Illumination–Invariant Color Object Recognition via Compressed Chromaticity Histograms of Color–Channel–Normalized Images

Mark S. Drew, Jie Wei, and Ze-Nian Li
School of Computing Science, Simon Fraser University,
Vancouver, B.C. Canada V5A 1S6
email {mark, jiew, li}@cs.sfu.ca

Abstract

Several color object recognition methods that are based on image retrieval algorithms attempt to discount changes of illumination in order to increase performance when test image illumination conditions differ from those that obtained when the image database was created. Here we extend the seminal method of Swain and Ballard to discount changing illumination. The new method is based on the first stage of the simplest color indexing method, which uses angular invariants between color image and edge image channels. That method first normalizes image channels, and then effectively discards much of the remaining information. Here we adopt the color–normalization stage as an adequate color constancy step. Further, we replace 3D color histograms by 2D chromaticity histograms. Treating these as images, we implement the method in a compressed histogram–image domain using a combination of wavelet compression and Discrete Cosine Transform (DCT) to fully exploit the technique of low–pass filtering for efficiency. Results are very encouraging, with substantially better performance than other methods tested. The method is also fast, in that the indexing process is entirely carried out in the compressed domain and uses a feature vector of only 36 or 72 values.

1 Introduction

Retrieval from an image or video database can be expected to play a larger and larger role as such collections proliferate. Swain and Ballard's seminal work on color object recognition by means of a fast matching of color histograms [1] began an interest in the use of simple color–based features for such applications. In this method, a database of coarse histograms indexed by three color values is built up. A very simple and fast histogram matching strategy can often identify the correct match for a new image, or a near–match, by using an L1 metric of histogram differences. It was soon realized that, along with confounding factors such as object pose, noise, occlusion, shadows, clutter, specularities, and pixel saturation, a major problem arose because of the effect of changing illumination on images of color objects [2]. After all, it would be quite natural for an object in a database to be presented as it appears imaged under some other lighting.

Several color object recognition schemes have been developed that purport to take illumination change into account in an invariant fashion. Nevertheless, while producing good results in many cases, we show below that some of these methods may actually produce poor results when confronted by illumination change.

In this paper we address the problem of illumination change by extending the original Swain and Ballard method to include illumination invariance in a natural and simpler way than heretofore. To do so, we apply the first step, color–channel normalization, that appears in the simplest illumination–invariant method, based on the angles between color image channels [3], and argue that this first step is really all that is required to deal properly with illumination invariance. Then, with an aim of reducing the dimensionality of the feature space involved, we shift from a full–color (3–dimensional) representation to one based on (2–dimensional) chromaticity, and show that the essential illumination–invariant color information is maintained across this data reduction. The normalization step has the effect of undoing a changing–illumination induced shift in pixel color–space position, in chromaticity space.

Histograms in chromaticity space are indexed by two values, and chromaticity histograms for two different objects are often very different. Since we are now effectively dealing with feature–space *images*, we can easily apply image–based compression techniques to the chromaticity histograms and then index into a database with a small feature vector based on the compressed histogram. Here, we combine a technique of first applying a wavelet–based reduction with a second step of truncation of the Discrete Cosine Transform (DCT) image. This results in an effective low–pass filtering of the chromaticity histogram. The resulting image indexing scheme is very efficient in that it uses a feature vector of only 36 or 72 values. The method is remarkably accurate for such a simple, compact scheme and, as we show below, is more resilient to noise than other methods explored here.

In §2 we outline the significance of image normalization and the role of chromaticity histograming in image database indexing. In §3 we describe a histogram–image compression scheme using wavelet–based reduction, DCT, and truncation. Experimental results are presented in §4 and some conclusions are drawn in §5.

2 Illumination–Invariant Indexing

2.1 Illumination invariance and image normalization

The simplest previous method for illuminant–invariant recognition is the Color Angles method of

Finlayson et al., who proposed indexing on six numbers corresponding to the "angles" amongst image and edge–image channels [3]. That method consists of the following: *normalize* each mean–subtracted color channel R, G, B to length 1 by dividing by the square root of the sum of squares of that channel, and then take as indexing numbers the three angles formed from the inverse cosine of the products $R \cdot G$, $R \cdot B$, $G \cdot B$. Along with these, append a second set of angles derived in exactly the same way from the edge image of the smoothed color–normalized color image, $\nabla^2 G\boldsymbol{\rho}$, where $\boldsymbol{\rho} = (R, G, B)$. The idea is that, if camera sensors are sufficiently narrowband, these angles are invariant to color shifts in the illuminant because in that situation illumination change corresponds to a simple scaling of the color channels (the "coefficient rule"). In the limit of Dirac delta function sensors, this approximation would hold identically. If sensors are not "sharp" enough, then provided one knows the camera sensor curves one can carry out a "spectral sharpening" operation [4] to bring the color angle approximation more in line with actual conditions [3]. Thus illumination change amounts to a diagonal matrix transform among color channels. Even without sharp sensors, if the illuminant is fairly white then a factor model [5] of color formation still leads to a diagonal transform.

