Image indexing using compressed colour histograms

J. Berens, G.D. Finlayson and G. Qiu

Abstract: The distribution of colours in an image has proven to be very useful for object recognition. Building on Swain's colour indexing, colour distributions are now an integral part of many recognition schemes. This is not to say that colour alone suffices but rather that colour is one important cue that aids recognition. Colour indexing is performed on colour distribution histograms, and as such, the speed of the system is directly related to the size of the histogram to be indexed. It is shown how colour histograms can be effectively compressed and how compressed colour histograms can be compared for indexing. The authors make two important contributions. First, they show that an opponent colour histogram can be compressed more readily than can conventional colour space. Secondly, they use the standard transform encoding methods (the Karhunen–Loève transform, the discrete cosine transform, the Hadamard transform and hybrid transforms) to compress colour histograms. Experiments show that compression rates of up to 250:1 are possible without affecting indexing performance. This means that a database can be searched that is 250 times larger in the same time as that searched by conventional indexing.

1 Introduction

The distribution of colours in an image is an important cue for image indexing (finding similar images) and for object recognition. Building on Swain's colour indexing, many image database systems support colour content queries [1–5]. Importantly, users find colour-based queries an intuitive and effective mechanism for navigating in an image database [6, 7]. Colour distribution information can be represented in a number of ways: mean RGB co-variant RGB [8], colour clusters [9–11]), and colour names [12, 13]). However, the most general (and arguably the simplest) representation for colour distributions is the colour histogram. It is most general in the sense that colour statistics such as the mean or colour clusters are usually calculated from the colour histogram. Unfortunately, it is argued that there is a downside to this generality: it is relatively more expensive to match or compare colour histograms. In this paper we show that colour histograms contain highly correlated information and so they can be effectively compressed: they can be represented by a few numbers. As such, colour histogram comparison is no slower than any other colour-based indexing method.

Colour histograms are created by partitioning colour space into equi-area regions and counting the number of pixels falling in each region, then allocating that total to the related histogram bin (each histogram has the same number

of bins as there are equi-area regions). We use the notation H to denote a colour histogram and H_i to denote the RGBs falling in the i th region of colour space. The distance between two images I_1 and I_2 is defined as the distance between their respective histograms

$$\|I_1 - I_2\| \equiv \|H_1 - H_2\| \quad (1)$$

The smaller the distance, the closer the similarity of the images.

In considering how colour histograms might be most efficiently encoded, we must first look at how colour itself is represented. For example, image RGBs are susceptible to illumination; if the brightness of a light source illuminating a scene is increased, all the magnitudes of all the RGBs increase by the same factor. That is, under a brightness change the three-dimensional colour histogram also changes and so a naïve three-dimensional partition of colour space often supports poor indexing. To circumvent this problem, image RGBs are mapped to brightness-independent chromaticities prior to indexing [1]

$$\begin{aligned} r &= \frac{R}{R + G + B} \\ \text{chromaticity}(R, G, B) &= g = \frac{G}{R + G + B} \\ b &= \frac{B}{R + G + B} \end{aligned} \quad (2)$$

Because $b = 1 - r - g$, $\text{chromaticity}(R, G, B)$ is usually written as the 2-tuple (r, g) (since b can be calculated from (r, g)). This also has the advantage of reducing the histogram dimensionality to 2-D. It has been shown [1] that indexing on chromaticity histograms is an effective way of removing the brightness ambiguity problem. We investigated the effect that histogram size (the number of bins contained in the histogram) has on colour indexing, and found that for chromaticity histograms, 32 bins per

colour axis gave good results for all our data. Interestingly, the (poor) results for matching on only four bins per axis were almost identical to that of 128 bins per axis. On investigation we found that most of these bins actually contain no pixels, and only a small number of bins contain most of the image pixels. The effect of noise on these finely quantised histograms can be significant, and can result in the non-zero counts being shifted to neighbouring bins. Fig. 1 shows how a small shift in bin registration can result in an error. Noise is therefore a major factor in causing poor results in 128×128 bin histograms. Stricker [2] discusses this problem in more detail.

