Gamma Comprehensive Normalisation

Graham Finlayson and Ruixia Xu
School of Information Systems, University of East Anglia
Norwich, United Kingdom
graham@sys.uea.ac.uk

Abstract

The light reflected from an object depends not only on surface reflectance of the object but also on lighting geometry and illuminant color. As a consequence, the raw color recorded by a camera is not a reliable cue for color based tasks such as object recognition and tracking. One solution to this problem is to find color invariants which are independent of illumination. While many invariant functions cancel out dependency due to geometry and light color it is less easy to remove both dependencies. The comprehensive normalisation removes both geometry and color but at the cost of an iterative procedure. In earlier work we showed how the need for iteration could be removed by carrying out normalisations in the log color domain. However, we have found that both these normalisations, though theoretically sound, do not account for all dependencies that might realistically be present.

Indeed, in image processing pipelines it is common to raise an image to the power of gamma either to change the contrast (see into shadows or highlights) or to account for display non-linearities. In this paper we ask, “for systems to which gamma functions are applied, how can we make the invariant approach work to facilitate color based object recognition?” Clearly we need to deal with gamma and develop a framework where gamma is removed. This is the major contribution of this paper. We show how a simple extension of the log normalisation strategy also suffices to remove gamma. We tested our method both on linear and nonlinear datasets. While producing similarly results for linear dataset as our previous methods, our new method significantly outperformed previous methods for the nonlinear dataset.

1. Introduction

Color is an important cue for object recognition and often used as an indexing feature for image databases.[2,6,17,19] However, object recognition experiments based on the raw RGB values fails when either lighting geometry or illuminant color changes. The reason for this is that color is dependent not only on surface reflectance, but also on lighting geometry and illuminant color.

Two approaches are reported to circumvent this problem: the color constancy approach[7,10,13,16,21] and color illuminant invariant approach.[1,3,4,8,14,18,22] Color constancy approach attempts to recover surface reflectance (or more precisely correlates of surface reflectance) and this is, by definition, independent of lighting conditions. Unfortunately, color constancy has proved to be a very hard problem to solve. The current state of the art can only deliver approximate constancy. Moreover, the work of B.V.Funt et al.[21] showed that this approach is not good enough to render color a stable enough cue for object recognition. The second approach is to index on invariant functions of RGBs. These functions are usually non-linear combinations of proximate pixels which are designed to factor out dependency due to illumination.

While many illuminant invariants functions can remove either color dependency due to lighting geometry or color dependency due to illuminant color, the comprehensive normalization method[8] was the first to cancel both. But, here invariance was achieved through an iterative procedure. In recent work we revisited the comprehensive normalization and showed that comprehensive normalization might be achieved without iteration so long as the analysis is carried out in log RGB space. Specifically, we showed that if we subtract the mean of each log RGB triple from each pixel and then subtract the mean log R, log G and log B (for the whole image) from each color channel that the result is independent of light color and light geometry.

While experiments showed that this normalization worked well (it facilitated extremely good indexing performance) it is based on the assumption that camera response is linear. However, a gamma function (the image is raised to the power gamma) is often applied in real image coding. For example, because PC monitors apply a gamma of 2.2, the reciprocal (1/2.2) is usually applied to RGBs prior to display. That is to say that stored images are rarely linear. Moreover, Apple systems apply a gamma of 1.8 implying that images of the same scene may differ according to the display system in use. Moreover, gamma is commonly applied to change the relative contrast in an image. Applying a gamma of less than one tends to bring out detail in darker regions (at the expense of the lighter regions) and conversely a gamma larger than one is used to bring out detail in the highlights. Simply put, images of the same scene taken with different cameras can differ in terms of the lighting conditions and the gamma employed.

In this paper we show how the simple log normalization scheme can be extended to remove gamma dependency. There are two steps to **log gamma**

normalisation. First, we, as before, remove lighting geometry and light color by subtracting pixel means then color channel means. We then observe that x^γ maps to $\gamma \ln x$ when logarithms are applied. It follows that if we divide each *R, G* and *B* by their respective standard deviations that γ will cancel.

The gamma normalisation was tested in the context of color object recognition experiments. We found that for the dataset from different devices where various nonlinearities were applied,[11] that log gamma normalization significantly out performed the non gamma invariant, log normalisation. For linear image data set the new log gamma normalisation delivered comparable performance.