To see this, consider the RGB triple $\boldsymbol{\rho}$ resulting from some set of lights with spectral power distributions $E_i(\lambda)$ and indexed by $i = 1..L$ impinging on a Lambertian surface with surface spectral reflectance function $S^x(\lambda)$, under conditions of orthography. If the camera system color sensors have sensitivity functions $\boldsymbol{q}(\lambda)$ then

$$\boldsymbol{\rho}^x = \sum_{i=1}^{L} (\boldsymbol{a}_i^T \boldsymbol{n}^x) \int E_i(\lambda)\, S^x(\lambda)\, \boldsymbol{q}(\lambda)\, d\lambda \qquad (1)$$

where $\boldsymbol{n}^x$ is the surface normal for that pixel and light E_i has normalized spatial direction $\boldsymbol{a}_i$. If camera sensors $\boldsymbol{q}$ are narrow band enough to be approximated as sampling a single frequency,

$$q_k(\lambda) \simeq q_k(\lambda_k)\delta(\lambda - \lambda_k) \;\; ; \;\; k = 1..3$$

with constants $q_k(\lambda_k)$, then we have

$$\rho_k^x = \tau_k^x S^x(\lambda_k) q_k(\lambda_k) \qquad (2)$$

where

$$\tau_k^x = \sum_{i=1}^{L} (\boldsymbol{a}_i^T \boldsymbol{n}^x)\, E_i(\lambda_k)$$

Clearly, under a change of illumination environment $E_i \rightarrow E_i'$, $\boldsymbol{a}_i \rightarrow \boldsymbol{a}_i'$, $L \rightarrow L'$, we have a diagonal transformation of RGB values $\boldsymbol{\rho}$ *provided the surface is flat* so that τ_k is independent of x. Then the normal $\boldsymbol{n}$ of the flat surface patch can be tilted, in the new image under new lighting, and still have a diagonal transform relate color under the old illumination to color under the new illumination with the diagonal elements τ_k'/τ_k being the same for every pixel. Ignoring surface normal terms or considering very distant viewing as in satellite imaging (cf. [6]) constitutes a good working hypothesis, at least for image retrieval purposes, if not for physically correct surface reconstruction, and we see below in §4 that indeed applying the diagonal model of illuminant change works well enough for this purpose, even for non–flat objects.

Even without sharp sensors, a factor model [5] approximates the kth component of the color b_k^i reflected by a surface with color s_k, for the ith incident light with color e_k^i, by the product $b_k^i \simeq s_k e_k^i / \sigma_k$, $k = 1..3$, with camera scaling term $\sigma_k = \int q_k(\lambda) d\lambda$. Again one arrives at an expression like (2), with

$$\boldsymbol{\rho}^x = \tau_k^x\, s_k^x \,/\, \sigma_k \qquad (2')$$

where now

$$\tau_k^x = \sum_{i=1}^{L} (\boldsymbol{a}_i^T \boldsymbol{n}^x)\, e_k^i$$

Normalization of each color band in an L2 norm effectively enforces illumination invariance in a diagonal model: for considering each color channel $\rho_k, k = 1..3$ separately as a long vector (cf. [3]) we see that an illumination change

$$\rho_k^x \rightarrow \rho_k^{x\prime} = (\tau_k'/\tau_k)\, \rho_k^x \qquad (3)$$

simply corresponds to a change of vector length. Normalizing these lengths to unity for every image effectively discounts change of lighting.

In this paper, we adopt the first, normalization step of the Color Angles method (but do not subtract the mean from images). In our view, and as shown below in §4, since the normalization step constitutes the illumination invariance of the method, it is needlessly prodigal to not employ more of the information in the image once illumination change has been guarded against. And in §4 the Color Angles method is shown to fail, very likely because it does throw away so much information.

Indeed, one could simply utilize the original Swain and Ballard method, but using 3–dimensional histograms of *normalized* color bands (i.e., **each color channel as a long vector is normalized**).

However, once we give up the range 0..255, as when we normalize each color band, the issue remains of exactly what our histogram bins shall be. That is, we can make a guess at what a likely maximum and minimum value would be, based on a random input color image say, and then use either uniform bins or some form of histogram equalization. However, the statistics of any particular image dataset will likely be different than those for a stochastic image and we would perhaps be better off using an adaptive Lloyd–Max quantizer. But this detracts somewhat from the original attractive simplicity of the method. E.g., in [2] a 3–dimensional, 4096 bin edge–based method is hampered somewhat by the necessity of inventing 'good' histogram bins. As well, that method is limited by the fact that edge ratios at low intensity levels can become dominated by noise.

Instead, here we can again fix a natural range for maxima and minima, and explore the possibility of using uniform bins, if we go to a more compact, 2-dimensional histogram structure. By sacrificing image intensity we can still capture the essential color information by going over to chromaticity images. Starting with normalized color images, the resulting chromaticity images will be relatively illuminant invariant.