Dealing with the bin mismatch problem by choosing a small number of bins is only one solution. The ‘earth movers distance’ (EMD) metric [10] explicitly models how data may shift between histogram bins, and so can calculate similarity for fine resolution histograms. Unfortunately the cost of calculating EMD is proportional to the number of bins, so for computational reasons a small bin size is used with the EMD metric. Henceforth, this paper concerns itself only with Euclidian distance.

This paper first shows that the nature of (r, g) chromaticity space makes it an inappropriate colour space for indexing. We show empirically that the co-ordinate system on top of which image formation is predicated impacts on the saliency and compressability of the histogram. Points in the rg chromaticity space were found to be quite correlated, and as such an equi-area partition of colour space is not appropriate (histogram formation assumes an equi-area partition of colour space).

In contrast, the *red–green/yellow–blue* ($r-g, y-b$) colour space better decorrelates colour information and is also more uniform [14, 15], i.e. uniform in the sense that, on average, each bin will capture a similar proportion of colours. In a non-uniform histogram it is likely that a small number of bins will always have a high count and therefore pairs of non-similar histograms will appear similar. It follows that these histograms are more difficult to code since small differences are most important.

We observe that under the assumption of uniformity, two-dimensional histograms share many properties of conventional grey scale images. They are both 2-D lattices. Moreover, the frequency coded in a histogram bin correlates with grey-scale intensity. This idea is sketched in Fig. 2. The image analogy is not only useful for visualisation

Fig. 1 Small shift in bin registration can result in error
a Histograms H_1 and H_2 in registration and the result $H_1 - H_2$
b H_1 and H_2 out of registration and the resultant error
c H_1 and H_2 out of registration and the resultant error

Fig. 2 Transforming RGB image into 2-D opponent colour histogram showing separate R,G and B images, 2-D opponent-colour space representation and distribution histogram

Note how few bins actually contain noticeable information

but it also allows us to treat the data encoding issue (representing histograms using a small set of coefficients) as a problem of image compression.

This paper also shows how opponent colour histograms can be compressed using off-the-shelf image coding techniques. In an extensive set of experiments the performance of discrete cosine transform (DCT), Hadamard transform (HADT), Karhunen–Loève transform (KLT) and hybrid methods are evaluated. All methods work well with the KLT encoding, post DCT compression working slightly better than the other methods. Experiments show that compression rates of up to 250 to 1 are possible without affecting colour indexing performance. The corollary of this result is that one can search a database that is 250 times as large as can be indexed using conventional colour indexing.

2 Background

In eqn. 1 the distance between two images is defined to be the distance between two histograms. Representing histograms as 1-dimensional vectors, we use the notation $\underline{H}$ to denote the one-dimensional vector counterpart of H (H is stretched out as a vector). Revisiting eqn. 1, which defined image similarity as histogram similarity, the distance between histograms H_1 and H_2 is defined to be

$$\|H_1 - H_2\| = \sqrt{\sum_i (\underline{H}_{1,i} - \underline{H}_{2,i})^2} \quad (3)$$

Using the result that $\|\underline{v}\|^2 = \underline{v} \cdot \underline{v}$ we rewrite the distance as

$$\|H_1 - H_2\|^2 = (\underline{H}_1 - \underline{H}_2) \cdot (\underline{H}_1 - \underline{H}_2) \quad (4)$$

The vector dot-product can be written as $\underline{v} \cdot \underline{v} = \underline{v}^t \underline{v}$ (where t denotes transpose) [16] and so

$$\|H_1 - H_2\|^2 = (\underline{H}_1 - \underline{H}_2)^t (\underline{H}_1 - \underline{H}_2) \quad (5)$$

Eqn. 5 is important as it makes the cost of histogram indexing explicit. If there are n bins in a histogram then histogram comparison costs n subtractions plus n squaring operations plus n additions. Now if a database is small and n is not large the cost of indexing though significant is not

intractable. However, it is not uncommon for image databases to have more than a million images and for these large databases indexing, even for a moderate n is a costly operation.