The rest of the paper is organised as follows: the basis of color image formation is introduced and color image normalisation methods are reviewed in Section 2. The new normalisation method is presented in Section 3. In section 4 the object recognition experiments are described and results presented. The paper finishes with conclusions in section 5.

2. Background

In order to develop the theory we will adopt some widely used simplifying assumptions. First, we assume that the response of a color camera is linear. That is, if we view a surface under a given light and then double the intensity of the light we expect a doubling in the recorded RGB values. If we denote the ith pixel in an image (R_i, G_i, B_i) then as the power of the incident illumination changes then

$$(R_i, G_i, B_i) \rightarrow ([\rho_i R_i, \rho_i G_i, \rho_i B_i) \quad (1)$$

where ρ_i is a simple scalar. Note that this scalar has a subscript *i* indicating that all pixels can have their own individual brightness factors. Brightness changes, or lighting geometry, is a local phenomenon.

Changing the relative position of the light source with the surface introduces shading. Assuming matte Lambertian reflectance and letting $\underline{n}$ denote surface normal and $\underline{e}$ the lighting direction then the power of the light striking a surface is proportional to the scalar $\underline{n}.\underline{e}$ (the vector dot-product). It follows then that a change in shading can also be described according to Equation(1). It is important to note that the Lambertian assumption is important here. Equation (1) cannot account for lighting geometry changes for highly specular surfaces.

Let us now consider a change in lighting color (assuming lighting geometry is held fixed). In almost all circumstances, Equation (2) approximately holds[23] (or can be made to hold[9]).

$$\begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \rightarrow \begin{pmatrix} x & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & z \end{pmatrix} \begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \quad (2)$$

where *x, y* and *z* are scalars. Note these scalars do not depend on the pixel (there is no subscript *i*). That is color change is a global phenomenon affecting the whole image.

In this paper we assume that a gamma function is applied to each R, G and B channel. That is R, G, and B are raised to the same power gamma:

$$(R, G, B) \rightarrow (R^\gamma, G^\gamma, B^\gamma) \quad (3)$$

Notice that the same gamma is applied to all pixels in the image. Now combining (1), (2) and (3) we see that:

$$(R_i, G_i, B_i) \rightarrow ([x\rho_i R_i]^\gamma, [y\rho_i G_i]^\gamma, [z\rho_i B_i]^\gamma) \quad (4)$$

To simplify matters we incorporate into the scalars

$$(R_i, G_i, B_i) \rightarrow ([x\rho_i R_i]^\gamma, [y\rho_i G_i]^\gamma, [z\rho_i B_i]^\gamma) \quad (5)$$

3. Log Gamma Normalization

It is clear in RGB space that as the lighting conditions change, the effect on a pixel is multiplicative. In log RGB space, multiplication is turned into addition. As we presented in previous paper, this simple conceptual step eventually leads us to an idempotent (non-iterative) normalization. More importantly here, we track γ through the normalization stages and see that the log normalisation renders gamma simple to remove.

Let $r = \ln[x\rho_i R_i]^\gamma$, $g = \ln[y\rho_i G_i]^\gamma$ and $b = \ln[z\rho_i B_i]^\gamma$. Then we can rewrite (5) as:

$$(r_i, g_i, b_i) \rightarrow (x + \rho_I + \gamma r_i, y + \rho_i + \gamma g_i, z + \rho_i + \gamma b_i) \quad (6)$$

$$(r_i, g_i, b_i) \rightarrow (x, y, z) + \rho_i(1, 1, 1) + (\gamma r_i, \gamma g_i, \gamma b_i) \quad (7)$$

In (7) the dependencies we wish to remove are clear. Every pixel is translated by the same illuminant color vector (*x, y, z*) and by a second pixel dependent translation $\rho_i(1, 1, 1)$. We achieve lighting independence if we can remove both these translation terms.

Equation (7) tells us in log RGB space, lighting geometry changing only affects the length of the log rgb vector in the direction of *U* = (1, 1, 1). That is, the directions orthogonal to (1, 1, 1) are unaffected by brightness change. It follows that we can normalize a log rgb to remove brightness by projecting it onto the 2-dimensional space which is orthogonal to the line that spanned by *U*. We can do this by applying some simple results from linear algebra. We define a 3 x 3 projection matrix *Pr* for the space spanned by *U*, and a complementary projection matrix [*I* - *Pr*] for the space which is orthogonal to the space spanned by *U* (where I denotes the 3 x 3 identity matrix). By definition these matrices have the property that $Pr * (1, 1, 1)^t = (1, 1, 1)^t$ and $[I - Pr] * (1, 1, 1) = (0, 0, 0)^t$[15, 5]:

$$\Pr = U'(UU')^{-1}U = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix} \tag{8}$$

and so

$$I - \Pr = \begin{pmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{pmatrix} \tag{9}$$