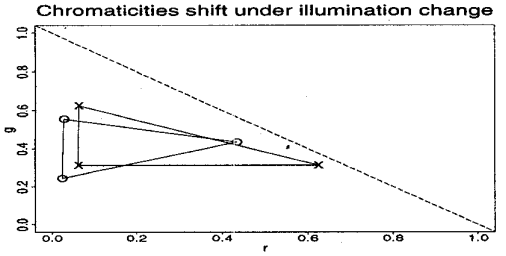
Since we use simple histogram matching, the method will also be invariant to 2–dimensional linear transformations in the image, such as rotations and shifts. By making each histogram sum to unity we also make the method invariant to the size of the object in the image.

2.2 Illuminant–invariant chromaticity histograms

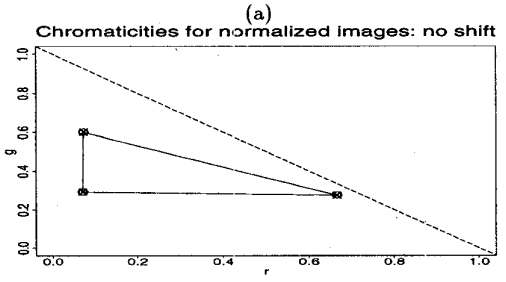
Define the chromaticity (r, g) for each normalized pixel by [7]

$$r = R/(R+G+B) \quad , \quad g = G/(R+G+B) \quad (4)$$

and consider the 2–channel chromaticity image as that $N \times 2$ array formed from the color channels of an N–pixel image. Swain and Ballard also considered histograms of chromaticity, but with no color–band normalization step — instead, they suggested some unresolved form of color constancy preprocessing. Here, the normalization step essentially accomplishes compensation for changing illumination.



(a)



(b)

Figure 1: (a): Chromaticity shift, under an illumination change in the diagonal model. (b): Chromaticities for color–normalized images do not shift.

To see how illumination change affects chromaticity, suppose we adopt the diagonal model of illumination change. Then under a diagonal transform $diag(1, 2, 3)$ three pixels with RGB values $(1, 10, 5)$, $(1, 5, 10)$, $(10, 5, 1)$ have their chromaticities, labeled 'X' in Fig. 1(a), shifted to values labeled 'O'. (Note that chromaticity values are floats in the range 0 to 1 and must occur below the dashed line in the figure since $r + g \leq 1$.) [1] But employing the L2 normalization of the three RGB vectors before calculating their chromaticities, we find in Fig. 1(b) that the shift is eliminated.

Consider the image of Fig. 2(a). If we plot the chromaticity r versus the chromaticity g for this image, as in Fig. 2(b), the eye tend to accumulate points into a histogram. Since chromaticity values must lie below the dashed line, a chromaticity histogram may use half the number of bins.

Now consider a similar object under different lighting, rotated, and translated somewhat (and even 'translated' to/from French), as shown in Fig. 2(c). Notice that the plot of r versus g, shown in Fig. 2(d), has shifted to some degree. Figure 2(e) shows the two plots overlaid.

Now applying the L2 normalization step first, we see that the two chromaticity plots are much closer to each other, as shown in Fig. 2(f). Here the scatterplot of the chromaticity of the original, color–normalized image is overlaid by that of the object imaged under changed illumination.

Thus a chromaticity–based image retrieval method has a good chance of succeeding, even in the face of changing illumination.

2.3 Histogram equalization

The original Swain and Ballard algorithm used uniform histogram bins, coarsely accumulating into 16 bins over the range 0..255. [2] Since image channel normalization and the subsequent step of forming chromaticities may make uniform bins non–optimal, here we consider histogram equalization for chromaticities. To do so, we histogram–equalize the probability density for chromaticity values in either channel for color–channel–normalized versions of the 66 model images used in Swain and Ballard's original method. These consist of various manufactured products, clothing, etc. and are shown in color in [1] (Page 30, Figure 4).

As another possible source of histogram bins, we could instead simply normalize the color channels of a large image with values drawn from a uniform random distribution in the range 0..255, form the chromaticity, and histogram–equalization.

Below, in §4, we examine what effect using these histogram bins, rather than uniform ones, may have an the accuracy of image retrieval. There we see that using optimized histogram bins may have a real, but somewhat marginal effect, relative to the much more efficient uniform bin method.

2.4 Histogram intersection

In [1], a very useful histogram metric is developed, which we adopt here for uncompressed histograms.

Adapting Swain and Ballard's definition to the present situation, we define the intersection of chromaticity histograms $\boldsymbol{H}_a$ and $\boldsymbol{H}_b$ as:

$$\mu \equiv \sum_{i,j} min\{H_a(i,j), H_b(i,j)\} \quad (5)$$

[1] Lin and Lee [8] also use chromaticity–based indexing, and project histograms under new illuminants into an eigenspace to accomplish color constancy, whereas we simply normalize the length of each color channel.

[2] Swain and Ballard use a matrix–transformed color space as well as RGB, and more coarsely sample the first dimension, for a total of $8 \times 16 \times 16 = 2048$ bins, but we simply use RGB and 16^3 bins in our implementation of their original method here. As well, Swain's algorithm explores accounting for changes in 3-dimensional orientation by storing several views of the object in the model database. We simply store a single model image.

