USING FEATURE SELECTION TO AID AN ICONIC SEARCH THROUGH AN IMAGE DATABASE

Kieron Messer *Josef Kittler*
University of Surrey, Guildford, Surrey, GU2 5XH, United Kingdom.
eep1km@ee.surrey.ac.uk

ABSTRACT

In this paper a method that facilitates an iconic query of an image/video database is presented. A query object is characterised by colour and texture properties. The same characteristics are computed locally for the database images. A statistical decision rule is then used to test for similarity between the iconically specified query and the database image descriptors. We show that by carefully selecting the set of descriptors the false alarm rate can be significantly reduced. The floating search feature selection method has been adapted to make it applicable to the hypothesis testing based query processing. The dimensionality reduction not only improves the performance but also enhances the computational efficiency of the method.

1. INTRODUCTION

Text and numeric databases have existed for many years. Well developed and very efficient database systems have been designed to facilitate rapid access. Many algorithms have been suggested to search through text and numeric data and many database languages have been developed to help in the efficient organisation and indexing of data. All this allows for a user to enter and retrieve information from their database at will, with ease and great speed.

However, as the price of computer storage falls rapidly, the number of large image databases is dramatically increasing. Examples of these can be found in the areas of medicine, science, defence and environmental applications. Images are a completely different data type and format to alpha-numeric data and new methods of organisation, indexing and querying must be devised to manage them well.

Current image database systems fall into one of two categories; text-based systems or image content-based systems. In general, when an image is entered into the database some data, be it text or numbers, is stored along with it which describes the content of that image.

In text-based systems an image is assigned keywords that describe its content. This system has many drawbacks. Humans are required to describe an image and enter this description into a computer. The vocabulary chosen to describe the image is restrictive and describing images involves a subjective evaluation. This type of indexing does not allow for similarity retrieval. Examples of such systems can be found in [1], [2] and [3].

A more intelligent way to solve this problem would be to get a computer to assign these keywords automatically. However, at present no algorithms have been developed that allow for a computer to understand an image at such a high level and it is unlikely that this situation will be resolved in the near future.

Many algorithms exist though that are able to capture low-level properties of images such as texture, colour, shape and size. Thus one can base image retrieval on indices coding such low-level image content. The advantage of this approach is that it can become a basis for a fully automated system relying upon little human interaction. Several such systems have been reported in the literature. Many image database systems under development base their indexing and retrieval algorithms on the colour properties of an image, these include [4], [5], [6], [7] and [8]. Other systems rely upon the shape attributes, these include [9] and [10].

Commercial systems available include IBM's QBIC, *Query By Image Content*, which allows users to index/query their image database using colour, texture, shape and position attributes, [11] and [12]. A demonstration of this system can be found on the WWW at *http://wwwqbic.almaden.ibm.com.* Virage, [13], is another commercial image database system which is also demonstrated on the WWW at *http://www.virage.com.*

However, these systems do not allow iconically specified queries and the image properties used are too coarse to facilitate very selective searches through a large database.

This paper describes an image database system that allows for such an iconic search through an image database. The user gives an example to the system of the object he/she is interested in. Colour and texture features on this object are then calculated. The same features have been pre-calculated on all the database images and stored. A search is then done which classifies the database features according to statistics derived from the specified object features. The regions in the images which were judged to be similar are then presented to the user as a result of the search.

Knowing what features to calculate for a query can be a problem. Calculating a set of texture features for a sequence of cartoon images, which carry no textural information, will not give good results. However, using textural information to identify different rock samples, which have a highly structured texture, can give excellent results. For most problems the best results will be obtained if a mixture of colour and texture features is used. However this opens the question which combination should be used?

It is well known that using as many features as possible not only results in a computational costly system but can also give worse results then if a sub-set of those features had been used. This is more commonly known as the peaking phenomenon.

In this paper a modified approach to feature selection is presented where the best set of features is selected for a particular query.

The rest of this paper is organised as follows: we begin in section 2 with the introduction of the database search. The feature selection process is explained in section 3. In section 4 we detail the experimental setup and findings made . Finally, in the last section, some conclusions are made.

