



PERGAMON

Available at  
[www.ElsevierComputerScience.com](http://www.ElsevierComputerScience.com)  
POWERED BY SCIENCE @ DIRECT®

Pattern Recognition 37 (2004) 2177–2193

**PATTERN  
RECOGNITION**

THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

[www.elsevier.com/locate/patcog](http://www.elsevier.com/locate/patcog)

# Compressing histogram representations for automatic colour photo categorization

Guoping Qiu\*, Xia Feng<sup>1</sup>, Jianzhong Fang

*School of Computer Science, The University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, UK*

Received 12 September 2003; received in revised form 8 March 2004; accepted 8 March 2004

## Abstract

Organizing images into semantic categories can be very useful for searching and browsing through large image repositories. In this work, we use machine learning to associate low level colour representations of digital colour photos with their high level semantic categories. We investigate the redundancy and performance of a number of histogram-based colour image content representations in the context of automatic colour photo categorization using support vector machines. We use principal component analysis to reduce the dimensionality of (high dimensional) histogram based colour descriptors and use support vector machines to learn to classify the images into various high level categories in the histograms subspaces. We present experimental results to demonstrate the usefulness of such an approach to organizing colour photos into semantic categories. Our results show that the colour content descriptors constructed in different ways perform quite differently and the performances are data dependent hence it is difficult to pick a “winning” descriptor. Our results demonstrate conclusively that all descriptors studied in this paper are highly redundant and that regardless of their performances, the dimensionalities of these histogram based colour content descriptors can be significantly reduced without affecting their classification performances.

© 2004 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

*Keywords:* Color histogram; Content-based indexing and retrieval; PCA; SVM

## 1. Introduction

Fast advancement in digital imaging technology has resulted in the exponential increase of image data in both professional archives and personal leisure collections. Effectively managing large image repositories and making them easily accessible poses significant technical challenges. In the past decade, there has been significant research effort in content-based image retrieval (CBIR) [1]. In a CBIR system, a user can query the image repositories with a visual example and the system will return an ordered list of images that are similar to the query in some visual sense. Traditionally, image similarity is measured by some forms of

distance metrics in the feature space. However, similarity is a subjective concept, it is therefore not surprising and perhaps inevitable that there is a “gap” between a similarity measured by the CBIR systems and a similarity perceived by human observers. How to reduce this gap, often referred to as the “semantic gap” in content-based image retrieval has received much attention in recent years, and technologies that provide solutions to reduce the semantic gap of CBIR are likely to play a pivotal role in content-based image indexing and retrieval.

One direction pursued by researchers to narrow the semantic gap is based on the “learning by example” principle. Machine learning is an integral and essential part of human efforts to build intelligent machines, and the subject has been studied extensively in various disciplines of scientific and engineering pursuits. In the context of CBIR, machine learning has been applied to classify collections of images into categories or classes of various descriptions [2]. In particular, image classification techniques based on various

\* Corresponding author. Tel.: +44-115-8466507;  
fax: +44-115-951-4254.

*E-mail addresses:* [qiu@cs.nott.ac.uk](mailto:qiu@cs.nott.ac.uk) (G. Qiu),  
[xxf@cs.nott.ac.uk](mailto:xxf@cs.nott.ac.uk) (X. Feng), [jzf@cs.nott.ac.uk](mailto:jzf@cs.nott.ac.uk) (J. Fang).

<sup>1</sup> On leave from The Civil Aviation University of China and sponsored by the China Scholarship Council.

machine learning approaches have been shown to be particularly useful in providing relevant feedback in CBIR [1,3–5] to boost the CBIR systems usability and performance. The essential idea behind these approaches is to use machine learning to capture or learn the high level concept of visual similarity (as perceived by human observers) from examples of similar images chosen by humans. In any machine learning system, an essential issue is the numerical (low-level) representation of the objects to be learned. Different representation schemes may affect many aspects of a machine learning system, including computational complexity and performance.

Colour is one of the most popular, and arguably the most effective low-level cue for content-based image retrieval. Histogram based colour representations, despite their many weaknesses, are the most often used numerical representation schemes for the colour contents of digital images and many forms of histogram-based colour representation have been proposed in the literature [1,6–10]. However, histograms are normally of very high dimension and are therefore computationally expensive for a CBIR system. Recently, it has been shown that colour histograms can be significantly compressed without affecting much the performance of content-based image retrieval [8]. In this work, we investigate the redundancy and performance of a number of histogram-based colour image content representations in the context of automatic colour photo categorization using machine learning. Five histogram-based colour content representations are studied. We use principal component analysis (PCA) to reduce the dimensionality of these (high dimensional) histogram colour descriptors. Support vector machine (SVM) is used to learn to categorize images in the histogram eigenspace. Our results show that a combination of low level colour descriptors and machine learning can be quite useful in categorizing images into semantic classes, however, they also show the inadequacy of current technology in bridging the “semantic gap” of content-based image retrieval in particular and the difficulty of associating low level descriptors with high level concepts in general. It is found that different descriptors have discernible classification accuracy. It is also found that the categorization performances not only are representation dependent, but also are data dependent, thus making it difficult to pick an overall “winning” colour descriptor. However, it is found conclusively that in the context of color photo categorization, histogram-based descriptors are highly redundant and their dimensionality can be reduced significantly without affecting the performances. For example, reducing the dimension of some descriptors by as much as 70% causes no deterioration in performance, and reducing the dimension of some descriptors by over 90% only causes about 2% drop in classification performance.

The organization of the paper is as follows. In Section 2, we briefly review related work. Section 3 gives a brief description of 5 histogram based colour image content de-

scriptors studied in this paper. Section 4 briefly summarizes PCA for data compression. Section 5 briefly describes support vector machine. Section 6 presents our experiment setting. Section 7 present experimental results and Section 8 concludes the paper.

## 2. Related work

Organizing images into semantic categories can be extremely useful for searching and browsing through large collections of images. Recently, various methods have been proposed for this purpose. Very high dimensional color histograms and SVMs have been used to classify color photographs into various categories [2]. SVM has also been used to perform relevant feedback in content based image retrieval [3,4]. AdaBoost algorithm [11] has been used for natural image retrieval [12]. Both AdaBoost learning and SVM were used in Ref. [13] as relevant feedback mechanism for natural image retrieval. The author in Ref. [14] has presented empirical results of boosted image classification.

Projecting image features onto sub spaces for classification and recognition has been investigated by a number of authors in various applications. In Ref. [15], subspace method has been used for image classification. PCA has been used to project vectors onto eigenspace for face recognition [16]. In Ref. [8] the authors have successfully applied PCA to compress color histogram for content-based image indexing. PCA has been used in a feature selection technique to reduce feature dimensionality in an SVM based face detection system [17].

In relevant feedback [28], the user interacts with the system which has to re-learn a classification model based on the user’s feedback. It is therefore very important that the system can perform real-time learning. One way to achieve this is to use low dimensional feature representations to reduce the computational complexity. In the literature, this is a largely ignored problem. In Ref. [2], feature vectors of 4096 components were used for image classification. Feature vectors of 435 components were used in Ref. [13] for relevant feedback. The authors in Ref. [4] used color and texture features of 144 components. Such high dimensional vectors not only make re-learning and re-classification computationally expensive, more importantly, they will also make the system suffer from the so called “peaking phenomenon” in pattern recognition [18,19]. If the training samples used to design the classifier are small relative to the number of features, the system can be easily overtrained. In order to build good classifiers, the size of the training samples should be an order of magnitude larger than the dimensionality of the feature vector. That is, if the feature vector is of 435 dimensional, the training sample size should at least be 4350. In the case of relevant feedback in CBIR, the training samples are picked by the user on line at an interactive rate, and therefore it is very unlikely a large relevant/irrelevant samples are available for training. Given that the size of training

samples is small, the representation features should have low dimension in order to build good classifiers. Based on these considerations, it is highly desirable to have representation features of low dimensionality for relevant feedback application in a CBIR system. This provides the motivation of the current work: we seek to find representation features of low dimensionality without sacrificing classification accuracy.

### 3. Histogram-based color image content descriptors

Many image content descriptors have been proposed in the content-based image indexing and retrieval literature [1,6–10]. In this paper, we examine five histogram-based descriptors. This section gives a brief description of each.

#### 3.1. Opponent color histogram

Color histogram is one of the earliest image content descriptors used in content-based image indexing and retrieval [1]. A color histogram can be constructed in a 3d color space. However, the dimensionality of such a histogram is often very high. Researchers have found that it is possible to construct a 2d chromaticity histogram which will often give equal performance to 3d ones [8].