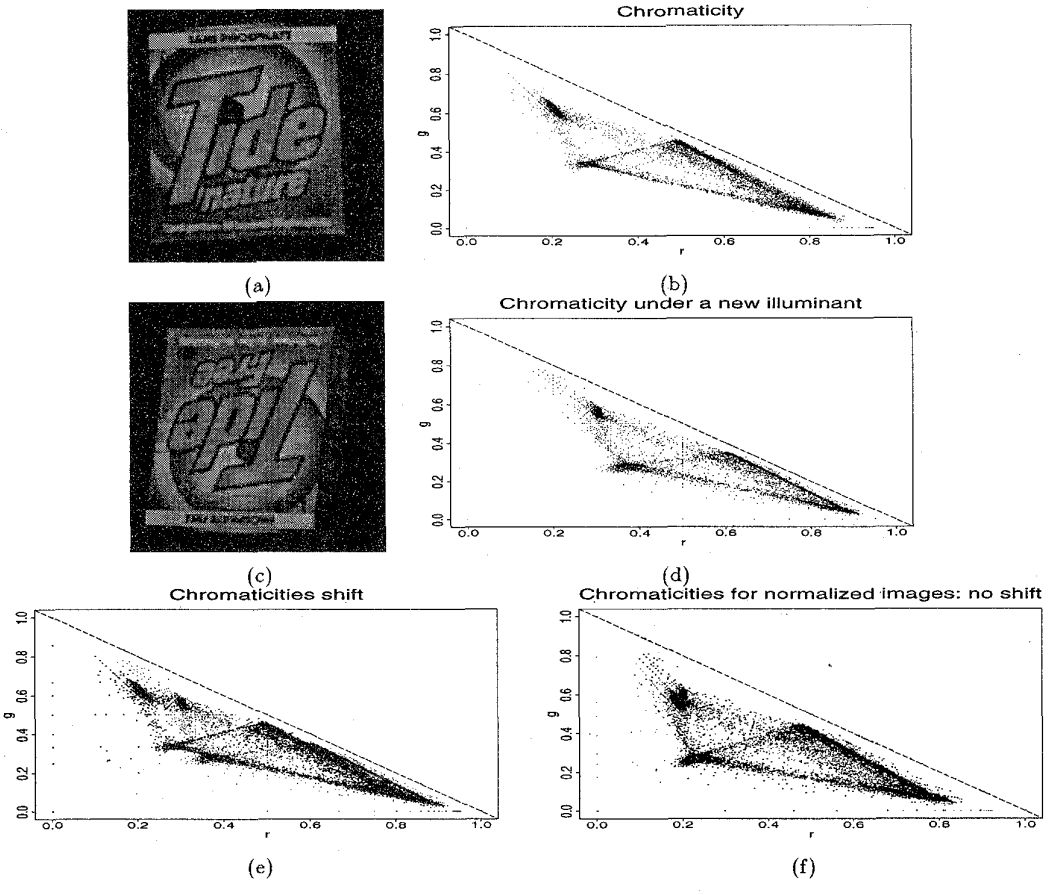


Figure 2: (a): Model image. (b): Chromaticity for unnormalized model image. (c): Test image, under new illumination. (d): Chromaticity for unnormalized test image. (e): Chromaticities shift under change of illumination. (f): Chromaticities for color–normalized images do not shift.

Swain and Ballard normalize intersection (or match) values by the number of pixels in the model histogram, and thus matches are between 0 and 1. Alternatively, one can make the area under each histogram equal to unity, effectively making each image have the same number of pixels and turning the histogram into a probability density. Time for histogram intersection is proportional to the number of histogram bins and so is very fast.

It can be shown that histogram intersection is equivalent to 1 minus an L1 distance, and so $(1-\mu)$ forms a metric δ, with $\delta = 1-\mu = (1/n)\sum |\boldsymbol{H}_a - \boldsymbol{H}_b|$, where n is the number of histogram bins. The utility of this metric is that it helps to alleviate the effects of noise, in the following way. Suppose an image has a great many very small histogram entries, arising perhaps from illumination that varies spectrally across the image, so that object colors tend to not form well–defined clusters. If such low values do not occur in the particular model image chromaticity histogram being compared to, then such low values might tend to dominate, in an L2, squared–differences norm. Instead, here, zero occurrences in the model histogram are counted instead and such effects do not contribute to the metric.

Object recognition proceeds by intersecting the chromaticity histogram of an unknown object with similar histograms precomputed and stored in a data base. The highest value of μ, or in other words the smallest distance value $(1-\mu)$ indicates the database image that matches best.

We computed matches using the above method, and also using an L2 distance and found little change in match rankings amongst images in the database when the algorithm was presented with a novel image. As well, since we may view the chromaticity histogram as a kind of 'image', we also calculated the image correlation between test and model images, again with little advantage over the histogram intersection method.