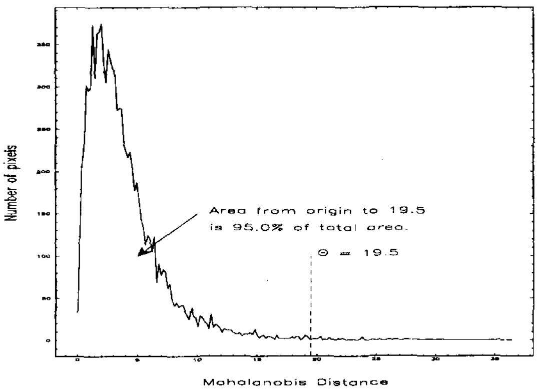


Figure 1. Typical histogram plot of Mahalanobis distance from all pixels in training region B to $\underline{\mu}_a$

2. THE DATABASE SEARCH

The database search proposed supports an iconic query. The user selects two regions A and B in a reference image. They are homogeneous in colour and texture (as perceived by the user) and representative of what the user is interested in finding in his/her image database.

Next various colour and texture representations are used to calculate n feature descriptors for both regions. These feature descriptors then form a feature vector $\underline{f}$ for every pixel, $\underline{f} = [y_1, y_2 ... y_n]^T$, where each y_i represents a different feature measurement.

It is assumed that the probability density function of the feature vectors for a homogeneous region follow an n-dimensional Gaussian. The feature distribution is completely described by its mean vector $\underline{\mu}$ and co-variance matrix, Σ.

If the same feature descriptors of an unknown pixel are calculated we can then statistically test it to see whether it belongs to the same cluster. In this system a measure known as the Mahalanobis distance is used for this as it gives a distance metric between the mean vector of the training region and a test vector, $\underline{f}_{test}$.

However, a threshold value is required to determine when to accept a pixel as belonging to the training cluster. The algorithm developed can be stated as follows.

- *Step 1* Calculate the mean vector $\underline{\mu}_a$ and covariance matrix Σ_a of region A.
- *Step 2* Calculate the Mahalanobis distance between every pixel in region B and mean vector $\underline{\mu}_a$ using $d = \left(\underline{f}_b - \underline{\mu}_a\right)^T \Sigma_a^{-1} \left(\underline{f}_b - \underline{\mu}_a\right)$. A histogram of these distance is then formed, see figure 1.
- *Step 3* From this plot the value which gives a user set percentage (around ninety-five percent) of the area under the graph from the origin is found. This value is the threshold value, Θ.

To search through an unknown image the Mahalanobis distance to $\underline{\mu}_a$ is calculated for every pixel in the test image. If this distance is less than Θ then that pixel is judged to be similar. A binary image is then produced for the user which shows all pixels that have been judged to be similar.

3. FEATURE SELECTION

It has long been a fundamental problem to identify which features are the best to use for a particular decision making task. Calculating hundreds of different features results in using a high dimensional feature space which can yield worse results than a lower dimensional feature space. High dimensionality will also increase the computational cost. If some features are not improving the results then to save time and expense these features need not be extracted for the same query on new images.

To solve this problem the process of feature selection is employed. A standard large set of texture and colour features are calculated for the query and then the best sub-set of these features is selected.

To perform feature selection a criterion function is required which evaluates the performance of a candidate feature sub-set. It is usual to use some inter-class separability measure such as divergence or entropy. In our problem only one class is statistically modelled, the class of interest therefore a new criterion function had to be developed in order to obtain this performance measure.