In [8], it was shown that the red–green and blue–yellow opponent chromaticity space decorrelates color information well and the 2d color histograms constructed in this space are also more uniformly distributed hence more compressible. The opponent chromaticities of red–green ( $rg$ ) and blue–yellow ( $by$ ) are defined in terms of  $r$ ,  $g$ ,  $b$  chromaticities:

$$(rg, by) = \left( r - g, \frac{r + g}{2} - b \right). \quad (1)$$

The opponent color histogram investigated in this paper is constructed in the  $rg$  and  $by$  space according to Eq. (1).

#### 3.2. Color correlogram

One of the well-known problems of color histogram is that it contains no spatial information which has greatly limited its discriminative power. Recently researchers have developed a better and more discriminative technique known as the color correlogram ( $CC$ ) [9], which has been shown to provide significant improvement over the original color histogram approach.  $CC$  is very similar to co-occurrence matrix developed some 20 years ago for grey scale texture classification. Formally, the  $CC$  of image  $\{Z(x, y), x = 1, 2, \dots, M, y = 1, 2, \dots, N\}$  is defined as

$$CC(i, j, k) = \Pr(Z(x_1, y_1) \in C_i | Z(x_2, y_2) \in C_j), \quad (2)$$

$$k = \max\{|x_1 - x_2|, |y_1 - y_2|\},$$

where the original image  $Z(x, y)$  is quantized to a fixed number of colors  $C_1, C_2, \dots, C_L$  and the distance between the two pixels  $k = 1, 2, \dots, K$  is fixed a priori. In other words, the  $CC$  of an image is the probability of joint occurrence

of two pixels some  $k$  distance apart that one pixel belongs to color  $C_i$  and the other belongs to color  $C_j$ . The size of  $CC$  is  $O(L^2K)$ . To reduce storage requirement, [9] concentrated on auto-correlogram whereby  $i = j$  and its size is of  $O(LK)$ . We also only investigate the auto-correlogram in this paper. We used 64 colors ( $L = 64$ ) and 4 distances ( $K = 4$  and  $k = \{1, 3, 5, 7\}$ ) as suggested in [9] to build the auto-correlogram. The colors,  $C_1, C_2, \dots, C_{64}$ , were obtained via vector quantization (VQ) [20]. The training samples used to create these colors were all pixels from two hundred and thirty five  $512 \times 512$  pixels, 24 bits/pixel true color texture images from the MIT VisTex<sup>2</sup> collection. These same 64 colors were also used for layered color indexing described in Section 3.5.

#### 3.3. MPEG-7 color structure descriptor

The MPEG-7 color structure (CS) descriptor is a color feature descriptor that captures both color content (similar to a color histogram) and information about the structure of this content [10]. Instead of considering each pixel separately, the extraction method embeds color structure information into the descriptor by taking into account all colors in a structuring element of  $8 \times 8$  pixels that slides over the image. This descriptor is more discriminative than color histogram because it can distinguish between two images in which a given color is present in identical amounts but where the structure of the groups of pixels having that color is different in the two images. MPEG-7 standard specifies that color values are represented in the double-coned HMMD color space, which is quantized non-uniformly. In the standard [21], there are detailed descriptions of how the color structure descriptor can be constructed and readers are referred to Ref. [10] and Ref. [21] for full details.

#### 3.4. Colored pattern appearance histogram

The colored pattern appearance model (CPAM) which has two channels capturing the characteristics of the chromatic and achromatic spatial patterns of small image regions has been used to compile content descriptors for image retrieval [6]. In this method, the visual appearance of a small image block is modeled by three components: the stimulus strength, the spatial pattern and the color pattern. In Ref. [6], the YCbCr space is used, and the stimulus strength  $S$  is approximated by the local mean of the  $Y$  component. The pixels in  $Y$  normalized by  $S$  form the achromatic spatial pattern (ASP) vector. Because  $C_b$  and  $C_r$  have lower bandwidth, they are sub-sampled. The sub-sampled pixels of  $C_b$  and  $C_r$  are normalized by  $S$ , and then concatenated together to form the chromatic spatial pattern (CSP) vector. Vector quantization (VQ) [20] is used to encode the ASP and CSP vectors. Let  $Q_c$  with a codebook  $C_c = \{C_c(j); j \in M\}$  be the

<sup>2</sup> <http://vismod.www.media.mit.edu/vismod/imagery/Vision-Texture/vistex.htm>

VQ quantizer for the CSP vectors and  $Q_A$  with a codebook  $C_A = \{C_A(i); i \in N\}$  be the VQ quantizer for the ASP vectors. Typically, the sizes of the codebooks range from 64 to 256 work well. The VQ codebooks were also trained using MIT VisTex images [6]. The probabilities that the CSP vectors are coded by the  $i$ th codeword of  $C_C$  and the probabilities of the ASP vectors are coded by the  $j$ th codeword of  $C_A$ , for all possible  $i$  and  $j$  form the colored pattern appearance histogram (CPAH)

$$\text{CPAH} = \{\{\Pr(Q_A(\text{ASP}) = i), \forall i\}, \{\Pr(Q_C(\text{CSP}) = j), \forall j\}\}. \quad (3)$$

Both CPAH and MPEG-7 color structure descriptors are designed to take into account the spatial distributions of colors. However, MPEG-7 CS only counts the colors within a mask and how the colors are spatially distributed within the mask is ignored. CPAH takes this one step further by capturing the spatial chromatic and achromatic patterns of a neighborhood of pixels, it therefore captures not only color, but also texture characteristics of the images as well [6].

### 3.5. Layered color indexing

From human perception's point of view; different frequency components of the visual stimulus may be treated differently by the visual system. Physically, different frequency components of the signal may correspond to different objects or object parts. Consequently, different frequency components of the visual stimulus may not only have different perceptual significance, they may also correspond to different physical objects. Incorporating such concept is helpful to the development of more effective image retrieval techniques. In Ref. [7] a layered color indexing method was developed to represent colour contents of images. An image is first passed through a filter bank (each filter of the filter bank covers a specified spectral bandwidth). The output of the filter bank is used to classify the pixels. Pixels in an area with similar spatial frequencies are then retained on the same layer. Each individual layer, which contains only those pixels in areas with similar frequency distributions, is used to form its own color index. The aggregation of the feature indices from all the layers then forms the overall index of the image.

Let  $H_k$  be the color histogram of layer  $L_k$ , then layered color indexing (LCI) histogram is formed as

$$H(x) = \{H_k\}, \forall k. \quad (4)$$

That is, multiple color histograms for an image, each taking colors from pixels in areas with similar sharpness are compiled. This approach not only indexes the color, but also associates color distributions with their surface roughness. It has been argued that such an association not only has perceptual significance, it also makes sense physically. It has been shown that the LCI method significantly enhances the power of color indexing and at the same time retains its simplicity and elegance.

In practice, classification of the pixels into layers can be done by thresholding the Laplacian image of the Y (achromatic) channel of the image. We found that dividing the pixels into 4 layers and using 64 colors worked well [7]. The 64 colors used for this method are the same as those used for the auto-correlogram described in Section 3.2.

## 4. Data compression using PCA

PCA, also known as Karhunen–Loeve Transform (KLT) [22] is a well-known classical statistical method which has been widely used in data analysis and compression. The basic idea of PCA is to find the  $n$  linearly transformed components  $\{s_i, i = 1, 2, \dots, n\}$  so that they explain the maximum amount of variance in the input data. In practice, PCA is done by calculating the  $n$  largest eigenvectors of the (sample) covariance matrix  $\Sigma = E(\mathbf{x}^t \mathbf{x})$ , where  $\mathbf{x}$  is the  $m$ -dimensional sample vectors.

In our current application, the basic goal in PCA is to reduce the dimension of the data. Thus one usually chooses  $n \ll m$ . Indeed, it can be proven that the representation given by PCA is an optimal linear dimension reduction technique in the mean-square sense. Such a reduction in dimension has important benefits. First, the computational overhead of the subsequent processing stages is reduced. Second, noise may be reduced, as the data not contained in the  $n$  first components may be mostly due to noise. Third, a projection into a subspace of a very low dimension is useful for visualizing the data.