2.5 Other illumination–invariant methods

The color angle idea can be summarized as a use of a covariance matrix (plus the addition of an edge image covariance). As such, its closest relative is the

method of Healey and Wang [9], which has been applied to color textures. Histogram–based methods have distinct advantages over [9], however, in that they are invariant to 2D rotations and translations.

Healey and Wang's method [9] consists of considering a set of correlations $R_{jk}(n,m)$, $j,k = 1..3$ (i.e., covariance terms for mean–subtracted image channels), with shifts n,m from 0 to 16 pixels. In practice, they use $n = 0..16$ and $m = -16..16$, so for each set of channel products jk they use 561 correlations. Then there are 6 different color channel products (RR, RG, etc.) for each set of shifts, so an image feature vector has $561 \times 6 = 3366$ elements. For matching, an error measure is taken to be the L2 distance between a new image's feature vector and the result of writing that vector in terms of the eigenvectors of each model image's set of correlations. Let us refer to this method as the Correlation Method.

In comparison, other illumination–invariant matching schemes include the color–constant–color–indexing (CCCI) method [2], which uses the counts in 4096 histogram bins for a color edge image, and the method of Slater and Healey [10], which uses moments of color histograms and makes the assumption that local image patches can be modeled as textured, flat planes.

3 Chromaticity Histogram–Image Compression

In the next section we show that if we store a representation of each histogram–image that is first reduced in size by a wavelet transformation, and then further reduced by going to the frequency domain and discarding higher–frequency DCT coefficients, one can derive a simple indexing method that is efficient and invariant under illuminant change. In the following method, the number of parameters in the feature vector is only 36 or 72 values.

Chromaticity histogram matching without compression could be computationally intensive; we would like to recover an accurate approximation of histogram–images without sacrificing efficiency. As a guiding principle it would also be sensible to maintain a linear relationship between the histogram–image and its compressed representation. We can adhere to this principle by applying only linear operations while compressing the histogram–images. Therefore, here we first apply a linear low–pass filter to both histogram–images, resulting in new histograms $\boldsymbol{H}$ and $\boldsymbol{H}'$. To best approximate the chromaticity histograms the low–pass filtered histogram–images should approximate the original ones as closely as possible, yet be of lower resolution. The scaling function of biorthonormal wavelets, as a symmetrical low–pass filter, can be exploited to that end. Basically, scaling functions with more "taps" use polynomials of higher order to approximate the original function (the number of taps is the number of nonzero coefficients) [11]. Our main concern is to capture the most detail but in lower resolution, and after some experiments we arrived at a good balance between efficiency and precision by using the symmetrical 9–tap filter [12]. The 1D mask of the separable 2D scaling function is:

$$\{0.037829, -0.023849, -0.110624, 0.377403, 0.852699, 0.377403, -0.110624, -0.023849, 0.037829\}$$

After applying the scaling function several times to the original histograms, assuming for simplicity square histograms with resolution $2^n \times 2^n$, we obtain size 16×16 lower resolution histogram images.

We then apply a DCT, transforming $\boldsymbol{H}$ into $\widehat{\boldsymbol{H}}$ and indexing on $\widehat{\boldsymbol{H}}$. Since the lower frequencies capture most of the energy of an image, after applying the DCT we can retain just the lower frequency coefficients for histogram–image database indexing — a very effective and efficient way of realizing a further low–pass filtering. By experiment we found that using only 36 coefficients worked well, these being those in the first 36 numbers in the upper left corner of the DCT coefficient matrix, as shown in Fig. 3. [3]

Denote by $\boldsymbol{H}_d$ the 36 values (derived from DCT coefficients) in the order illustrated in Fig. 3. We index on the L2 distance between $\boldsymbol{H}_d$ for the model histogram and that for the test histogram. Let us call the chromaticity–histogram method (with no compression) the *ChromHist–A method*, and that based on compressed histograms the *ChromHist–B1 method*.

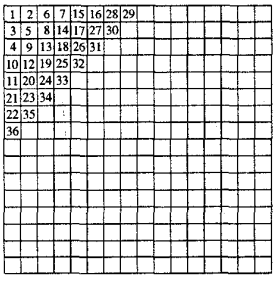


Figure 3: Arrangement of DCT coefficients

Populating the database, then, consists of calculating off-line the 36 values $\boldsymbol{H}_d$, viewed as indices for each model image. For image query, first the 36 values for the query image are computed, thus obtaining $\boldsymbol{H}'_d$; then for every model image, the L2 distance $[\sum(\boldsymbol{H}'_d - \boldsymbol{H}_d)^2]^{1/2}$ is calculated. The model image minimizing the distance is taken to be a match for the query image.

Note that in the ChromHist–B1 method only reduced, DCT transformed, truncated histograms are used — no inverse transforms are necessary and the indexing process is entirely carried out in the compressed domain.

[3] Instead of using a conventional 8×8 window for the DCT, a 16×16 window is adopted. As a result, a finer resolution (twice as high as with 8×8) in the spatial frequency domain is realized. Since the low–pass filtering after DCT can only retain a limited number of coefficients for efficiency, the net effect of having a larger (16×16) window is that a more detailed parameterized description at the lower end of the spectrum is facilitated. This is beneficial when very low–resolution wavelet images are used for matching in our method.