Three feature data sets, labelled A, B and T, are required. The first two data sets, A and B of size N_a and N_b respectively contain feature vectors extracted from the training object, $\underline{f}_{(a,i)}$ and $\underline{f}_{(b,j)}$ where $0 < i < N_a$ and $0 < j < N_b$. The third dataset, T of size N_t, contains feature vectors extracted from any background regions of the reference image which do not relate to the object of interest, $\underline{f}_{(t,k)}$ and $0 < k < N_t$. The performance of a feature selection set is then evaluated as follows:

- *Step 1* Get the sub-set of features for which you want to evaluate the performance.
- *Step 2* Estimate the mean vector and covariance matrix for the selected features of dataset A, i.e. $\underline{\mu}_a$ and Σ_a.
- *Step 3* Calculate the Mahalanobis distance $d_{(b,j)}$ for each feature vector, $\underline{f}_{(b,j)}$, in dataset B using
$$d_{(b,j)} = \left(\underline{f}_{(b,j)} - \underline{\mu}_a\right)^T \Sigma_a^{-1} \left(\underline{f}_{(b,j)} - \underline{\mu}_a\right)$$
- *Step 4* Form a histogram of these distances. Find the value which gives a user set percentage (around ninety-five percent) of the area under the graph from the origin. This value is the threshold value Θ. *Note: this percentage level remains constant throughout the entire feature selection process.*
- *Step 5* Calculate and store $d_{(t,k)}$ for each feature vector $\underline{f}_{(t,k)}$ in dataset T, where
$$d_{(t,k)} = \left(\underline{f}_{(t,k)} - \underline{\mu}_a\right)^T \Sigma_a^{-1} \left(\underline{f}_{(t,k)} - \underline{\mu}_a\right)$$
- *Step 6* Set an error count to zero, $\epsilon = 0$. For each $d_{(t,k)}$, where $(0 < k < N_t)$, compare $d_{(t,k)}$ with Θ. If $d_{(t,k)} < \Theta$ then increase the error count by one, $\epsilon = \epsilon + 1$. The final error count is the performance measure. The lower this count then the better the feature subset performance. The criterion function value is then defined by $J = 1 - \frac{\epsilon}{N_t}$

Now a criterion function has been chosen, feature selection is reduced to a problem that tries to detect the optimal feature subset from all possible feature sub-sets, i.e. the one with the lowest error count. This can be very time consuming as search algorithms, in general, are prone to combinatorial explosion. For this system the *sequential floating forward search* [14] was implemented which gives near optimal performance but with much lower computational cost.

4. THE EXPERIMENTS

Several experiments were performed to demonstrate the power of the proposed method for iconic query/search of an image database. Typical results are demonstrated by the following.

4.1. Experiment One

In this experiment many images from a *Tom and Jerry* video were grabbed from VHS video tape including figures 2(a) and (d). A total of 33 colour and texture features were computed and stored for each image. Next, three regions from figure 2(a) were selected. Regions 1 and 2 contained a homogeneous area of *Toms* blue coat. Region 3 was all the background area which was not part of *Toms* coat. Feature selection, as outlined in the previous section, was then carried out to find the optimal feature subset.

Two database searches were then performed on the images, one using the complete feature set (33 features) the other using the reduced feature subset (3 features). The classification for the region 2 data was set at 99.0% for both the database search and feature selection processes in order to find a suitable threshold.

Figures 2(b) and (e) show the similarity maps whilst using the complete feature set. The results are poor. The cat has been picked out but so have a lot of the background regions. Figures 2(c) and (f) show the similarity maps produced for each database image whilst using the reduced feature set. The results are excellent with the cat being correctly identified in all images. The improvements in the results are achieved even in spite of using less information.

4.2. Experiment Two

This experiment is the same as above but using images of a girl,*Kelly*, figures 3(a) and (d). Three regions of figure 3(a) were chosen, the first two comprising of areas of the *Kellys* stripy top. The third region being all other areas of this training image.

The optimal feature subset was then found using the feature selection process detailed earlier. A feature sub-set of size 9 was found to be optimal.

Figures 3(b) and (e) show the similarity maps whilst using the complete feature set. The results achieved are fairly reasonable. The top has been correctly identified in most images. Figures 3(c) and (f) show the similarity maps produced using the reduced feature set. There is an improvement over the previous results as more of the top has been recognised in the images.

5. CONCLUSION

In this paper a method for an iconic query to an image/video database was presented. It involved calculating both colour and texture features on the training region. The best subset of these features was then used for the database search. It was demonstrated that using as many features as possible did not necessarily give good results.