If we write a log rgb-vector as $\underline{v} = u * (1, 1, 1) + v * (-1, 1, 0) + w * (1, 1, -2)$, it is straightforward to show that $Pr\underline{v} = u * (1, 1, 1)$ and $[I - Pr]\underline{v} = v * (-1, 1, 0) + w * (1, 1, -2)$. Looking at the structure of matrix $[I - Pr]$, we can see that the meaning of the matrix multiplication is that we subtract the mean log rgb from the *r, g* and *b* values:

$$\begin{pmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{pmatrix}\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \frac{2a}{3}-\frac{b}{3}-\frac{c}{3} \\ \frac{2b}{3}-\frac{a}{3}-\frac{c}{3} \\ \frac{2c}{3}-\frac{a}{3}-\frac{b}{3} \end{pmatrix}$$

$$\begin{pmatrix} \frac{2a}{3}-\frac{b}{3}-\frac{c}{3} \\ \frac{2b}{3}-\frac{a}{3}-\frac{c}{3} \\ \frac{2c}{3}-\frac{a}{3}-\frac{b}{3} \end{pmatrix} = \begin{pmatrix} a-\frac{a+b+c}{3} \\ b-\frac{a+b+c}{3} \\ c-\frac{a+b+c}{3} \end{pmatrix}$$

That is, we can remove dependency on lighting geometry by subtracting the mean log response (a ‘brightness’ correlate) from each pixel.

The effect of illuminant color can be removed in a similar way. However, rather than dealing with log rgb vectors we must operate on the vector of all log red responses (or log green responses or log blue responses). From (7) we can write

$$(x + \gamma r_1, x + \gamma r_2, \cdots, x + \gamma r_n) \rightarrow (r_1\gamma, r_2\gamma, \cdots, r_n\gamma) + x(1, 1, \cdots, 1)$$

It follows that the following projection matrices (which are n x n for an n pixel image) will respectively project a color channel in the direction of all ones $(1, 1, \cdots, 1)$ or the space orthogonal to this direction.

$$Pc = \begin{pmatrix} \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \vdots & \vdots & & \vdots \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \end{pmatrix} \tag{10}$$

$$I - Pc = \begin{pmatrix} \frac{n-1}{n} & -\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n} & \frac{n-1}{n} & \cdots & -\frac{1}{n} \\ \vdots & \vdots & & \vdots \\ -\frac{1}{n} & -\frac{1}{n} & \cdots & \frac{n-1}{n} \end{pmatrix} \tag{11}$$

Again, by inspecting the structure of the projection matrix, we can see that to implement this normalisation we only need to subtract the mean red log value from all log red pixel values and subtract the mean log green and log blue values from the log green and log blue pixel values.

To remove lighting geometry and illuminant color both at the same time we just apply the projectors (9) and (11) to the log image. This operation is easy to write down mathematically if we think of an n pixel image as and n x 3 matrix of log rgbs. Denoting this image Y, we can write down an explicit equation for the application of the lighting geometry and light color normalizations:

$$Y_{normalised} = (I - Pc)Y(I - Pr) \tag{12}$$

here $Y_{normalised}$ represents the normalised image. Though, it is important to remember that to implement (12) we simply subtract row means from rows and then column means from columns.

From projection theory, we know that matrix $[I - Pr]$ and $[I - Pc]$ are both idempotent. That is to say that $[I - Pr][I - Pr] = [I - Pr]$ and $[I - Pc][I - Pc] = [I - Pc]$. It follows then that removing shading or light color once, removes it completely:

$$Y_{normalised} = (I - Pc)(I - Pc)Y(I - Pr)(I - Pr) = (I - Pc)Y(I - Pr) \tag{13}$$