The choice of a reduced resolution of 16 × 16 for chromaticity histograms and 36 DCT coefficients retained was made quite empirically. For comparison we also define a *ChromHist-B2 method* in which the wavelet-reduced chromaticity histogram images have a resolution of 32 × 32. For this method we could still use a 16 × 16 window for the DCT by applying the DCT to each of the four quadrants of the 32 × 32 histogram-image. However, a more compact representation can be used here. Recall that only the lower left-half of the chromaticity histogram contains non-zero counts (shown as the shaded region in Fig. 4(a)). To reduce computation we rearrange the histogram by cutting off the right-half of the shaded triangle and append it to the top of the remaining region to get a new rectangular region shown in Fig. 4(b). The new histogram consists of two 16 × 16 subimages; applying a 16 × 16 DCT to each of these, with 36 coefficients each, gives method ChromHist-B2, which indexes on a total of 72 values.

As well, an auxiliary step of carrying out a square root operation on the histogram values before turning them into probability densities was found to improve the performance over the results presented in [13] where no square root was applied.

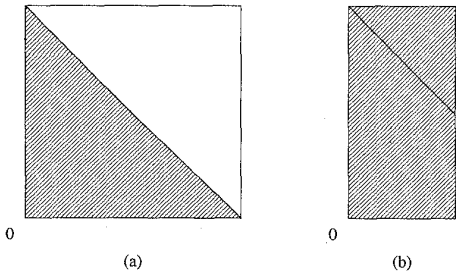


Figure 4: ChromHist-B2:Preparing two 16 × 16 subimages

4 Experiments

4.1 Swain's database: no illumination change

Here we compare the new method (denoted ChromHist-A), based on the chromaticity of color-normalized images, and its variants that operate in the compressed domain (denoted ChromHist-B1 and B2), with the original Color Indexing method [1], and with the Color Angles method [3], with model and test images taken from Swain's original paper. We find that the new method does well on this database, better than the other methods tested. Since the original images are quite special in the sense that objects appear on a black background that is easily thresholded away, we also test how the performance of each method degrades in the presence of clutter. A simple approach to providing a background for each image is to generate random noise all over the image—this also provides a standard partial sensitivity analysis. To the eye, this makes no difference in recognizing correct matches.

The new ChromHist method is relatively insensitive to the size of the histogram. Here we use 128 × 128 histograms; this is not necessarily optimum, but is useful for wavelet-based compression because each side of the histogram 'image' is a power of 2.

Table 1 shows matching scores for the methods tested, on the original images and on images with uniform random noise with rms value 15 added. A rank of 1 means a correct match, and a higher value indicates at what relative order the correct match was to be found. We find that the new method is more resilient to noise than previous methods tested.

Along with matching scores, we also wish to know to what degree each algorithm fails, if it does fail. To characterize each algorithm in these terms, we wish to compare values of metric distances from each test image to every one of the model images in terms of the distance from the test image to its corresponding correct model image. Therefore, for each test image, we divide these distances by the distance from the test image to the correct model image, and thus form distance ratios. Then a good characterization of performance is the mean value of distance ratios for *worst-case incorrect* models for each test (cf. [12]). Here, the worst case is the model that is closest to the test image, but is not the correct model. I.e., the worst case is the "next-best" mismatch model. Table 1 shows the mean and standard deviation over all experiments of this next-best mismatch distance, always using a metric distance. For the present database, this means that we give the mean and standard deviation of next-best distance ratios for the 65 incorrect model images over 30 tests. For algorithms that use an L2 distance metric, we use that distance. For algorithms that use histogram-match values μ with a maximum value of 1, we use the L1 distance $(1-\mu)$. Comparing L1 distance with L2 distance is not entirely reasonable, but still distance ratio values do characterize algorithm performance. In Table 1 we show L1 distances in normal font and L2 distances in italics. Good performance is indicated by a mean distance ratio greater than 1 and the higher the better, since that would mean performance in which the algorithm rarely makes a wrong guess.

Table 1: Swain's images. Matches of 30 test images for database of 66 model images

Algorithm	Rankings				Next-Best Mismatch	
	1	2	3	> 3	Mean	SD
Color Indexing (L1)	23	2	0	5	1.32	0.49
Color Angles (*L2*)	24	1	2	3	*2.41*	*2.10*
ChromHist-A (L1)	28	2	0	0	1.47	0.36
ChromHist-B1 (*L2*)	26	1	1	2	*2.65*	*2.28*
ChromHist-B2 (*L2*)	27	1	1	1	*2.61*	*1.52*
With Added Noise						
Color Indexing (L1):	22	2	1	5	1.17	0.23
Color Angles (*L2*):	12	4	2	12	*1.02*	*0.71*
ChromHist-A (L1):	27	2	0	1	1.10	0.09
ChromHist-B1 (*L2*):	27	0	2	1	*5.24*	*5.77*
ChromHist-B2 (*L2*):	27	2	0	1	*4.59*	*4.47*