In the literature, PCA has been successfully applied to dimensionality reduction in many fields of scientific and engineering pursuits, e.g., [8,23]. Since the dimensionality of typical histogram descriptors are quite high which will result in high computational cost in distance calculation, we use PCA to reduce the dimensionality of histograms described in the last section before they are presented to the classifier. Our goal is to investigate the amount of redundancy existing in each of these descriptors so that we can develop computationally efficient methods using lower dimensional features without sacrificing classification accuracy performances.

At this point, it is appropriate to mention other widely used data compression techniques, such as discrete cosine transform (DCT) and wavelet transform (WT) [24]. Unlike KLT, the bases of these transforms are data independent. We argue here that these transforms are in general unsuitable for our current tasks for the following reasons (as will also be demonstrated in the experimental results, Section 7.3).

Firstly, in order to be useful, we want the energy of the signal to be concentrated in as few transformation coefficients as possible so that we can throw away those coefficients containing little energy. A compression method such as DCT or WT may compress the histograms very well in terms of the number of bits it used to represent the data, but this does not necessarily help our case if the energy of the signal is distributed widely among the transformation

coefficients. Because our goal is not data compression, instead our purpose is to achieve dimension reduction. These two are not always the same.

Secondly, both DCT and WT (and similar transforms) perform frequency analysis. One of the reasons that these transforms are so successful in image compression is that most image signals tend to change slowly, i.e., a pixel and its neighbors tend to have similar values. In other words, most of the energy is concentrated in the low-frequency coefficients, we can thus throw away high-frequency coefficients without affecting too much the quality of the reconstructed signal. In our histogram data, the neighboring bins do not necessarily correlate as neighboring pixels do. Fig. 1 shows the five histograms of a typical image and it is seen that there are a lot of abrupt changes from one bin to the next (although treating the *rg-by* opponent histogram as an image tends to make neighboring bins correlate better and render the histogram more compressible [8]). As a result, the energy of the signal will be distributed amongst all transform coefficients making it difficult to use a few of these coefficients to preserve most of the energy of the original signal.

It is clear that a transform (compression technique) will only be useful in our current application if it can concentrate the signal's energy in a few coefficients thus enabling dimensionality reduction. As argued above, the nature of DCT and WT, and the properties of the histogram data, make DCT and WT unsuitable for our application (in order to make the histograms more compressible by DCT or WT, the histogram bins will have to be ordered in such a way that neighboring bins correlate with each other. We will come back to this point in Section 7.3). On the other hand, KLT can do exactly what is required: most of the signal's energy is concentrated in the first few of its transform coefficients.

## 5. Support vector machines

Support vector machine [25,26] is an emerging machine learning technology that has been successfully used in a variety of pattern recognition tasks. We shall consider SVMs in the binary classification setting. We are given training data  $\{x_1 \dots x_n\}$  that are vectors in some space  $x \in \mathbb{R}^m$ . We are also given their labels  $\{y_1 \dots y_n\}$  where  $y_i \in \{-1, 1\}$ . SVMs perform pattern recognition for two-class problem by finding the decision boundary (a hyperplane) that has the maximum distance to the closest points in the training set. Training instances that lie closest to the hyperplane are called *support vectors*. All vectors lying on one side of the hyperplane are labeled as  $-1$ , those on the other side are labeled as  $1$ . Fig. 2 illustrates a simple linearly separable problem in 2d space.

If the data are not linearly separable in the input space, SVMs allow us to project the data in space  $x$  to a higher dimensional feature space  $F$  via a Mercer kernel operator  $K$  [26]. In other words, we consider the set of classifiers of

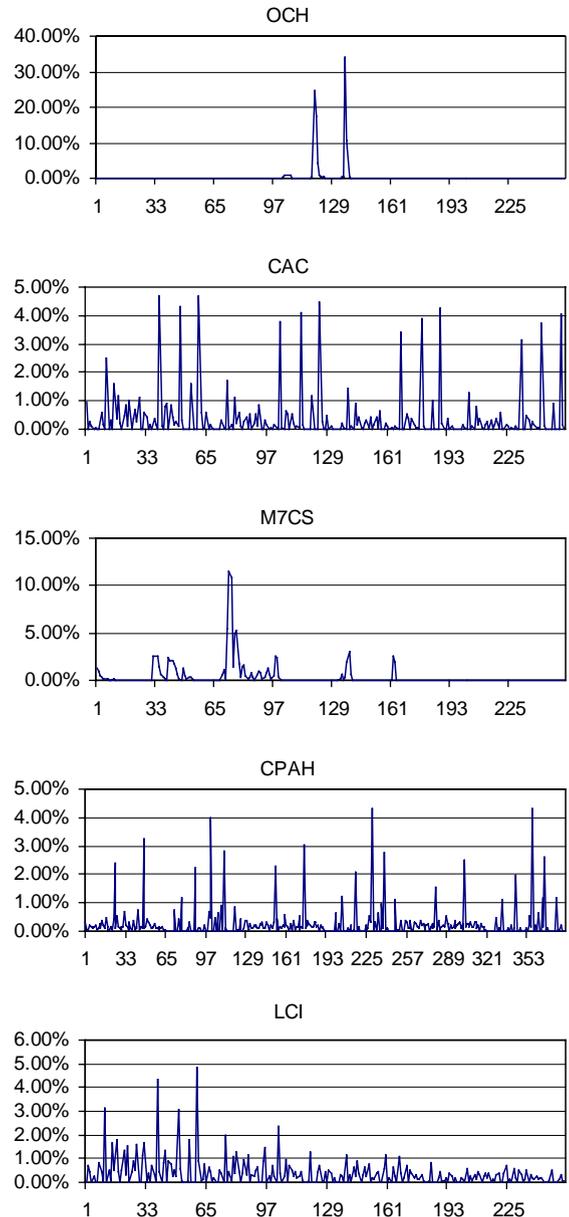


Fig. 1. The five histograms of a typical image. It is seen that the neighboring data points do not necessarily correlate.

the form

$$f(x) = \sum_{i=1}^n \alpha_i K(x_i, x). \quad (5)$$

The SVM computes the  $\alpha_i$ s that correspond to the maximal margin hyperplane in  $F$  space. By choosing different  $K$ , the SVMs can project the training data in  $x$  into  $F$  for which hyperplanes in  $F$  correspond to more complex decision boundaries in  $x$ . For two-class problems, the decision

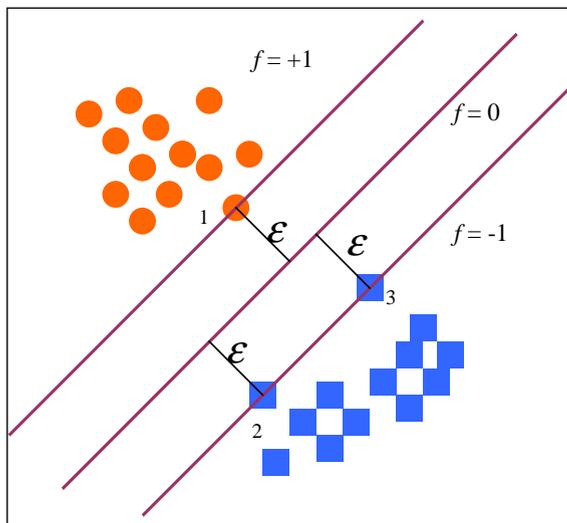


Fig. 2. A simple support machine. The optimal hyperplane maximizes the distance to the closest training points. These points (1, 2, and 3) are called the support vectors (SVs). The distance  $\varepsilon$  between the hyperplane and the SVs is called the margin.

is made according to: when  $f(x) \geq 0$  we classify  $x$  as  $+1$  (class 1); otherwise we classify  $x$  as  $-1$  (class 2).

SVMs are designed for binary classification [25,26]. When dealing with several classes, such as in image classification, one needs an appropriate multiclass method. Our approach is one per class, also known as one against the others, which compares a given class with all the others put together [2]. In the one against the others algorithm,  $N$  hyper planes are constructed, where  $N$  is the number of classes. Each hyperplane separates one class from the other classes. In this way, we get  $N$  decision functions,  $f_k(x)$ ,  $k=1, \dots, N$ . The class label of a new sample  $x$ ,  $L_x$ , is given by

$$L_x = \arg \left( \max_{\forall k} (f_k(x)) \right), \quad (6)$$

i.e., the class with the largest decision function.

