From Sensory Coding to Scene Classification

Guoping Qiu[1], Jeremy Morris and Xunli Fan
School of Computer Science, The University of Nottingham

Abstract - **We present a storage-efficient and computationally fast method for rapid navigation/browsing through large image repositories and for content-based image retrieval. In the developed system, multiple resolution and orientation achromatic and opponent chromatic channels are sequentially encoded by a maximal information sensory encoding model, which conveniently and effectively indexes the images into a binary tree data structure. Content-based image retrieval, database navigation and image browsing are done very efficiently and rapidly by manipulating the n-bit binary keys in the binary tree data structure. We present experimental results to demonstrate the effectiveness of our method**.

1. INTRODUCTION

In conventional content-based image retrieval (CBIR) image database paradigm [1], image contents are represented by low level feature descriptors such as color histograms [2], texture characteristics [3] and "blobs" [4]. The descriptors are normally of very high dimension and image similarities are measured by the distances of their low level content descriptors. Retrieval is normally done by exhaustively searching the whole database and very few works, e.g., [5], have addressed the issues of efficient indexing and retrieval.

In this paper, we present a novel framework for efficient image indexing and fast retrieval. Our method has following innovative features. The images are first represented in a pattern color separable model in an opponent color space. The spatial pattern information is derived from the achromatic channel where most of the signal energy is concentrated and the color information is derived directly from the opponent chromatic signals. We then use a maximum entropy sensory coding strategy [13, 14] to code the achromatic spatial pattern channels and the chromatic channels in a sequential and hierarchical manner. Such a coding scheme readily renders a very efficient binary indexing tree data structure where images in the database can be represented with extreme efficiency, each image is indexed by an *n*-bit binary key, where *n* is the depth of the tree. Content-based or example-based image retrieval amounts to matching the *n*-bit binary keys of the querying image and that of the database images. Such an indexing scheme not only renders storage and searching efficiency, equally importantly, it also makes visual sense. An earlier version of this method has appeared in [15].

The organization of the paper is as follows. In the next section, we briefly review image representation and sensory coding which have provided the theoretical and scientific guidance for this work. In section 3, we present a sequential maximum entropy partitioning techniques for coding the images. Section 4 explains how such a maximal information coding technique can be exploited for efficient indexing and fast navigation through large image databases. Section 5 presents experimental results and section 6 concludes the paper.

[1] qiu@cs.nott.ac.uk

2. MAXIMAL INFORMATION SENSORY CODING

A well-known colour vision theory is the opponent colour theory [7], which suggests that there are three visual pathways in the human colour vision system. One pathway is sensitive mainly to light-dark variations; this pathway has the best spatial resolution. The other two pathways are sensitive to red-green and blue-yellow variation (the opponent channels). The blue-yellow pathway has the worst spatial resolution. In opponent-colour representation, the spatial sharpness of a colour image depends mainly on the sharpness of the light dark component of the images and very little on the structure of the opponent-colour image components. There is evidence to suggest that different visual pathways process colour and spatial pattern in the human visual system. The pattern-colour-separable model of human colour vision [6] suggests that the value of one neural image is the product of three terms. One term defines the pathway's sensitivity to the stimulus colour direction. A second term defines the pathway's sensitivity to the spatial patterns of the stimulus and the third term defines the pathway's sensitivity to the stimulus strength.

There are many studies that suggest that receptive fields of simple cells in mammalian primary visual cortex can be characterized as being spatially oriented and bandpass comparable to the basis of wavelet transforms [8, 9]. It is also a common practice in computer vision to represent the appearance of the objects by multiresolution directional steerable filters [10]. In this work, we use the following scheme to represent an image.

Assuming that the original data is in RGB color space, we first convert it into an opponent color space

$$Y = \frac{1}{3}(R+G+B), rg = R-G, yb = \frac{1}{2}(R+G)-B \quad (1)$$

Y is then filtered by oriented Gaussian derivative filters, $g(\sigma, \theta)$ of scale σ and orientation θ (other multiresolution representations such as wavelet can also be used)

$$Y(i,j) = \left| Y * g(\sigma_i, \theta_j) \right| \quad (2)$$

And finally, an image I is represents as

$$I = \left(Y, \{Y(i,j)\}, rg, yb\right) \quad (3)$$

Once an image is represented with schemes such as (3), each component will have a clear visual interpretation. For example, $Y(i, j)$ relates to the surface roughness at a particular scale and orientation, rg relates to the red-green component and yb represents the yellow–blue component of the image. For example, Figure 1 shows the color appearances in the opponent chromatic space. What we would like to do is to exploit this intuitive visual interpretation of the representation.