A novel method for feature selection was introduced that used a criterion function which only required the training class to be statistically modelled. We found that this criterion function performed promisingly selecting sensible features for each query task i.e. in the *Tom and Jerry* sequence colour features were selected as optimal. For the *Kelly* sequence a mixture a colour and texture features were selected.

This method of iconic query has many applications, such as medical, seismic and bore hole images, where specific regions of interest in images need to be identified. It is especially useful when one does not know what other classes are likely to be present in the database as this method selects features according to the classification performance on the training object. This means that the level of classification performance should increase when the object is present in different backgrounds. This is seen in experiment 2 with results improving in the second scene and not just in the training scene.

REFERENCES

[1] S Chang, Q Shi, and C Yan. Iconic indexing by 2-D strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3):413–428, may 1987.

[2] P Huan and Y Jean. Using 2D C+ strings as spatial knowledge for image database systems. *Pattern Recognition*, 27(9):1249–1257, 1994.

[3] S Tanimoto. Hierarchal picture indexing and description. In *IEEE 1980 Workshop on Picture Data Description Management*, pages 103–105, 1980.

[4] R Rickman and P Rosin. Content-based image retrieval using colour n-grams. In *IEE Colloquium on Intelligent Image Databases*, pages 151–156, Savoy Place , London, England, May 1996.

[5] M Stricker and M Swain. The capacity of color histogram indexing. *Proceedings of IEEE*, page 704, 1994.

[6] M Sato. A fast accessing method of color image. In *SPIE Applications of Digital Image Processing XV*, volume 1771, pages 519–526, 1992.

[7] M Swain. Interactive indexing into image databases. In *SPIE Electronic Imaging: Storage and Retrieval for Image and Video Databases I*, volume 1908. SPIE, February 1993.

[8] Y Gong et al. An image database system with content capturing and fast indexing abilities. In *IEEE International Conference on Multimedia Computing and Systems*, pages 121–130, May 1994.

[9] J Eakins, J Boardman, and K Shields. Retrieval of trade mark images by shape feature: The artisan project. In *IEE Colloquium on Intelligent Image Databases*, pages 91–96, Savoy Place , London, England, May 1996.

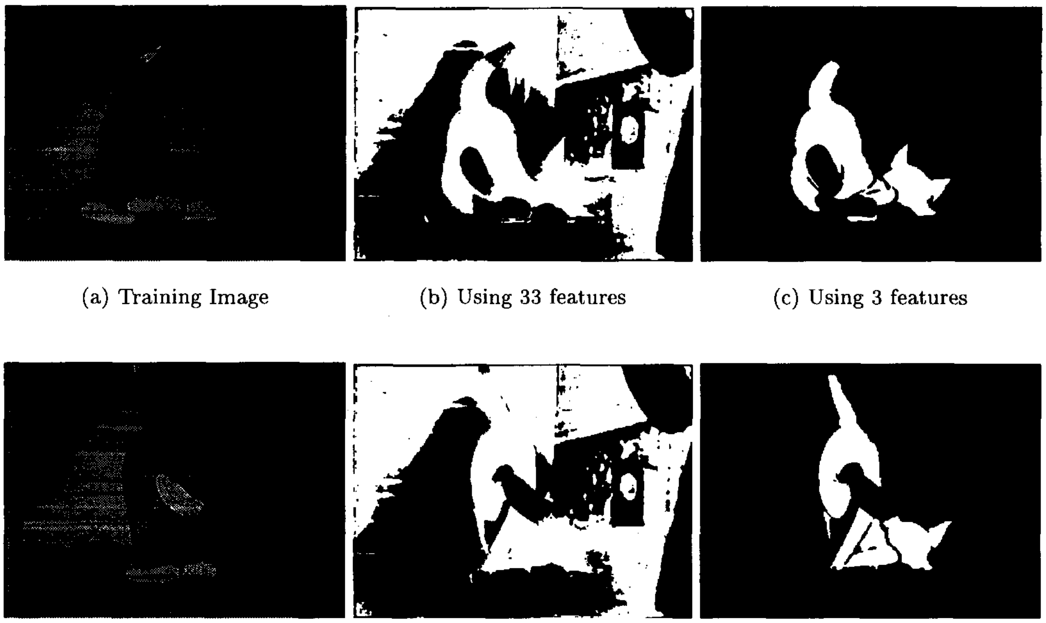
[10] F Mohktarian, S Abbasi, and J Kittler. Indexing an image database by shape content using curvature scale space. In *IEE Colloquium on Intelligent Image Databases*, pages 41–46, Savoy Place , London, England, May 1996.