## 6. Learning image classification using SVM in histogram subspaces

We investigate the classification performance and compressibility of the various histogram-based image content descriptors using a system set up as shown in Fig. 3. From an input image, we derive its histogram descriptors, which are then passed to the PCA module to reduce their dimensionality. The compressed histogram features are then fed to the SVM classifiers which then output the identity of the input image. Let  $\psi_i$  be the  $i$ th eigenvector of the co-variance matrix of the  $m$ -dimensional image descriptor  $x$ . Let  $s = \{s_i, i=1, \dots, m\}$  be the projections of  $x$  onto the eigenspace,

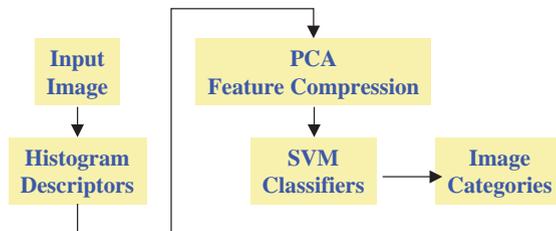


Fig. 3. Image categorization experiment set up.

and  $s_i = x\psi_i$ . It is well known that the vector formed from the first few  $s_i$ ,  $s' = \{s_i, i=1, \dots, n\}$ ,  $n < m$ , can be used to reconstruct an approximation to  $x$ . We can therefore use a lower dimensional vector  $s'$  as input to the SVMs for classifying the input images. We investigate how the selection of  $n$  affects the classification performances. Obviously, the smaller the value of  $n$ , the higher the compression ratio and the cheaper it is to compute the categories by the support vector machine.

## 7. Experimental results

### 7.1. Data set A and results

In this first test, we use 10 categories of color photographs from the Corel color photo collection. These categories are labeled by human. The categories are: Lions, Elephants, Tigers, Horses, Sky scenes, Cheetahs, Eagles, Night Scene, Sunset and Roses. Each category consists of 100 color photos. Fig. 4 shows examples of typical images from each of these categories.

At this point, it is appropriate to make some observations about the data and the adequacy of colour cue alone for its content description. The images in each category represent a particular theme—a high-level concept. For example, in the animals categories, such as Lions, Cheetahs and Elephants, each image will contain one or more of these animals. However, the animals may only occupy a small section of the image and the background may occupy most of the image area. The backgrounds of images within the same category may vary significantly, and to make the task more difficult the backgrounds of an image in one category may be very similar to those of images in another category. Images in the same theme category may have different colours as well. For example, in the Roses category, some Roses are yellow and some are red. A number of different categories, such as Eagles, Sky Scenes, Elephants, Tigers, Cheetahs all have images containing large part of blue sky scenes. Clearly, colour alone will be inadequate to distinguish among these categories. To be able to represent the images better, other low-level features, such as texture descriptor, and medium/high level object shape descriptor will be necessary. However, medium and high level descriptors are difficult to obtain and the topic is still being actively

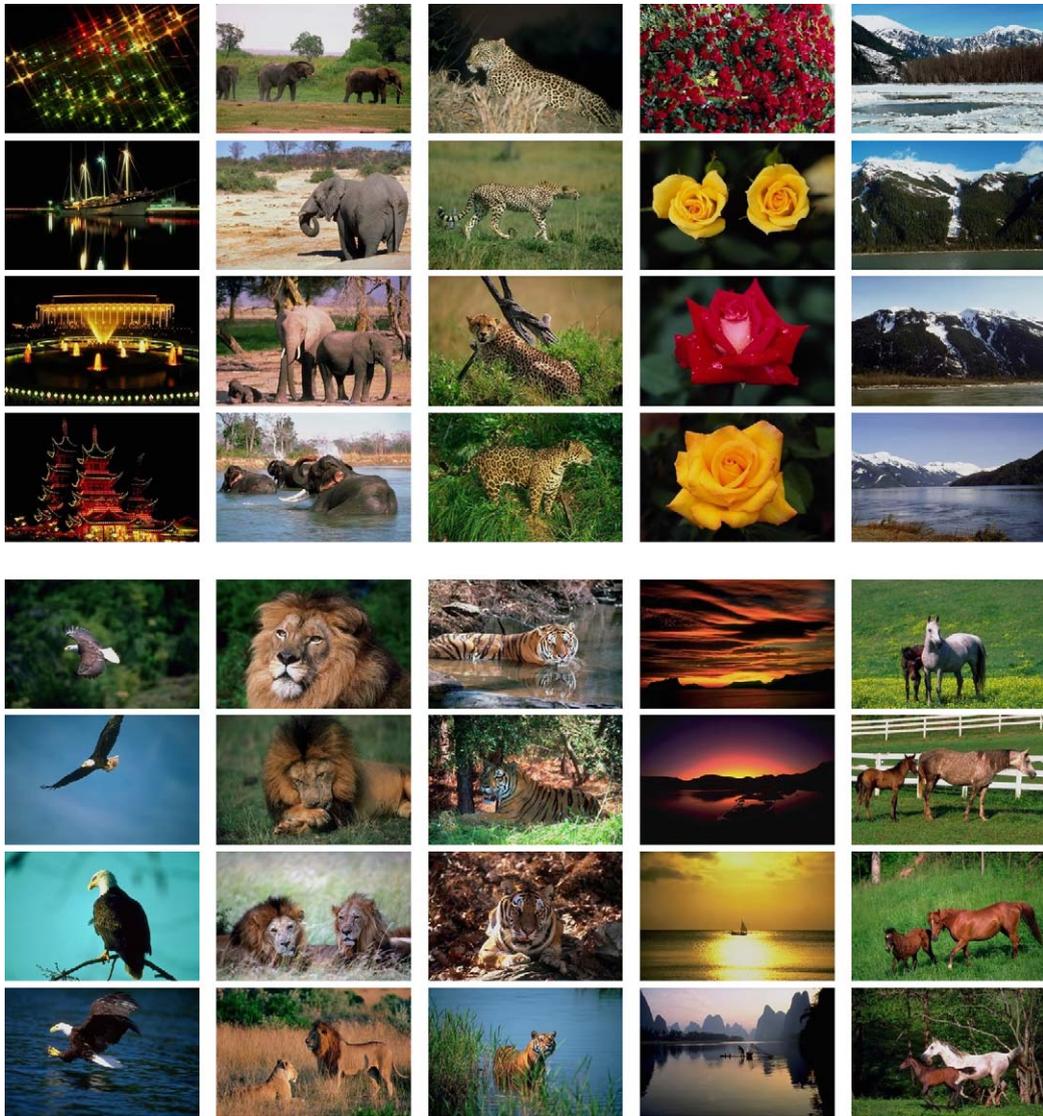


Fig. 4. Example images from the 10 categories (4 typical images from each category are shown here) used in the experiment.

researched. Even with more comprehensive low-level features and some medium- and high-level features available, the task of automatically classifying these images according to the categories as perceived by humans is still an extremely challenging task. This data set illustrates why the “semantic gap” exists and why it is so difficult to bridge such gaps. To be able to close this gap, it will require much effort in many fields, including computer vision, machine learning, psychology and many other related fields. At this stage, it is hard to see how such a gap can be completely closed.

From the above observations, it is very clear that we cannot expect to achieve 100% correct classification using the colour descriptors of Section 3 and that a complete solution will require much more comprehensive representation

schemes and more sophisticated learning strategies than that illustrated in Fig. 2. Our goal is instead to investigate how the various colour content descriptors constructed in different ways will perform in this setting with an aim to aid the effort of finding better solutions in the future. We also want to investigate the redundancies that may exist in various forms of histogram-based colour content representations with the aim of finding more compact colour descriptors.

For each class, we randomly select 50 images as the training samples and use the other 50 for testing. For each of the 5 types of histograms described in Section 3, one co-variance matrix is constructed using the 500 training samples to calculate the eigenvectors for feature reduction. For each category, we train an SVM for each type of histogram descriptor.

Table 1

Correct classification performances for different descriptors trained using different penalty factors (results are for training data only)

$C$	OCH (%)	CAC (%)	M7CS (%)	CPAH (%)	LCI (%)
1	68	65	80	70	76
1.4	68	66	80	72	77
1.8	69	66	81	73	79
2.2	70	66	81	73	78

In all experiments, we used a 256d opponent color histogram (OCH), a 256d color auto-correlogram (CAC), a 256d MPEG-7 color structure histogram (M7CS), a 384d colored pattern appearance histogram (CPAH) and a 256d layered color indexing (LCI). In constructing the OCH, the histogram bins were obtained by dividing the  $rg$  and  $by$  axes each into 16 equal intervals. The CAC histograms were constructed using 64 colors and 4 distances as described in Section 2.2. In building the CPAHs,  $N = 256$  achromatic bin and  $M = 128$  chromatic bins were used. The codebooks were again trained using the two hundred and thirty five  $512 \times 512$  pixels, 24 bits/pixel true color texture images from the MIT VisTex collection [6]. The M7CS histograms were constructed by following the standard [21]. In constructing the LCI histograms, the same 64 colors used for the CACs were used here and 4 layers were constructed as described in Ref. [1].