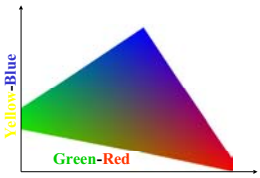

Figure 1, Color appearances in the red-green (*rg*) and yellow-blue (*yb*) chromatic space.

Now that the images have been represented at multiple resolutions and orientations in an opponent color space, what process should be done in the next stage in order to recognize the scene and objects? There have been many research works study sensory coding and natural scene statistics. The efficient coding hypothesis predicts that individual neurons should maximize information transmission [13, 14]. Although we are not interested in the hypothesis itself, the information maximization coding principle may be "borrowed" for indexing large image database.

With an image represented in multiple channels by (3), we can then use the information maximization principle to code each of the channels. How this principle can be practically exploited will be explained in the next section, we here look at what does information maximization means.

Suppose we have a signal $\boldsymbol{s}$, to be coded by an encoder as $\boldsymbol{C}(\boldsymbol{s})$ and the response of the encoder is limited within some bonded intervals. In order to convey the maximal information, it is straightforward to show that the distribution of the encoder response must be uniformly distributed within the bonded interval. According to Shannon information theory [12], information content is measured as entropy:

$$H(X) = \sum_{\forall X} P(X)\ln(P(X)) \qquad (4)$$

$H(X)$ reaches maximum when $P(X)$ is a uniform distribution. Therefore, maximal information and maximum entropy (maxent) are equivalent.

3. SEQUENTIAL MAXIMAL INFORMATION CODING

To exploit the maximal information encoding principle for image indexing and retrievals turns out to be less straightforward than at first sight. In the human visual system, the multiple channels in (3) could well be coded in parallel based on the maximal information principle. However, in engineering practice, it is very difficult to compute information of multidimensional random variables because of the difficulty in estimating the probability density functions in high dimensional spaces. To get round this computational difficulty, we simplify the model by assuming (not necessarily make biological sense) that the channels are coded independently in a sequential manner and each follows a maximal information principle.

Each component in (3) is still of very high dimension (image height by image width). To simply the problem further, we only model the first order statistics by considering pixels in each of the component independently, that is, we collapse each component into a sequence of scalar values. In this case, a histogram of the pixel distribution in each of the channels can be easily compiled. For a scalar random variable, constructing a maximal information encoder with limited alphabets becomes a simple task. All is needed is to ensure that the pixel populations distributed to each alphabet is identical.

With such a simplified model, we can encode the components in (3) independently. Each component is then again treated as multiple instances of a scalar random variable. A scalar encoder that conveys the maximal information for each channels can then be easily constructed.

4. FAST SCENE CLASSIFICATION

How the maximal information encoders such as those described in section 3 can be used for image database indexing and retrieval? Here is a scheme we have developed which has been found to work very well (see next section). In our developed scheme, a binary tree data structure is constructed by encoding the components in (3) in a hierarchical order as illustrated in Figure 2.

Starting from level 0, for a pre-selected component C_0, we find a value that cuts all pixels in this component from all images in the database into two equal population halves, this value is sometime known as the median of the pixels. It is important to note that at this initial node, all pixels from all images in the database participate in finding this median value. Note also that to code the component into two alphabets, the maximal information coding dictates that the pixels must be equally distributed into the two alphabets. Once we have found the median value M_0, we then divide the images in the database into two clusters each placed in one of the two child nodes. Those images whose C_0 component has more pixel values greater than M_0 are classified to the right child node and those images with more pixel values smaller than M_0 are put into the left child node. The two child nodes of the root node form level 1 of the hierarchy. At level 1, images are indexed by a 1-bit key. All images in the left child node will be indexed as 0-value and all images in the right child node will be indexed as 1-value. The component used by the node and the median value of that component are stored with the tree.