In his original study Swain suggests that around 2000 histogram bins are required for good performance: histogram comparison is a 2000-dimensional vector calculation. Though the cost of this calculation is quite large the figure is, in the context of recently published work, quite conservative. Hafner *et al.* [17] have proposed that the interrelationships between bins are also important in calculating histogram distances and this led them to the more general but more expensive distance measure

$$\|H_1 - H_2\|^2 = (\underline{H}_1 - \underline{H}_2)' \mathcal{Q} \mathcal{Q}' (\underline{H}_1 - \underline{H}_2) \quad (6)$$

where $\mathcal{Q} \mathcal{Q}'$ is a $n \times n$ quadratic form. The matrix $\mathcal{Q}$ is designed so that eqn. 6 calculates the local average of histogram counts (and so eqn. 6 calculates a difference of 'smoothed' histograms). The rationale of the local averaging is that it incorporates perceptual colour metrics: the idea that RGB colour space is not perceptually uniform (the areas in which an observer has greater sensitivity would be smoothed more finely than those where our sensitivity is less). In effect, $\mathcal{Q}$ can be thought of as a linear operator which convolves the histogram image with a smoothing filter where the extent of the filter depends on location in colour space.

One might also imagine that local averaging would be needed if colour space were partitioned too finely (the pathological case discussed earlier) since there would be a need to re-integrate information. Unfortunately, in a general sense, incorporating the quadratic form increases the cost of histogram comparison: there are now $O(n^2)$ operations rather than $O(n)$ (though if the averaging is sufficiently local then the cost is $O(kn)$ [17]).

Eqn. 6 draws attention to the fact that we may wish to modify the distance measure we use if we know something about the data. In Hafner *et al.*'s case the rationale is to account for non-uniform colour spaces. In the following Section we look at eqn. 6 in terms of *a priori* knowledge about the statistics of histograms themselves. We show that by adopting an appropriate $\mathcal{Q}$, the cost of matching is $O(m)$ where $m \ll n$.

However, before progressing we address Hafner *et al.*'s claim [17] (borne out by experiment [10]) that indexing should be carried out using a perceptual colour space. Indeed, it is well known that the (r, g) colour space is not perceptually relevant. Moreover, we also observe that the (r, g) chromaticity space is non-uniform in the sense that certain histogram bins have more counts than others (i.e. more RGB colours map to particular bins than to others). To get around these problems we, in line with other authors [10], choose to histogram the more perceptually relevant *red-green/blue-yellow* opponent chromaticity space. Earlier work [7] demonstrated that this colour space is also more uniform in the sense that opponent histogram bin counts are more evenly distributed. The opponent chromaticities are defined in terms of r, g, b chromaticities:

$$(rg, yb) = \left(r - g, \frac{r+g}{2} - b \right) \quad (7)$$

3 Image compression

Suppose $\mathcal{Q}$ is a $n \times m$ orthonormal matrix ($\mathcal{Q}' \mathcal{Q} = \mathcal{I}$) such that linear combinations of the columns of $\mathcal{Q}$ suffice to describe histograms. Assuming that $m \ll n$ then histogram

comparison is an $O(m)$ operation (i.e. it is much faster). To see that this is so, by assumption we let

$$\underline{H} = \sum_{i=1}^m \mathcal{Q}_i q_i \quad (8)$$

where $\mathcal{Q}_i$ denotes the i th column of $\mathcal{Q}$ and q_i is a scalar weight. If we group these weights into the m -vector $\underline{q}$

$$\underline{H} = \mathcal{Q} \underline{q} \quad (9)$$

Remembering that $\mathcal{Q}' \mathcal{Q} = \mathcal{I}$ (the $m \times m$ identity matrix) and substituting $\underline{H}_1 = \mathcal{Q} \underline{q}_1$ and $\underline{H}_2 = \mathcal{Q} \underline{q}_2$ into eqn. 6

$$\begin{aligned} \|\mathcal{Q} \underline{q}_1 - \mathcal{Q} \underline{q}_2\|^2 &= (\underline{H}_1 - \underline{H}_2)' \mathcal{Q} \mathcal{Q}' (\underline{H}_1 - \underline{H}_2) \\ &\Rightarrow \|\underline{H}_1 - \underline{H}_2\|^2 = (\underline{q}_1 - \underline{q}_2)' (\underline{q}_1 - \underline{q}_2) \end{aligned} \quad (10)$$