For support vector machine training, we follow John C. Platt's sequential minimal optimization (SMO) implementation [27]. Because we do not expect these photo categories can be completely separated by the SVMs, we train the machines follow the imperfect separation scheme. The penalty factor  $C$  which determines the degree of imperfection ( $C = \infty$  corresponds to a perfect separation) has to be set manually (please see Ref. [27] for SVM implementation details). In order to investigate how sensitive the performances are to this parameter, we train the SVMs with different values of  $C$ . We measure the classification performance by the overall correct classification rate (CCR) defined as

$$CCR = \frac{\text{Total-Number-of-Correctly-categorised-Photos}}{\text{Total-number-of-Testing-Photos}}. \quad (7)$$

Table 1 shows the CCR performances of various uncompressed histograms when the penalty factor  $C$  was set to various values. It is found that the correct classification performances do not change significantly with this value. It is also found setting  $C$  any larger do not improve the performance. From this table, we see that M7CS gives the best performance. It is also interesting to note that the number of supporting vectors for this descriptor is three times as large as those of other four descriptors. This may explain the good performance of the M7CS descriptor. It would be very useful and interesting to understand the significance of this correlation.

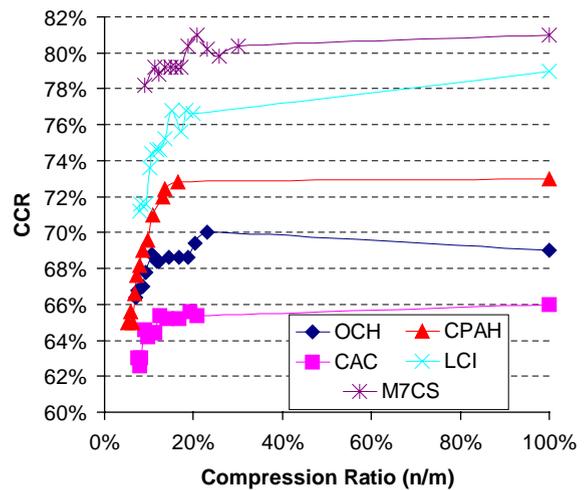


Fig. 5. Data set A correct classification rates for different descriptors for different compression factors.  $m$ —the dimensionality of the raw histogram and  $n$ —the dimensionality of its eigen subspace vector used by the SVMs for categorization.

Now that we have an idea how large the training penalty factor should be, we choose the one that performs well ( $C = 1.8$ ) for the rest of our experiments. We investigate how the amount of compression affects the classification performances. Various factors of compression are tried, and results are shown in Fig. 5. These results clearly demonstrate that different representations do give quite different performances and that these descriptors are all highly compressible. It is found that reducing the dimension of some descriptors by 70% does not affect the performance accuracy at all. By reducing some descriptors dimension by 90% we only notice 2% drop in performance.

It is also seen that these curves are not entirely monotonic, in some cases, fewer coefficients resulted in slightly better CCRs. Since the fluctuations were quite small, we believe this was caused mainly by numerical errors. A high compression ratio may sometimes remove noise better thus resulting in slight better performance. The overall trend is that, for compression ratios up to 70%, the CCR performances changed very little. But, for compression greater than a certain value (e.g.,  $> 90\%$ ), the CCR performances almost dropped monotonically.

Table 2

Summary of correct classification performances of different categories using different histograms before compression and after compression

Histograms										
Categories	OCH		CAC		M7CS		CPAH		LCI	
	$n/m = 1$	$n/m = 0.2$								
Lions	33	33	33	36	35	36	40	40	37	37
Elephants	33	33	31	32	34	34	21	20	40	39
Tigers	19	19	18	14	40	41	20	19	31	29
Horses	46	46	47	47	48	47	48	47	44	44
Sky	44	44	46	46	46	46	48	48	48	49
Cheetahs	26	24	11	11	29	29	23	23	21	18
Eagles	30	30	33	33	39	36	33	33	40	40
Night	44	41	38	38	39	39	41	41	41	42
Sunset	29	30	36	34	43	41	42	41	38	35
Roses	38	39	36	35	46	46	45	45	46	45

The figures are the number of correctly classified images in each category (results are for testing data only).

In Appendix, we tabulate the confusion matrices for these descriptors both before compression and after compression. It is seen that regardless of the performances of different descriptors, confusion matrices for the same descriptor change very little before and after dimension reduction, again indicating the highly compressible nature of these raw descriptors.

Table 2 summarizes the correct classification performance of different categories when different colour descriptors are used. It is seen that for most categories, the methods actually perform quite well with very high percentage of the images being classified into the correct categories. There are two categories, Tigers and Cheetahs, that are difficult to categorize. In the case of Tigers, M7CS descriptor works quite well, but all other descriptors perform very poorly. For the Cheetahs category, all representations do not perform well. From these results, it is fair to say that the success of such a scheme for the categorization of colour photos is mixed, in the sense that for some categories, the performances are quite good, whilst for other categories, the performances are poor. Overall, we feel that such a scheme could be useful in automatic color photo categorization.

Fig. 6 shows some typical misclassified images. In fact, all these misclassified images are explainable. For example, one Elephant image contains a large portion of sky, therefore classifying it into sky category cannot be regarded as totally wrong. All these examples demonstrate once more the difficulty of automatically associating low-level representations with high-level concepts. This difficulty typifies the huge challenges for all the fields related to machine intelligence—computer vision, image understanding, pattern recognition, machine learning, etc

## 7.2. Data set B and results

We have seen that for data set A, the images within a category can be different and images from different cate-

gories can be similar. In this second test data set, we used data classified as similar and different by human observers to test the method. We manually classified the first 6400 images from the Corel collection into four groups, each contained visually similar images, those images cannot be put into any of these groups were thrown away. The sizes of these four groups were 399, 354, 541 and 392 images, respectively. The thumbnails of these four groups of images are shown in Fig. 7 (not all images are displayed). It is seen that the visual contents of images within the same group are quite homogeneous and images in different groups are quite different.

We constructed five histogram descriptors for this data set in exactly the same way as for data set A. We then randomly picked 150 images from each group to develop the classifiers and the KLT bases and used the rest images for testing. The experimental procedure was the same as in Section 7.1 for data set A.

Fig. 8 shows how compression affects the performances. Without any compression, all descriptors performed very well. As can be seen from Fig. 7, images were judged similar mainly because of their color themes are very similar. Given that all descriptors extract color information in one way or another, these good performance results are not entirely surprising. We also see that LCI gave the best result; OCH and CPAH performed similarly, and M7CS and CAC did similarly well. What is also striking for this data set is that the amount of compression can be achieved without affecting the CCR performances. It is seen that compressing LCI to only 7% of its original dimension did not affect its performance at all (although not shown in Fig. 8, we noticed that CCR started to drop gradually for compression ratios  $n/m < 7\%$  for the LCI method). We again see the occasional minor fluctuation of the CCR/compression ratio curve, we believe this was again caused by numerical errors and can be explained similarly as the case in Fig. 5. These results suggested that if we want to classify images into groups of