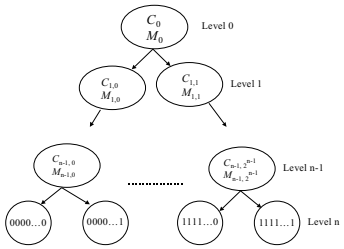

Figure 2, Binary tree indexing of images based on maximal information sequential visual coding

In any non-leaf node in the tree, the component of the node (pre-defined and will be explained more in the next section) and its median value computed from all images falling into the node, are again used to divide the images in this node into its left and right child nodes. For images classified into the left child node, a 0-value bit is appended to their key and for images classified into the right child node, a 1-value bit is appended to the images key. Again information about the component used in the node and its median value are stored with the tree.

For an n-level tree so constructed, all images are indexed by n-bit binary keys. The keys between images in a parent node and its immediate two child nodes differ by one bit.

Once the tree has been constructed, new images can be easily put into the tree. All we have to do is to compare the median values of the various components of the image corresponding to those orders used in the tree to assign the image into appropriate nodes,

and the image will be indexed with the keys of the nodes it belongs to.

Query-by-example follows the same procedures as that for adding an image to the database. One the binary key of the query image has been computed, there are considerable flexibility in retrieval, we can either return all images in the leaf node the query image falling into, or images at the lower level nodes of the tree the querying image belongs to (more returns).

Navigating through the tree is also easy, we can transverse the tree by going up and down the nodes. At any particular node, we could output the images at the node for visualization and then decide to see more images (by going to the parent node) or see fewer images (by going to one of the chide nodes).

From storage's point of view, each image has only n-bit key extra information to store. For the tree itself, we only need to store information of the component used in each non-leaf node and the component's median value of all (training) images. Therefore the overhead of our system is very small. We are not aware of any CBIR image database management system has the efficiency of our method. Because we actually index our images, which is unlike many existing methods for CBIR, where much side information in the form of image content descriptors, such as color histograms, will have to be stored, and image retrieval is done by linearly searching the image database, our method is much more efficient. In the next section, we will present experimental results to demonstrate the effectiveness of our method.

5. EXPERIMENTS

We have tested our method using a very large image database. We have tried both image database browsing and navigation, and content-based image retrieval. In our experiments, we only use the original of the achromatic channel and chromatic channels. For nodes at the same level, the same component has been used. The orders of the channels used are randomly picked.

5.1 Navigating/Browsing Image Databases

We have implemented a navigation/browsing tool using Java. The user interface of the simple tool is shown in Figure 3.

Figure 3, User interface of a simple tool for browsing/navigating image database.

On the left, there is a panel which displays random images from the database. The users can pick images from this panel as querying examples. Once an image is selected from this panel, it will be displayed on the upper right hand corner above the result display panel. Simultaneously (and instantly on a 2.66 GHz Pentium Processor PC and 10,000-image database), all images in the leaf node that has the same binary key as the querying image are displayed on the display panel. Two navigation buttons allow the users to transverse the binary tree. Images at the current node are displayed on the display panel. The querying image can also be picked up from the display panel. Obviously at this stage, the tool is very crude, and more functionalities will be added on later.

Figure 4, Navigating the database using an example image with red flowers in green plants backgrounds. Top: leaf node images. Middle: Parent node one level above the leaf node. Bottom: Parent node 2 levels above the leaf node.

Figure 4 shows results of an example round of navigation through a 10,000 image database which has been indexed into an 11 level binary tree as described in this paper. Of course, at this stage,

there is no intelligence built in the system, and we are currently trying to make the system smarter. One of the major advantages of our systems is its simplicity and efficiency, which may compensate the accuracy.