Of course, by construction, our choice of quadratic form leads to much quicker histogram comparison since now the distance between a pair of histograms is defined by the distance between histogram weights. Eqn. 10 implies

$$\|\underline{H}_1 - \underline{H}_2\|^2 = \|\underline{q}_1 - \underline{q}_2\|^2 \quad (11)$$

an $O(m)$ operation.

It is apparent then that to speed-up histogram comparison we need only find an appropriate matrix, or basis, $\mathcal{Q}$. To see how to calculate $\mathcal{Q}$ we use an image analogy: the idea that a colour distribution (two-dimensional colour histogram) can be interpreted as an image. It is well known that conventional images are highly correlated. That is to say that locally in an image the values of pixels do not change very quickly. If a pixel has value 100, its neighbours typically have similar values. This property of high correlation allows spatial images to be converted to the frequency domain using transform coding. (The term 'frequency' when used in image processing refers to changes in intensity or colour within the image: low frequencies represent gradual change, high frequencies represent rapid changes). As a result of spatial correlation, the image energy within the transform domain tends to be clustered towards a relatively small number of transform samples, which allows most of the transform samples to be discarded without seriously affecting the image quality on reconstruction (frequency transforms have the property of almost perfect reconstruction, using the inverse transform [18]). Transform coding calculates the linear combination of orthogonal basis functions that best approximates a signal (e.g. $\mathcal{Q} \underline{q}$ in the previous discussion).

The optimal transform is the Karhunen–Loève transform (KLT) [19–21]. The precise details of this transform are not important but it is important to understand the sense in which it is optimal. Suppose we have a set of training histograms $\underline{H}_1, \underline{H}_2, \dots, \underline{H}_w$ and we wish to represent these histograms using a m -dimensional basis $\mathcal{Q}$ such that the following error is minimised:

$$\sum_{i=1}^w \|\underline{H}_i - \mathcal{Q} \underline{q}_i\|^2 \quad (12)$$

where $\underline{q}_i$ is, as before, the m -vector of coefficients which minimises the distance to the i th histogram. The Karhunen–Loeve procedure returns the $\mathcal{Q}$ which minimises the error in expr. 12.

One of the problems of using the KLT is it is based on a training set of data. For very large images it is not really possible to find enough images to find a reliable basis decomposition. Thus, closed-form basis expansions are often used. The most common approach is to use the

Fig. 3 Two-cycle 1-D cosine and its equivalent 1-D Hadamard step waveform

— two-cycle
--- Hadamard

discrete cosine transform (DCT): where images are represented as a linear sum of progressively higher frequency co-sinusoidal basis functions. The utility of the DCT is widely accepted (among other uses, it is at the heart of the JPEG compression standard). Importantly, under quite reasonable statistical assumptions, DCT is in the limit asymptotically equivalent to KLT [20]. Pragmatically, it is also fast and easy to compute [18].

The Hadamard transform (HADT) is also used for image coding and so we also test this transform for coding colour histograms. The Hadamard transform is like the DCT except that the basis functions are composed entirely of +1s and -1s. The higher the order of the Hadamard function the more steps that are made (corresponding to a highly periodic cosine function). In Fig. 3, a two-cycle 1-D cosine is compared with the corresponding Hadamard function. Despite this stepping, the Hadamard expansion still basically encodes frequency information. The first few basis functions usually suffice to code most of the useful low-frequency information. More importantly because only +1 and -1 is used, all the histogram calculations can be carried out as integer operations and so is faster to compute than either DCT or KLT [18–21].