From the results in Table 1 we see that ChromHist-A performs best, and the ChromHist-B2 method, the version that operates entirely in the compressed do-

main, follows not too far behind. The ChromHist method in either the whole–histogram or compressed histogram version is least sensitive to noise. Below we show that for textures or when illumination change is actually present the ChromHist method in any form again does best of methods tested here. [4]

4.2 Textures: modeled illumination change

As another test, consider the model database of ten color texture images in [14]. For this model set, test images are created by modeling illumination change by means of imaging these textures through three separate colored filters. Thus there are a total of 30 test images under different simulated illuminations, for a model database of 10 images. After Finlayson et al. [3], we rotated test images by 30° with respect to the model image orientation before carrying out matching. As a result, the correlation-based method of [14] fails, with match rankings of 13 at rank 1, 7 at rank 2, 1 at rank 3, and 9 at rank > 3. A subsequent correlation-based method [15] addresses rotations, but we have not implemented that extension here.

Narrowband filters should favor the Color Angles method, and we find that this is indeed the case. In Table 2 results are shown for the six algorithms tested. Since it is asserted that image size is relatively unimportant for at least one of these methods [1], for the uncompressed methods we reduced the texture images by pixel averaging from 480 × 640 to 120 × 128.

Table 2: Textures. Matches of 30 reduced, rotated test images for database of 10 model images

Algorithm	Rankings				Next-Best Mismatch	
	1	2	3	> 3	Mean	SD
Correlation ($L2$)	13	7	1	9	*2.42*	*4.83*
Color Indexing (L1)	3	5	2	20	0.90	0.10
Color Angles ($L2$)	19	2	5	4	*1.23*	*0.61*
ChromHist–A (L1)	27	2	1	0	1.66	0.56
ChromHist–B1 ($L2$)	22	3	2	3	*3.29*	*4.73*
ChromHist–B2 ($L2$)	23	2	4	1	*2.33*	*1.75*

As can be seen, again methods ChromHist–B2 and B1 approximate fairly well to using full histograms and again the ChromHist method performs best. Note from the Next–Best Mismatch distance ratio that ChromHist–A performs much more robustly than its one L1–metric competitor. The mean of next–best mismatch values is 1.66 for ChromHist–A and 0.90 for Color Indexing. Thus the ChromHist method is much less likely to choose the wrong model image. [5]

4.3 Objects: true illumination change

To test the indexing algorithms on data that truly reflects changing illumination and not just placing a filter in front of the camera lens, we consider the 13 model images in [3].

Test images consist of images of various objects, rotated, shifted, or wrinkled, taken under two different illuminants (three different illuminants in one case). Thus there are 27 test images for a model database of 13 images. Results for the algorithms tested are given in Table 3.

Table 3: Objects under changing illuminants: Matches of 27 test images for database of 13 model images

Algorithm	Rankings				Next-Best Mismatch	
	1	2	3	> 3	Mean	SD
Color Indexing (L1)	2	6	7	12	0.76	0.35
Color Angles($L2$)	23	3	1	0	*3.48*	*3.64*
ChromHist–A (L1): uniform bins	24	1	2	0	1.24	0.25
ChromHist–A (L1): rand.image bins	23	2	2	0	1.20	0.26
ChromHist–A (L1): hist.-eq. bins	25	0	2	0	1.22	0.22
ChromHist–B1($L2$)	26	1	0	0	*5.00*	*5.57*
ChromHist–B2($L2$)	26	1	0	0	*3.50*	*2.29*
With Added Noise						
Color Indexing (L1):	3	3	1	20	0.56	0.36
Color Angles($L2$):	5	5	2	15	*0.71*	*0.55*
ChromHist–A (L1):	12	5	0	10	0.98	0.08
ChromHist–B1($L2$):	18	3	2	4	*2.09*	*2.32*
ChromHist–B2($L2$):	17	4	1	5	*1.83*	*1.73*

This database is the only one studied that truly utilizes changing illumination and here our method performs best, either in the whole–histogram or compressed–histogram version. As well, the new algorithm stands up best when challenged by noise.

5 Conclusion

The simple idea of normalizing color images separately in each band as a reasonable approach to color constancy preprocessing in the context of indexing into an image database is adopted here, but in this paper we do not throw away most of the information,

[4] Above we do not provide a comparison with the CCCI method [2]. That method selected only the 55 of Swain's 66 model images that had no pixel saturation. This concomitantly reduces the number of test images to 24. We do not preselect out any of the images. However, we did perform runs using just these 55 model images, with no noise, using the ChromHist–A method and also our implementation of the Color Indexing method, for comparison with CCCI. For CCCI, results reported are: 23 matches of rank 1 and 2 of rank 2 out of 24 (*sic*). For this data, the Color Indexing method finds 21 matches with rank 1 and 3 matches with rank 2. Our method, ChromHist–A, finds 22 of rank 1 and 2 of rank 2.