5.2 Image Retrieval Experiments

In a second experiment, we tested the methods retrieval accuracy performances. In this experiment, we used 60,000 images in our database. There are two test sets. The first set consists of 168 pairs of images. Examples of these pairs are shown in Figure 5. We then put the 168 pair images in the database. The purpose is to see whether the pairs of images will have the same key. If so, then imagine using one of the pairs as querying image, it will directly find the image cluster of the corresponding target. If the cluster is of small size, then finding the target should be made much easier. Therefore, the more pairs there are in the same nodes, the better would be the performance for this particular test. Results are shown in Table 1. The longer the keys are, the smaller are the sizes of image clusters in the leaf node, hence the chance of the same pairs falling into the same node is decreased. It is seen that even using an 11-bit key (on average the leaf nodes will contain clusters of 30 images and of course they are unevenly distributed), 111/168 pairs managed to stay together at some leave nodes. Reduce the key length to 9 bits (average leaf node cluster contains 117 images), there are 162/168 pairs managed stay together.

Set A Set B

Figures 5, Examples of image pairs. For an image in set A, there is one corresponding image in set B. The corresponding images form a pair of querying and target images.

Table 1. Pairs of images with the same leaf node keys

Key length	11 bits	10 bits	9 bits
# pairs with identical keys	111	156	162

In another test, we used 120 classes of color texture images. Each class consists of 6 similar images. Examples of these are shown in Figure 6.

Table 2. Texture classes images with the same leaf node keys

Key length	11 bits	10 bits	9 bits
# pairs with identical keys	92	115	120

Results of this second set test data is shown in table 2. It is seen that the results are very good.

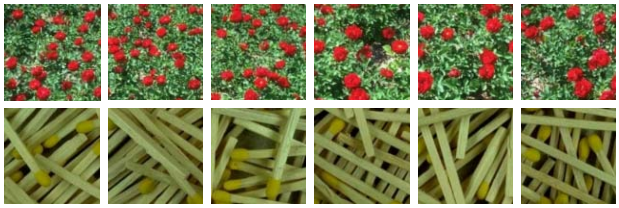

Figure 6, Example color texture images in a second test data set.

6. CONCLUDING REMARKS

In this paper, we have presented an effective and efficient method for fast navigating large image repository and for content-based image retrieval. The advantage of our method is its efficiency and simplicity. We have further presented experimental results to demonstrate the effectiveness of our method. We believe our technique can be developed into really useful tools for large image repository management.

REFERENCES

[1] A. W. M. Smeulders et al, "Content-based image retrieval at the end of the early years", IEEE Trans PAMI, vol. 22, pp. 1349 - 1380, 2000

[2] M. Swain and D. Ballard, "Color indexing", International Journal of Computer Vision, Vol. 7, pp. 11-32, Year 1991

[3] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data", IEEE Trans Pattern Analysis and Machine Intelli-gence, vol. 18, pp. 837 – 842, 1996

[4] C. Carson, S. Belongie, H. Greenspan and J. Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 1026 – 1038, 2002

[5] J. Y. Chen, C. A. Bouman, and J. C. Dalton, ``Hierarchical Browsing and Search of Large Image Databases," IEEE Trans. on Image Processing, vol. 9, no. 3, pp. 442-455, March 2000.

[6] A. Poirson and B. Wandell, "Appearance of colored patterns: pattern-color separability", J. Opt. Soc. Am. A, vol. 10, pp. 2458 – 2470, 1993

[7] P. K. Kaiser and R. M. Boynton, Human Color Vision, Optical Society of America, Washington DC, 1996

[8] B. A. Olshausen and D. J. Field, "Emergence of simple cell receptive field properties by learning a sparse code from natural images", Nature, 381, pp. 607 – 609, 1996

[9] S. G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol II, 7, pp. 674-693, 1989

[10] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters", PAMI (13), No. 9, pp. 891-906, 1991

[11] IEEE Abstract. IEEE Top Reference. BibRef 9109

[12] T. M. Cover, Elements of Information Theory, Wiley, 1991

[13] E. P. Simoncelli and B. A. Olshausen, "Natural image statistics and neural representation", Annual Review of Neuralscience, vol. 24, 2001

[14] S. B. Laughlin, "A simple coding procedure enhances a neuron's information capacity", Z. Naturforsch, 36c, pp. 910-912, 1981

[15] G. Qiu, L. Ye and X. Feng, "Fast image indexing and visual guided browsing", **CBMI 2003**, Third International Workshop on Content-Based Multimedia Indexing, September 22 - 24, 2003 IRISA, Rennes, France