Notice that in the arguments set forth we did not worry too much about γ. The reasons for this should be clear. In log space γ is a multiplicative factor. If we multiply Y in (13) by γ then this does not affect the operations of the projectors involved. Let us denote the normalized image matrix as

$$y = (I - Pc)[Y\gamma](I - Pr) \tag{14}$$

where the role of γ is made explicit. By definition the mean of all the elements in y must be zero (if the mean of the rows and columns are individually 0 then the overall mean must also be 0). It follows then that γ cannot be removed by dividing by the mean. Rather we must use a second order statistic. It is easy to show that the variance of all the elements in y can be calculated as:

$$\sigma^2(y) = \frac{trace(y^t y)}{3N} \tag{15}$$

where y has N rows and 3 columns (there are N pixels in the image) and trace() is the sum of the diagonal elements of a matrix. It follows that gamma can be removed by dividing through by the standard deviation.

$$\frac{y}{\sigma(y)} = \frac{\gamma(y)}{\gamma\sigma(y)} = \frac{y}{\sigma(y)} \tag{16}$$

Finally, we have equation,

$$\frac{Y}{\sigma(Y)} = \frac{\gamma Y}{\gamma\sigma(Y)} = \frac{Y}{\sigma(Y)} \tag{17}$$

which shows a gamma only normalisation. Equation (17) is useful as we are interested in quantifying the affect of gamma and a diagonal model on image indexing (together or separately).

In summary we take the log of the RGB image. At each pixel we subtracted the pixel mean (of the log R, G and B responses). We then subtracted the mean of all the resulting red responses from each pixels and the mean green and blue channel responses for the red and green color channels. The result is an image independent of the light color and lighting geometry. Further dividing by the standard deviation of the resultant data makes the representation independent of gamma.

4. Object Recognition Experiments

We now wished to test the log gamma normalization procedure as a preprocessing step for color object recognition. To do this we take an image, carry out a log-gamma normalisation, and then build its color histogram. This histogram is then compared with normalized histograms stored in a database and the closest overall is found. Because, the database contains images (and their histograms) of the same objects for which we have query images, the closet histogram match can be used to identify the query.

The first dataset we used is a composite dataset which is composed of Swain and Ballard[20] image set (66 database images 32 queries), the *legacy* Simon Fraser image set[12] (11 database images, 22 queries) and the Berwick and Lee image set[3] (10 database images and 10 queries). A second larger Simon Fraser image dataset with 20 objects imaged under 11 illuminations is also used as a test set. These two datasets have a linear response and so, in testing these, we were interested in whether the log gamma normalization might degrade the matching performance (since the more dependencies we cancel out the less information is left for matching). A third dataset comprises 28 designs captured under 3 lights and using 4 cameras and 2 color scanners.[11] The color response of these devices is highly non linear and it has been shown that a large variety of existing ’linear’ normalizations do not support image indexing. We hope that accounting for gamma in addition to light color and light geometry will lead to better indexing.

Tables 1 and 2 summarize the indexing performance for comprehensive normalisation, log comprehensive normalization and the new log-gamma normalisation operating on the first and second dataset. Performance is measured using the average match percentile match[20] If the closet database histogram to the query is the correct answer (both corresponding images are of the same object) then the correct answer is found in rank 1. If the correct answer is the kth closest then the correct answer has rank *k*. The corresponding percentile is calculated as

$$\frac{N-k}{N-1},$$

where N is the image number of the model dataset. In Tables 1 and 2 we also tabulat the % of matches in ranks 1 together with the worst case rank statistic. It is clear that for data sets 1 and 2 all three normalisations work well. Discounting gamma has not resulted in a significant fall in performance.

Table 1. Indexing Performance of Composite Dataset (Ranks are % of the Dataset)

Methods	percentile	Rank 1	Worst rank
Log normalisation	99.71	95.38	4 out of 87
comprehensive	99.71	92.31	3 out of 87
Log gamma normalisation	98.60	84.62	19 out of 87

Table 2. Indexing Performance on Large Simon Fraser Dataset (Ranks are % of the Dataset)

Methods	percentile	Rank 1	Worst rank
Log normalisation	99.13	91.00	5 out of 20
comprehensive	98.84	91.00	16 out of 20
Log gamma normalisation	97.61	84.50	10 out of 20

We examine the third design dataset in more detail. The whole dataset consists of 28 designs captured with four different cameras under three different illuminants and also with two different scanners: a total of 4 * 3 * 28 + 2 * 28 = 392 images. First we choose images of the 28 designs captured with a camera under a fixed illuminant as our database. Next we take all other images in the dataset as our query (test) set. Prior to indexing we introduced an ordering into this dataset. Specifically we asked how well a diagonal model of illuminant change together with a gamma correction accounted for the differences in color distributions for corresponding designs in the database and test set. So, given an image in the test set and its corresponding image in the database we find, by an optimization process, the parameters in Equation 5 which best transform the query image to the database image. The error in this transformation then gives us a measure of how well the model holds for this test image. We repeat this process for all test images and order the images according to this error. This ordering allowed us to choose the images in the test set for which a diagonal gamma color model worked with a certain degree of accuracy. For instance we found that the diagonal gamma model of color change accounted for about 20% of the test set with an error of less than 10%.