[5] Of course, if test and model images were registered, then the very best one could do would be to model illumination change via a 3 × 3 transformation matrix among color bands; in [12] we computed a Least Squares approximation of such a matrix with excellent results. As well, we showed that wavelet–based compression followed by DCT and truncation did not destroy the linearity of the transformation and therefore that a Least Squares analysis could be carried out entirely in the compressed domain, leading to an indexing scheme with very little degradation from using full images and based on only 63 values.

as in the Color Angles method. Instead, we transform to a 2D representation by using histograms of chromaticity. Viewing these 2D feature space histograms as images, we apply a wavelet–based image reduction transformation for low–pass filtering, a square root operation, and DCT and truncation. The resulting indexing scheme uses only 36 or 72 values to index into the image database. Results for experiments on test images with no change of illumination, illumination change modeled by changing camera filters, and true change of illumination, with and without noise added, show that the method does better or a good deal better than the other methods tested. As well, we find that operating entirely in the compressed domain has an appreciable but acceptable effect on the accuracy of the method, compared to using whole images and no compression. Generally speaking the compression method ChromHist–B2, based on folding a 32×32 histogram into a 32×16 one, does slightly better than method ChromHist–B1, which uses a 16×16 compressed histogram.

Future work could further consider the effect of histogram–equalization on the method. We have found that performing histogram–equalization on individual chromaticity histograms derived from color–normalized color images generally leads to histogram bins that are more or less the same across images, so histogram–equalized bins for each image could be employed. This could also benefit from our ChromHist–B methods in which severe low–pass filtering is applied. Alternatively, the database could be augmented by the best bins for each model image, either with both channels lumped or separately for each chromaticity band, for a resulting overhead of one or two times the number of histogram bins in use. Then a quick culling scheme could be based on whether a test image has similar histogram–equalized bins, as a first pass. Of course, straightforward adaptive schemes such as comparing largest bins first and so on could also be used.

As well, there may be other wavelets that could work better and the issues of optimum histogram resolution and schemes other than the square root for reducing the AC energy could be addressed. Nevertheless, the results reported here are in some cases already quite good, even using uniform bins and a fairly ruthless compression scheme.

The fact that the color–channel normalization step reduces every image to the same scale means that the present method can also be applied to transition detection in video [16]. Because transition thresholds can be predetermined from a training set of videos, new videos or streaming videos can be segmented without having to generate descriptive statistics for the entire video, for each new video. The metric used in this domain is again based on compressed chromaticity histograms.

References

[1] M.J. Swain and D.H. Ballard. Color indexing. *Int. J. Comput. Vision*, 7(1):11–32, 1991.

[2] B.V. Funt and G.D. Finlayson. Color constant color indexing. *IEEE PAMI*, 17:522–529, 1995.

[3] G.D. Finlayson, S.S. Chatterjee, and B.V. Funt. Color angular indexing. In *ECCV96*, pages II:16–27, 1996.

[4] G.D. Finlayson, M.S. Drew, and B.V. Funt. Spectral sharpening: sensor transformations for improved color constancy. *J. Opt. Soc. Am. A*, 11(5):1553–1563, May 1994.

[5] C.F. Borges. Trichromatic approximation method for surface illumination. *J. Opt. Soc. Am. A*, 8:1319–1323, 1991.

[6] G. Healey and A. Jain. Retrieving multispectral satellite images using physics–based invariant representations. *IEEE PAMI*, 18:842–848, 1996.

[7] G. Wyszecki and W.S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulas.* Wiley, New York, 2nd edition, 1982.

[8] S. Lin and S.W. Lee. Using chromaticty distributions and eigenspace analysis for pose-, illumination- and specularity-invariant recognition of 3D objects. In *CVPR97*, pages 426–431, 1997.

[9] G. Healey and L. Wang. The illumination–invariant recognition of color. In *ICCV95*, volume 12, pages 128–133, 1995.

[10] D. Slater and G. Healey. Combining color and geometric information for the illumination–invariant recognition of 3–d objects. In *ICCV95*, pages 563–568, 1995.

[11] M. Antonini et al. Image coding using wavelet transform. *IEEE Trans. on Image Processing*, 1(2):205–221, 1992.

[12] Mark S. Drew, Jie Wei, and Ze-Nian Li. On illumination invariance in color object recognition. Technical Report TR 97-07, Simon Fraser University, 1997.

[13] Mark S. Drew, Jie Wei, and Ze-Nian Li. Illumination–invariant color object recognition via compressed chromaticity histograms of normalized images. Technical Report TR 97-09, Simon Fraser University, 1997.

[14] G. Healey and L. Wang. Illumination–invariant recognition of texture in color images. *J. Opt. Soc. Am. A*, 12:1877–1883, 1995.

[15] L. Wang and G. Healey. Illumination and geometry invariant recognition of texture in color images. In *CVPR96*, pages 419–424, 1996.

[16] J. Wei, M.S. Drew, and Z.-N. Li. Illumination invariant video transition detection by robust thresholding, 1997 (submitted for publication).