It is sometimes beneficial to combine statistical techniques in image coding. In particular, we might code histograms using a large number of DCT components and then use KLT to code the DCT compressed histograms. The initial DCT compression serves only as a low-pass filter on the image data, resulting in images coded with fewer numbers. It is then easier to find the significant statistics of these coded images using KLT [22].

4 Results

Our first set of experiments was concerned with identifying the optimal bin size for both chromaticity and opponent colour histograms. We used Swain's dataset of 66 images [1] and 31 test images (Swain's objects viewed in different positions). Sample images are shown in Fig. 8. For this dataset, there will always be a correct answer for image indexing. The distance between each test distribution image and all distribution images in the database is calculated. The smallest distance overall is used to define

Table 1: Average match percentile performance for indexing on chromaticity and opponent colour histograms of varying bin sizes, using Swain's database

#Bins	Rank					AMP%
	1	2	3	>3	Worst	
Chromaticity space						
4 × 4	10	9	3	9	13	96.38
8 × 8	20	3	2	6	12	98.01
12 × 12	22	2	3	4	10	98.71
16 × 16	21	5	1	4	11	98.56
32 × 32	24	1	2	4	6	98.96
64 × 64	20	6	1	4	17	97.02
128 × 128	16	5	3	7	23	96.08
Opponent colour space						
4 × 4	15	3	2	11	37	92.16
8 × 8	23	2	1	5	7	98.56
12 × 12	23	3	3	2	11	98.71
16 × 16	27	2	0	2	4	99.60
32 × 32	25	0	0	6	43	95.98
64 × 64	21	2	2	6	41	94.24
128 × 128	20	5	2	4	13	98.46

the identity of the test object. Recognition is not perfect and sometimes the 'correct' answer is actually the second, third or even higher ranked smallest distance. Thus, we calculated the rank of the correct answer in the sorted list of distribution distances. The results are summarised according to the average match percentile (AMP) measure, which gives an indication of the overall performance of all test images against all database images. The results are listed in Table 1.

These results are very similar to those achieved by Swain on similar sized histograms [1]. The first part of Table 1 shows that chromaticity histograms are generally robust to quite large changes in bin sizes, but that indexing

Fig. 4 Swain's dataset: 66 database images and 31 queries

— average match percentile of KLT basis dimensions
--- average match percentile of DCT bases
..... average match percentile for HADT bases

Fig. 5 Dataset: 240 database images and 100 queries

— average match percentile of KLT basis dimensions
 --- average match percentile of DCT bases
 average match percentile for HADT bases

Fig. 6 Swain's dataset: 66 database images and 31 queries

--- average match percentile for KLT encoding of 66 DCT basis dimensions
 average match percentile for a KLT encoding of 66 HADT bases
 — average match percentile for KLT basis dimensions

Fig. 7 Dataset: 240 database images and 100 queries

--- average match percentile for KLT encoding of 66 DCT basis dimensions
 average match percentile for a KLT encoding of 66 HADT bases
 — average match percentile for KLT basis dimensions

performance is poor; the best match was achieved using the 32×32 bin histogram, but with an AMP of only 98.56%, far from acceptable when considering the small database size. Interestingly, the overall performance of the finely partitioned 128×128 bin histogram was similar to that of the 4×4 histogram. This bears out our earlier assertion that noise can have severe detrimental effects on the coding of finely partitioned histograms.

The second part of Table 1 shows the results for the opponent colour histograms. Clearly this colour space is highly susceptible to changes in bin size, although the number of first rankings was far higher than that for the chromaticity histograms, the worst rank matches were generally far worse, a factor which reduced the AMP significantly. Importantly though, the results of using the 16×16 bin histogram achieved very good results: 99.60% AMP, which matches the results achieved by Swain using $16 \times 16 \times 8$ opponent colour histograms (he codes the intensity values as an 8 bin vector). In other words, we replicated Swain's results for 2048 bins using only 256 bins.