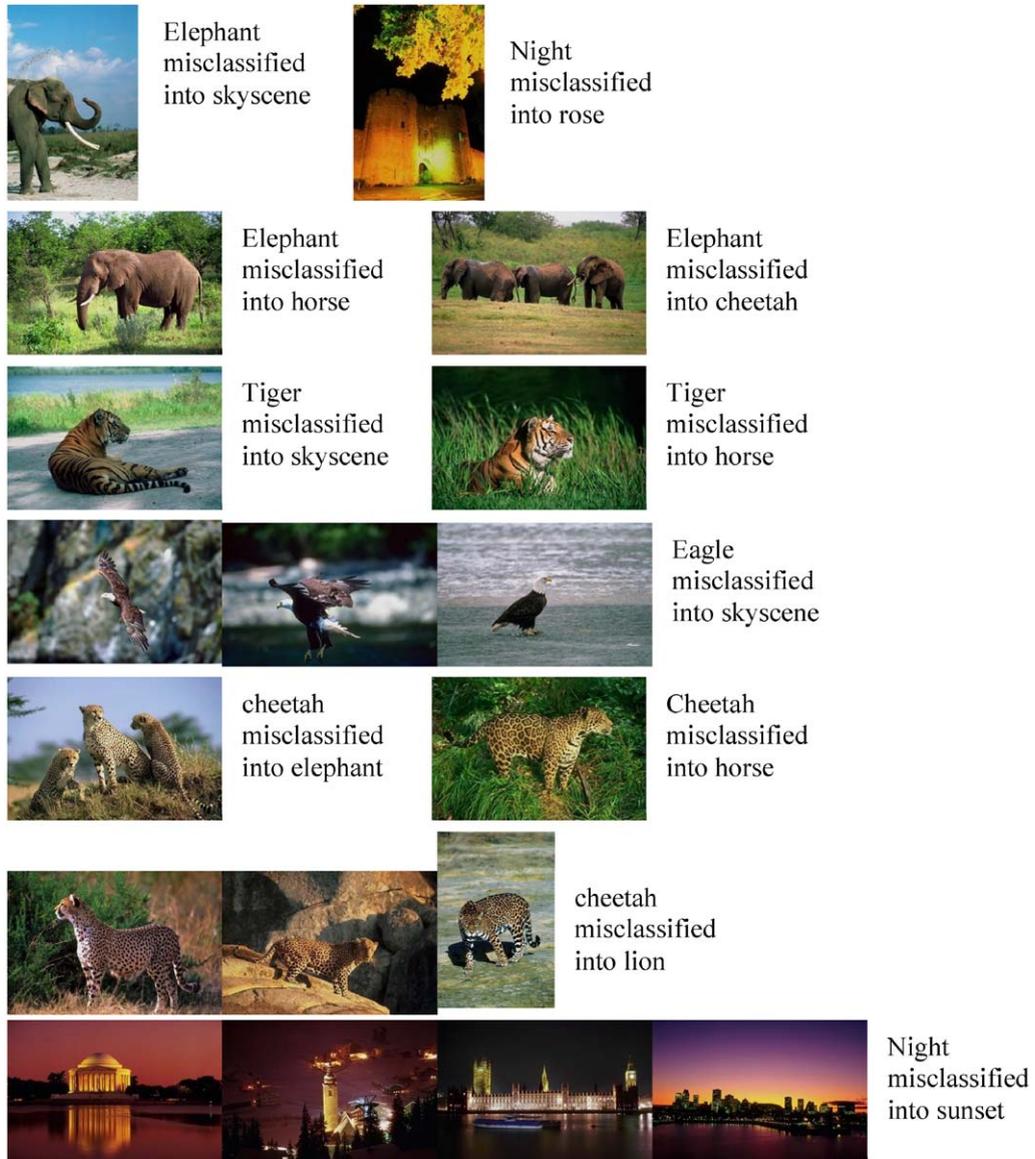


Fig. 6. Examples of typical mis-classified photos.

similar colour themes, then the descriptors could be compressed even harder.

### 7.3. Other compression schemes

We have implemented a WT method for reducing the dimensionality of our histogram features. We used the Haar wavelet [24] and our implementation was based on the following pseudocode:

```
proc WT(H: array[0..m- 1])
  H  $\leftarrow$  H/ $\sqrt{m}$ 
```

```
while m > n do:
  m  $\leftarrow$  m/2
  for i = 0 to m - 1 do
    H'[i]  $\leftarrow$  (H[2i] + H[2i + 1])/ $\sqrt{2}$ 
    H'[m+i]  $\leftarrow$  (H[2i] - H[2i + 1])/ $\sqrt{2}$ 
  end for
  H  $\leftarrow$  H'
end while
```

**end proc**

where H holds the histogram features before and after the transform,  $m$  is the original dimensionality of H and  $n$  is the



Fig. 7. Thumbnails of four visually similar groups of images in data set B.

reduced dimensionality. Although we should choose the  $n$  largest elements from the transformed  $H$  for classification, it cannot easily achieve dimensionality reduction because the largest  $n$  WT transform coefficients may fall into different locations for different images. In our implementation, we simply choose  $H[0 \dots n - 1]$  as the compressed histogram feature for classification, and for computational convenience, we restricted  $n = m/2^k$ ,  $k$  is an integer.

This was so in order to perform the test meaningfully and to be comparable to the way KLT was used in these applications.

We performed the test on both data set A and data set B. Results of the CCR performances are shown in Table 3. It is seen that in general WT performed far worse than KLT. It is also interesting to observe that the M7CS descriptor is more compressible by the WT as compared with other four de-

Table 3  
Comparison of the overall CCR performance of KLT and WT for dimensionality reduction

		Overall CCR				
		OCH (%)	CAC (%)	M7CS (%)	CPAH (%)	LCI (%)
Data set A	KLT	61	56	79	71	76
( $n/m = 25\%$ )	WT	40	41	67	54	54
Data set B	KLT	96	93	95	95	97
( $n/m = 12.5\%$ )	WT	79	56	92	68	60

The overall CCR of a particular method was obtained by averaging the CCRs over all the categories for the method (results are for testing data only).

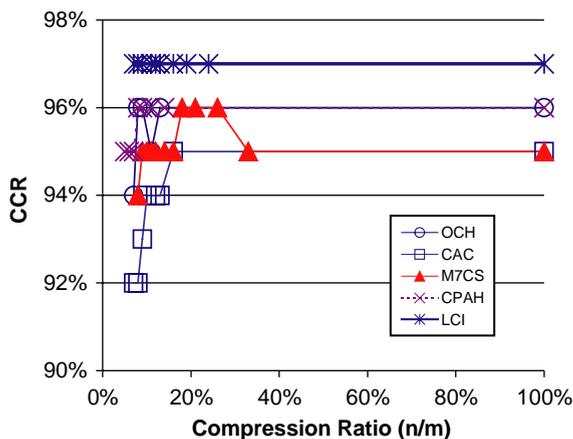


Fig. 8. Data set B correct classification rates for different descriptors for different compression factors.  $m$ —the dimensionality of the raw histogram and  $n$ —the dimensionality of its WT compressed vector used by the SVMs for categorization.

scriptors, especially for test data set B. According to the standard [21], the 256 color bins are quantized non-uniformly as follows: (1) the HMM color space is divided into five subspaces along the *diff* axis by dividing the *diff*-axis into the following five intervals: [0,6), [6, 20), [20, 60), [60, 110) and [110, 255]; (2) each subspace is uniformly quantized along the *hue* and *sum* axes, where the number of quantization levels along each axis is defined for each interval. The ordering of the bin numbers is first from bottom to top (parallel to the *sum*-axis), then from *diff*-*sum* plane to *diff*-*sum* plane (around the *hue*-axis) staying within a subspace, and finally from subspace to subspace. For detailed description of this numbering scheme, please refer to Ref. [21]. As a result, the neighboring bins were correlated and this explains why the M7CS descriptor was more compressible by WT. For other descriptors, the bin numbers were assigned without any specific ordering. In order to make these descriptors more compressible by data independent linear transforms such as WT, the bins have to be ordered in such a way that

neighboring bins correlate with each other. The question of how to order the bins of the histograms such that neighboring bins correlate well with each other, thus making the descriptors compressible by linear transform such as WT or DCT, is not addressed in the current paper.

#### 7.4. Result summary

These experimental results have demonstrated conclusively that for all the descriptors studied in this paper, each can be compressed significantly by KLT (PCA) without sacrificing the correct categorization performances. This is especially useful if such a scheme is used in an interactive setting for relevant feedback in a CBIR system where realtime (or very fast) online learning and reclassification of results are highly desirable. From these results we know that we can use colour descriptors that are of much lower dimensionality than those used in the literature, e.g. [2,4,13], by previous researchers. Indeed, one can simply use the PCA approach in this paper to derive lower dimensional descriptors for fast relevant feedback computation. Since dimensionality reduction is what we want to achieve, other data independent transform schemes suitable for general data compression such as DCT and WT are in general not suitable for this application. In order to make the histogram compressible by DCT or WT, the bins have to be ordered such that neighboring ones correlate with each other.

From these results, we can also see that, even though we understand the limitations of color cues in CBIR and semantic image categorization, they are still very powerful and useful cues. Color cues are easy to derive and simple to use compared to other representations. For the results of data set A, although M7CS achieves the best performance overall, it is not the case for all categories. For example, for the Night Scene category, OCH is in fact the best performer. For data set B, LCI has a slight edge overall (although the maximum difference was only 2% between different descriptors). Therefore, to a certain extent, the performances are (inevitably) data dependent, which indicates the need for the co-existence of various different methods to derive color descriptors.

