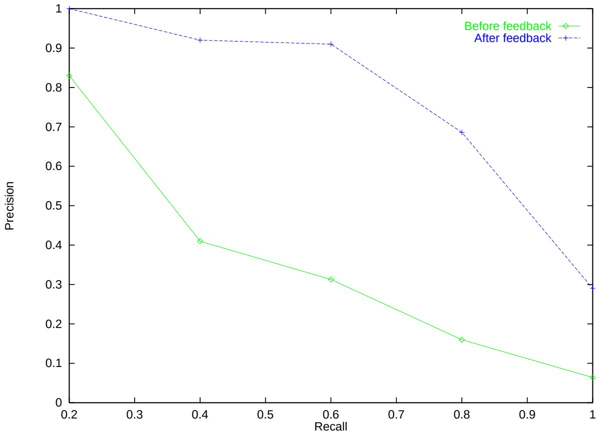


Figure 4: Precision-Recall values before and after feedback

eigen-vectors and eigen-values to identify the dominant components in the image vectors for a particular image data set. This allows the administrator to ignore the less significant values and make the model very compact. This improves the response time of the system as the indexing and retrieval process is simplified. The reduced dimensionality doesnot affect the retrieval performance. This has been demonstrated in the experimental results. The image model presented therefore provides a mechanism to improve the performance of image retrieval systems.

References

[1] Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. When is nearest neighbour meaningful. Tech. Rep. CS-TR-1998-1377, Computer Science Department, University of Wisconsin-Madison, Madison, WI 53706, 1998.

[2] Deerwester, S., Dumais, S., Furnas, G., Laundauer, T., and Harshman, R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, 6 (1990), 391–407.

[3] Guttman, A. R-trees: A dynamic index structure for spatial searching. *ACM Sigmod Record 14*, 2 (1984), 47–57.

[4] Haykin, S. *Neural Networks.* MacMillan Publishing, 1994.

[5] Jain, A. K., and Farrokhnia, F. Unsupervised texture segmentation using gabor filters. *Pattern Recognition 24*, 12 (1990), 1167–1186.

[6] Minka, T. P., and Picard, R. W. Interactive learning using a society of models. In *Proc. of IEEE Computer Vision and Pattern Recognition* (1996), pp. 447–452.

[7] Picard, R. W., Minka, T. P., and Szummer, M. Modelling user subjectivity in image libraries. In *Proc. of IEEE Conf on Image Processing* (Lausanne, Sep 1996). URL:http://vismod.media.mit.edu/vismod/publications/techdir/TR-382.ps.Z.

[8] Rui, Y., Huang, T., and Mehrotra, S. Relevance feedback techniques in interactive content based image retrieval. In *Proceedings of IS & T and SPIE Storage and Retrieval of Image and Video Databases* (San Jose, CA, 1998), pp. 25–36.

[9] Smith, J., and Chang, S. F. Visually searching the web for content. *IEEE Multimedia 4*, 3 (1997), 12–20.

[10] Smith, J. R., and Chang, S. F. An image and video search engine for the world wide web. In *Proceedings of of SPIE Storage and Retrieval for Image and Video Databases* (1997). URL:ftp://ftp.ctr.columbia.edu/CTR-Research/advent/public/papers/96/smith96g.ps.pz.

[11] Squire, D., Muller, W., Muller, H., and Raki, J. Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback. In *Proceedings of the* 11^{th} *Scandinavian Conference on Image Analysis* (Kangerlussuaq, Greenland, June 1999), pp. 143–149.