We examined the performance of colour indexing when 16×16 bin opponent chromaticity histograms are compressed using KLT, DCT and HADT. Indexing results are summarised in Fig. 4. In all cases as the number of components used increases so the indexing accuracy also increases. This is as we might expect. As we add more components to our image coding so we better reconstruct the original histogram. Matching on DCT and HADT coefficients reaches its maximum efficiency at 66 coefficients, however, rather remarkably, even when very few KLT components are used we achieve excellent indexing. Using just eight KLT components (i.e. coding histograms with eight numbers) we achieve 99.65% match percentile, a marginal improvement over Swain. Relative to the 16×16 distribution image, an eight-number image coding implies a $256/8 = 64$ times speed-up. However, the recognition rate that we achieved is slightly better than that achieved by Swain using a 2048 bin three-dimensional colour histogram. Relative to that representation, the compressed opponent distribution image facilitates a $2048/8 = 256$ factor speed-up.

While these results are interesting, they relate to a rather small image database, and so we increased the size of our database by adding a further 174 digitised photographic images and 69 cropped and rotated test images, increasing our database to 240 images and 100 query images (a selection of sample images from this database are displayed in Fig. 8). Though a 240 image database is still relatively modest in size it will allow us to examine the scalability of our method. We found the results did scale extremely well to the expanded database. These results are summarised in Fig. 5.

As a final experiment we combined the KLT and DCT and the KLT and HADT methods to see if a hybrid approach improved results still further [22]. Here the DCT and HADT transforms are applied very softly: many components are allowed. Then at a second stage, KLT is performed on what is a basically low-to-medium-pass filtered opponent colour chromaticity image. Results are summarised in Fig. 6 and 7. We do see a slight, though not really significant improvement. Yet we predict that the power of the hybrid approach may be useful in larger image databases.

Table 2 summarises the key results from all these experiments as well as the results for matching on uncompressed histograms.

Fig. 8 Selection from images used in experiments

Top: Swain's database images
2nd row: Swain's test images
3rd row: photographic database images
4th row: photographic test images

Table 2: Summary of key indexing results

Compression method	Rank					AMP %
	1	2	3	>3	Worst	
Using Swain's database						
None (histogram only)	27	2	0	2	4	99.60
66 DCT coefficients	28	1	1	1	5	99.65
66 HADT coefficients	28	1	1	1	4	99.70
8 KLT only	22	5	0	4	5	99.10
16 KLT only	29	1	0	1	4	99.80
8 KLT encoding of 66 DCT	27	3	0	1	5	99.65
8 KLT encoding of 66 HADT	24	2	2	3	4	99.26
Using photographic database						
None (histogram only)	95	3	0	2	6	99.95
66 DCT coefficients	94	3	0	3	8	99.92
66 HADT coefficients	92	4	1	3	21	99.86
8 KLT only	76	13	2	9	43	99.57
16 KLT only	86	9	1	4	20	99.81
8 KLT encoding of 66 DCT	85	11	1	3	14	99.86
8 KLT encoding of 66 HADT	80	9	4	7	30	99.69

5 Conclusion

Colours, minus intensity, are mapped in a uniform 2-D opponent colour space and this space is interpreted as a grey-scale image; the brightness of each pixel correlates with its frequency of occurrence in the original image. By comparing the distance between colour distribution images, an image database can be searched and images similar to a query image found (similar in the sense that they have similar colour content).

We have shown that colour distribution images are highly correlated and so are effectively compressed using off-the-shelf transform coding methods: the Karhunen-Loève transform, the discrete cosine transform and the Hadamard transform. In this framework the cost of colour histogram comparison is proportional only to the size of the compressed image. We found that colour distributions

could be coded with as few as eight coefficients without loss of indexing accuracy. The best compression method found was based on a KLT encoding of a soft DCT coding of an image. Soft in the sense that many DCT components were allowed (the application of the DCT served only to remove the highest frequency information). Indexing on compressed colour histograms is up to 256 times faster than without compression, and without loss of accuracy.

6 Acknowledgment

This work has been funded by EPSRC award ER L60852.