## 8. Concluding remarks

In this paper, we have investigated five histogram-based image descriptors for automatic digital color photo categorization using support vector machines. We have shown that different descriptors have different performances. The performances of different representations are also data dependent, thus it is not easy to pick an overall winning representation which indicates the need for the co-existence of various different colour descriptors. Despite the many limitations of colour cues, our results demonstrate once again that colour can be a very powerful cue for image classification. A combination of colour descriptors and machine learning can be used to build useful systems for color photo categorization. It is also found that all these histograms are highly redundant and can be significantly compressed without affecting the classification performance. Histograms are a descriptor used extensively in image content management

and they normally have very high dimensionality. From the results of this study, we conjecture that there may exist lower dimensionality alternatives to the existing high dimensional histogram-based image content descriptors, it may well worth pursuing these lower complexity alternatives. One possibility is to use the technique used in this paper, i.e., PCA, to reduce the histograms' dimensionality. Other possible methods, for example, re-ordering the histogram bins to make neighboring ones correlate well with each other, thus making the histograms compressible by data independent linear transforms such as DCT and WT, may well warrant further investigation.

## Appendix

Tables 4–13, confusion matrices of test data set A, before and after compression. All results are for testing data only.

Table 4  
Confusion matrix for the OCH descriptor without compression

Output categories		OCH	Lions	Elephants	Tigers	Horses	Sky	Cheetahs	Eagles	Night	Sunset	Roses
Input categories	Lions		33	8	0	1	3	2	2	1	0	0
	Elephants		3	33	1	2	4	6	0	0	0	1
	Tigers		9	2	19	6	6	3	1	3	1	0
	Horses		0	0	0	46	0	0	1	0	0	3
	Sky		0	1	0	0	44	0	5	0	0	0
	Cheetahs		6	6	3	2	4	26	2	1	0	0
	Eagles		1	1	3	4	10	1	30	0	0	0
	Night		0	0	0	0	0	0	0	44	4	2
	Sunset		12	2	1	0	1	0	0	1	29	4
	Roses		0	1	0	4	0	1	0	2	4	38

Table 5  
Confusion matrix for the OCH descriptor, compression ratio ( $n/m$ ) = 20%

Output categories		OCH	Lions	Elephants	Tigers	Horses	Sky	Cheetahs	Eagles	Night	Sunset	Roses
Input categories	Lions		33	8	0	1	3	2	2	1	0	0
	Elephants		3	33	0	2	4	7	0	0	0	1
	Tigers		9	2	19	6	6	3	1	3	1	0
	Horses		0	0	0	46	0	0	1	0	0	3
	Sky		0	1	0	0	44	0	5	0	0	0
	Cheetahs		6	6	3	3	4	24	2	1	1	0
	Eagles		1	1	2	4	10	1	30	0	0	1
	Night		0	0	0	0	0	0	0	41	3	6
	Sunset		13	2	1	0	1	0	0	1	30	2
	Roses		0	1	0	4	0	1	0	1	4	39

Table 6  
Confusion matrix for the CAC descriptor without compression

Output categories		CAC	Lions	Elephants	Tigers	Horses	Sky	Cheetahs	Eagles	Night	Sunset	Roses
Input categories	Lions	33	5	3	1	4	0	2	0	2	0	0
	Elephants	10	31	0	2	6	1	0	0	0	0	0
	Tigers	7	3	18	6	4	3	3	3	0	3	3
	Horses	1	0	0	47	1	0	0	0	0	0	1
	Sky	0	3	0	0	46	0	1	0	0	0	0
	Cheetahs	11	3	3	11	3	11	3	1	3	1	1
	Eagles	0	1	1	3	9	0	33	0	1	2	2
	Night	0	0	0	0	2	0	4	38	3	3	3
	Sunset	6	0	0	0	1	0	1	1	36	5	5
	Roses	0	0	0	2	2	0	1	8	1	36	36

Table 7  
Confusion matrix for the CAC descriptor, compression ratio ( $n/m$ ) = 20%

Output categories		CAC	Lions	Elephants	Tigers	Horses	Sky	Cheetahs	Eagles	Night	Sunset	Roses
Input categories	Lions	36	4	2	1	4	0	2	0	1	0	0
	Elephants	8	32	0	3	6	0	0	0	1	0	0
	Tigers	9	3	14	6	5	6	1	2	0	4	4
	Horses	1	0	0	47	1	0	0	0	0	1	1
	Sky	0	3	0	0	46	0	1	0	0	0	0
	Cheetahs	12	4	0	13	2	11	4	1	2	1	1
	Eagles	0	0	1	4	9	0	33	0	1	2	2
	Night	0	0	0	0	2	0	4	38	3	3	3
	Sunset	7	0	0	0	1	0	1	1	34	6	6
	Roses	0	0	0	2	2	0	1	8	2	35	35

Table 8  
Confusion matrix for the M7CS descriptor without compression

Output categories		M7CS	Lions	Elephants	Tigers	Horses	Sky	Cheetahs	Eagles	Night	Sunset	Roses
Input categories	Lions	35	7	3	0	1	2	0	0	2	0	0
	Elephants	7	34	4	2	2	1	0	0	0	0	0
	Tigers	0	1	40	1	5	0	3	0	0	0	0
	Horses	0	0	2	48	0	0	0	0	0	0	0
	Sky	0	1	1	0	46	0	2	0	0	0	0
	Cheetahs	3	9	2	1	2	29	2	0	1	1	1
	Eagles	0	1	1	1	6	1	39	0	0	1	1
	Night	0	0	0	0	1	1	1	39	6	2	2
	Sunset	2	0	1	0	0	0	1	0	43	3	3
	Roses	0	0	0	0	0	0	3	0	1	46	46

Table 9  
Confusion matrix for the M7CS descriptor, compression ratio ( $n/m$ ) = 20%

Output categories		M7CS	Lions	Elephants	Tigers	Horses	Sky	Cheetahs	Eagles	Night	Sunset	Roses
Input categories	Lions	36	6	3	0	1	2	0	0	0	2	0
	Elephants	6	34	4	2	2	1	0	0	0	1	0
	Tigers	0	1	41	1	3	0	4	0	0	0	0
	Horses	0	0	3	47	0	0	0	0	0	0	0
	Sky	0	1	1	0	46	0	2	0	0	0	0
	Cheetahs	3	9	2	1	2	29	2	0	1	1	1
	Eagles	0	1	3	1	7	1	36	0	0	1	1
	Night	0	0	0	0	1	1	1	39	7	1	1
	Sunset	4	0	0	0	0	0	0	0	41	5	5
	Roses	0	0	0	0	0	0	0	3	0	1	46

Table 10  
Confusion matrix for the CPAH descriptor without compression

Output categories		CPAH	Lions	Elephants	Tigers	Horses	Sky	Cheetahs	Eagles	Night	Sunset	Roses
Input categories	Lions	40	2	0	2	4	1	1	0	0	0	0
	Elephants	5	21	0	2	10	6	1	0	0	0	5
	Tigers	5	0	20	10	2	5	5	3	0	0	0
	Horses	0	0	0	48	0	0	0	0	1	1	1
	Sky	0	0	0	0	48	0	2	0	0	0	0
	Cheetahs	10	1	2	5	6	23	2	0	1	0	0
	Eagles	1	0	2	4	8	0	33	0	0	0	2
	Night	0	0	0	1	1	0	1	41	4	2	2
	Sunset	2	0	0	0	3	0	0	2	42	1	1
	Roses	0	0	0	2	0	0	0	0	0	3	45

Table 11  
Confusion matrix for the CPAH descriptor, compression ratio ( $n/m$ ) = 20%

Output categories		CPAH	Lions	Elephants	Tigers	Horses	Sky	Cheetahs	Eagles	Night	Sunset	Roses
Input categories	Lions	40	2	0	2	5	1	0	0	0	0	0
	Elephants	6	20	0	2	10	6	1	0	0	0	5
	Tigers	6	1	19	9	2	5	5	3	0	0	0
	Horses	0	0	0	47	0	0	1	0	1	1	1
	Sky	0	0	0	0	48	0	2	0	0	0	0
	Cheetahs	10	1	2	5	6	23	2	0	1	0	0
	Eagles	1	0	2	4	8	0	33	0	0	2	2
	Night	0	0	0	1	1	0	1	41	4	2	2
	Sunset	2	0	0	0	4	0	0	3	41	0	0
	Roses	0	0	0	2	0	0	0	1	2	45	45