[11] W Niblack and T Barber. Querying images by content using colour, texture and shape. In *SPIE Electronic Imaging: Storage and Retrieval for Image and Video Databases I*, volume 1908, February 1993.

[12] M Flickner et al. Query by image and video content: The QBIC system. *Computer*, 28(9):23–31, September 1995.

[13] J Bach et al. The virage image search engine: An open framework for image management. In *SPIE Electronic Imaging: Storage and Retrieval for Image and Video Databases IV*, volume 2670, pages 76–87, San Jose, California, February 1996. SPIE.

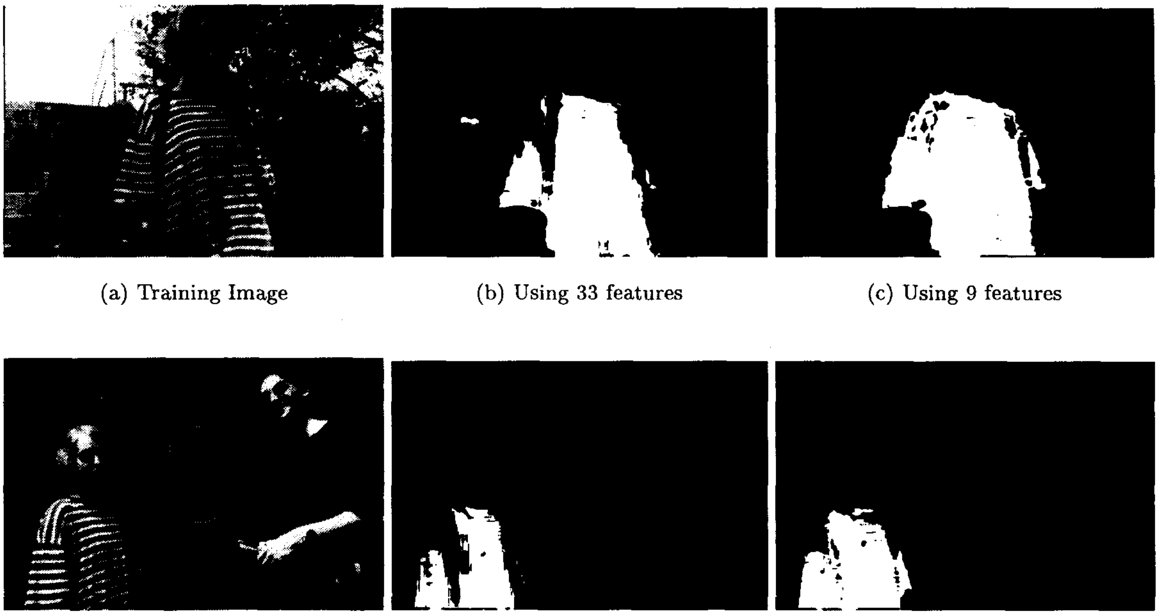
[14] P Pudil, J Novovicova, and J Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125, 1994.



(a) Training Image (b) Using 33 features (c) Using 3 features

(d) Test Image 1 (e) Using 33 features (f) Using 3 features

Figure 2. Experiment One: Searching for *Tom*.



(a) Training Image (b) Using 33 features (c) Using 9 features

(d) Test Image 1 (e) Using 33 features (f) Using 9 features

Figure 3. Experiment Two: Searching for *Kelly*.