7 References

- SWAIN, M.J., and BALLARD, D.H.: 'Color indexing', *Int. J. Comput. Vis.*, 1991, 7, (1), pp. 11–32
- STRICKER, M.A., and ORENGO, M.: 'Similarity of color images', *Proc. SPIE - Int. Soc. Opt. Eng.*, 1995, 2420, pp. 381–392
- NIBLACK, W., and BARBER, R.: 'The QBIC project: Querying images by content using color, texture and shape', *Proc. SPIE - Int. Soc. Opt. Eng.*, 1993, 1908, pp. 173–187
- LIN, S., and LEE, S.W.: 'Using chromaticity distributions and eigen-space analysis for pose-, illumination- and specular-invariant recognition of 3d object', *Proceedings of CVPR97*, 1997, pp. 426–431
- WEI, J., DREW, M.S., and LI, Z.: 'Illumination invariant video segmentation by hierarchical robust thresholding', *Proc. SPIE - Int. Soc. Opt. Eng.*, 1997, 3312, pp. 188–201
- FINLAYSON, G.D., BERENS, J., and QIU, G.: 'Image indexing using compressed colour histograms', in 'The challenge of image retrieval' 1999
- FINLAYSON, G.D., BERENS, J., and QIU, G.: 'A statistical image of colour space'. Presented at the 7th IEE international conference on *Image processing and its applications*, 1999
- FINLAYSON, G.D., CHATTERJEE, S.S., and FUNT, B.V.: 'Colour angular indexing'. *Proceedings of the European conference on Computer vision*, April 1996, pp. 16–27
- UCHIYAMA, T., and ARBIB, M.A.: 'Color image segmentation using competitive learning', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1994, 16, (12), pp. 1197–1206
- RUBNER, Y.: 'Perceptual metrics for image database navigation'. PhD thesis, Dept. of Computer Science, Stanford University, 1999
- SELIM, S.Z., and ISMAIL, M.A.: 'K-means-type algorithms: A generalized convergence theorem and characterization of local optimality', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1984, PAMI-6, (1), pp. 81–87
- SYEDA-MAHMOOD, T.F.: 'Data and model-driven selection using color regions', *Int. J. Computer Vis.*, 1997, 21, (1/2), pp. 9–36
- MEHTRE, P.M., KANKANHALLI, M.S., NARASIMHALU, A.D., and MAN, G.C.: 'Color matching for image retrieval', *Pattern Recognit. Lett.*, 1995, 16, pp. 325–331
- HURVICH, L.M., and JAMESON, D.: 'An opponent-process theory of color vision', *PsychR.*, 1957, 64, 1957
- FAUGERAS, O.D.: 'Digital color image processing within the framework of a human visual model', *IEEE Trans. Acoust. Speech Signal*

- Process.*, 1979, **ASSP-27**, pp. 380–393
- 16 GOLUB, G.H., and VAN LOAN, C.F.: ‘Matrix computations’ (John Hopkins University Press, 1983)
 - 17 HAFNER, J., SAWHNEY, H.S., EQUITZ, W., FLICKNER, M., and NIBLACK, W.: ‘Efficient color histogram indexing for quadratic form distance functions’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 1995, **17**, (7), pp. 729–736
 - 18 PRATT, W.K.: ‘Spatial transform coding of color images’, *IEEE Trans. Commun. Technol.*, 1971, **Com-196**, pp. 980–992
 - 19 ROSENFELD, A., and KAK, A.C.: ‘Digital picture processing’ (Academic Press, 1982)
 - 20 ELLIOTT, D.F., and RAO, K.R.: ‘Fast transforms algorithms, analyses, applications’ (Academic Press, 1982)
 - 21 PRATT, W.K., KANE, J.K., and ANDREWS, H.C.: ‘Hadamard transform image coding’, *Proc. IEEE.*, 1969, **57**, (1), pp. 58–68
 - 22 ZHAO, W.Y., CHELLAPPA, R., and NANDHAKUMAR, N.: ‘Empirical performance analysis of linear discriminant classifiers’. Proceedings of CVPR98, 1998, pp. 164–170