Table 12  
Confusion matrix for the LCI descriptor without compression

Output categories		LCI	Lions	Elephants	Tigers	Horses	Sky	Cheetahs	Eagles	Night	Sunset	Roses
Input categories	Lions	37	6	2	0	2	2	0	0	0	1	0
	Elephants	5	40	0	2	2	1	0	0	0	0	0
	Tigers	3	3	31	5	6	1	1	0	0	0	0
	Horses	1	0	2	44	2	0	0	0	0	0	1
	Sky	0	1	0	0	48	0	1	0	0	0	0
	Cheetahs	7	6	5	3	1	21	5	0	1	1	1
	Eagles	0	1	1	1	6	0	40	0	1	0	0
	Night	0	0	0	0	1	0	2	41	3	3	3
	Sunset	6	0	0	0	0	0	1	1	38	4	4
	Roses	0	0	0	0	0	0	1	2	1	46	46

Table 13  
Confusion matrix for the LCI descriptor, compression ratio ( $n/m$ ) = 20%

Output categories		LCI	Lions	Elephants	Tigers	Horses	Sky	Cheetahs	Eagles	Night	Sunset	Roses
Input categories	Lions	37	6	2	0	2	2	0	0	0	1	0
	Elephants	5	39	0	2	3	1	0	0	0	0	0
	Tigers	4	3	29	5	6	2	1	0	0	0	0
	Horses	1	0	2	44	2	0	0	0	0	0	1
	Sky	0	1	0	0	49	0	0	0	0	0	0
	Cheetahs	8	6	5	5	1	18	4	1	1	1	1
	Eagles	0	1	1	1	6	0	40	0	1	0	0
	Night	0	0	0	0	1	0	2	42	2	3	3
	Sunset	6	0	0	0	0	0	1	1	35	7	7
	Roses	0	0	0	0	0	0	1	2	2	45	45

## References

- [1] A.W.M. Smeulders, et al., Content-based image retrieval at the end of the early years, *IEEE Trans. PAMI* 22 (2000) 1349–1380.
- [2] O. Chapelle, P. Haffner, V. Vapnik, SVMs for histogram-based image classification, *IEEE Trans. Neural Networks*, 1999.
- [3] Q. Tian, P. Hong, T.S. Huang, Update relevant image weights for content-based image retrieval using support vector machines, *IEEE ICME'2000*.
- [4] S. Tong, E. Chang, Support vector machine active learning for image retrieval, *Proceedings of ACM MM'01*.
- [5] Y. Rui, T.S. Huang, M. Ortega, S. Mehrotra, Relevance feedback: a power tool for interactive content-based image retrieval, *IEEE Trans. Circuits Syst. Video Technol.*, 1998.
- [6] G. Qiu, Indexing chromatic and achromatic patterns for content-based color image retrieval, *Pattern Recognition* 35 (8) (2002) 1675–1686.
- [7] G. Qiu, K-M Lam, Spectrally layered color indexing, in: M.S. Lew, N. Sebe, J.P. Eakins (Eds.), *Image and Video Retrieval, CIVR2002, Lecture Notes in Computer Science*, Vol. 2383, Springer, Berlin, July 2002, pp. 100–107.
- [8] J. Berens, G.D. Finlayson, G. Qiu, Image indexing using compressed color histogram, *IEEE Proceedings, Vision, Image and Signal Processing* 147 (4) (2000) 349–355.
- [9] J. Huang, S.R. Kumar, M. Mitra, W.-J. Zhu, R. Zabih, Image indexing using color correlograms, *Int. J. Comput. Vision* (1999) 245–268.
- [10] B.S. Manjunath, et al., Color and texture descriptors, *IEEE Trans. Circuits Syst. Video Technol.* 11 (6) (2001).
- [11] Y. Freund, R. Schapire, A decision-theoretic generalization of online learning and an application to boosting, *J. Comput. Syst. Sci.* 55 (1) (1997) 119–139.
- [12] K. Tieu, P. Viola, Boosting image retrieval, *Proc. of CVPR* 1 (2000) 228–235.
- [13] G.D. Guo, A.K. Jain, W.Y. Ma, H-J Zhang, Learning similarity measure for natural image retrieval with relevant feedback, *IEEE Trans. Neural Networks* 13 (2002) 811–820.
- [14] N. Howe, Boosted Image Classification: An Empirical Study, *ICML Workshop on Machine Learning in Computer Vision*, July 9, 2002, Sydney, Australia.
- [15] T. Balachander, R. Kothari, Introducing Locality and Softness in Subspace Classification, *Pattern Analysis & Applications*, Vol. 2, Springer, London, 1999, pp. 53–58.

- [16] K-S Goh, E. Chang, K-T Cheng, Support Vector Machine Pairwise Classifiers with Error Reduction for Image Classification, Third International Workshop on Multimedia Information Retrieval (MIR2001), Ottawa, Canada, October 5, 2001.
- [17] T. Serre, B. Heisele, S. Mukherjee, T. Poggio, Feature selection for face detection, MIT A.I. Memo No. 1697, September, 2000.
- [18] A.K. Jain, B. Chandrasekaran, Dimensionality and sample size considerations in pattern recognition practice, in: P.R. Krishnaiah, L.N. Kanal (Eds.), Handbook of Statistics, Vol. 2, 1987, pp. 835–855.
- [19] S.J. Raudys, A.K. Jain, Small sample size effects in statistical pattern recognition: recommendations for practitioners, IEEE Trans. Pattern Anal. Machine Intelligence 13 (1991) 252–264.
- [20] A. Gersho, R.M. Gray, Vector quantization and signal compression, Kluwer Academic Publishers, Boston, 1992.
- [21] MPEG7 FCD, ISO/IEC JTC1/SC29/WG11, March 2001, Singapore.
- [22] C.M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.
- [23] M. Turk, A. Pentland, Eigenfaces for Recognition, J. Cognitive Neurosci. 3 (1) (1991) 71–86.
- [24] R.C. Gonzalez, R.E. Woods, Digital Image Processing, 2nd Edition, Prentice-Hall, Englewood Cliffs, NJ, 2002.
- [25] M. Hearst, Support vector machines, IEEE Intelligence Systems, pp. 18–28, July/August, 1998.
- [26] C. Burges, A tutorial on support machines for pattern recognition, Data Mining and Knowledge Discovery, Kluwer Academic Publishers, Vol. 2, 1998, pp. 121–167.
- [27] <http://www.research.microsoft.com/~jplatt/smo.html>.
- [28] G. Ciocca, R. Schettini, Using a relevant feedback mechanism to improve content based image retrieval, Proceedings of Visual'99, 1999, pp. 107–114.

**About the Author**—GUOPING QIU received the B.Sc. degree in Electronic Measurement and Instrumentation from the University of Electronic Science and Technology of China, Chendu, China, in July 1984 and the Ph.D. degree in Electrical and Electronic Engineering from the University of Central Lancashire, Preston, England, in 1993. He is currently a Reader in the School of Computer Science at the University of Nottingham, UK. He has research interests in the broad area of computational visual information processing and has published widely in this area. His current research is focused on two main areas—making better images and graphics and managing large image repositories. More about his research can be found in URL: <http://www.cs.nott.ac.uk/~qiu>.

**About the Author**—XIA FENG received the B.Sc. degree in Computer Science from Nan Kai University, Tianjin, China, in July 1991 and the M.Sc. degree in Computer Science from the Nan Jing University of Aeronautics and Astronautics, China, in 2001 and is pursuing her Ph.D. degree in Nan Kai University, China. She is currently an associate professor in the School of Computer Science and Technology at the Civil Aviation University of China, Tianjin, China. From May 2002 to May 2003, She was a visiting researcher in the School of Computer Science at the University of Nottingham, UK. Her current research interests are in image retrieval and data mining.

**About the Author**—JIANZHONG FANG received the Bachelor's degree in Computer Engineering from the National University of Defense Technology, Changsha, China, in 1989. From 1989 to 1997, he was engaged in research and development of Central Processing Unit of mainframe computers at the East China Research Institute of Computer Technology, Shanghai, China. From 1997 to 2001, he worked in R& D department in Printroxin in Singapore. Since 2001, he has been doing his Ph.D. program at the University of Nottingham, UK. He is now studying computational methods of generic visual object detection and recognition.