With this ordering defined we next ran a color indexing experiment as described previously: each image in the test set is matched to the database using a color histogram comparison where histograms are formed from normalized images. Now, rather than looking at matching results for the whole set of query images we investigate match performance as a function of how well the test images correspond to the diagonal-gamma model. So, for example we can investigate match performance for the 10% of images for which the model works best. These results are summarized in Figure 1. The *x*-axis of this figure denotes the proportion of test images for which matching performance is investigated and the corresponding match performance (average match percentile) is shown on the *y*-axis. So, for example we see that if we look at 0.5 on the *x*-axis (corresponding to the 50% of the test set for which a diagonal gamma model works best) the log gamma normalization delivers a percentile match of about 0.93 compared with 0.88 and 0.82 respectively for the comprehensive and log-normalisations.

From Figure 1 it is clear that the log gamma normalization works significantly better than the log normalisation, comprehensive normalisation, or gamma normalization only. By incorporating gamma into the invariant model we have improved matching performance. However, the overall match percentiles can be quite low. If we have a database where the average match percentile is 0.95 then this means that the correct answer is in the top 5% of matches. This might be a tolerable number for some applications. However, percentiles of 0.9 or even 0.8 are indicative of rather poor matching performance. From the figure we see that at the 0.95 level that comprehensive normalization delivers adequate matching only for about 10% of the test set. This increases to about 18% for the log normalisation and about 33% for the log-gamma procedure.

The import of this is that many of the test images cannot be normalised using the non-iterative plus gamma normalization (in that the normalised images do not form a stable cue for indexing). The reason for this is readily understood at an average percentile of 0.95 the diagonal gamma model of image formation models the data with an error of 20%. That is, if we find by optimization the best diagonal matrix and best gamma that maps the database image colors as close as possible to the test images there is a residual error of 20%. Empirically, an error of above 20% is too high to afford good matching.

5. Conclusion

We have presented a new normalisation scheme which can cancel out the color dependencies due to lighting geometry, illuminant color, and gamma functions. Experimental results showed that this normalisation scheme performed similarly when images are captured with linear response cameras but better when confronted with images which have non-linearities.

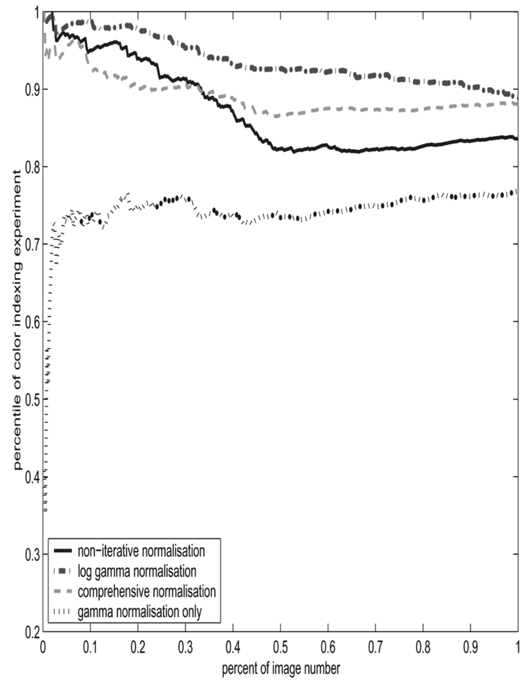

Figure 1. This figure shows performance of each of the four normalisation methods(log gamma, non-iterative, comprehensive normalisation, gamma normalisation only) on the design dataset. The x-axis corresponds to the proportion of query images for which average match percentile (y-axis) is investigated. The query set is sorted according to how well the images conform to a diagonal-gamma model.

References

1. Donald A. Adjeroh and M. C. Lee, On Ratio-Based Color Indexing. *IEEE transaction on image processing*, vol**(10)**, 2001.
2. J. Bach, C. Fuller, A. Gupta, and A. Hampapur, The virage image search engine: An open framework for image management. In *SPIE Conf. on Storage and Retrieval for Image and Video Databases*, volume **2670**, pages 76-87, 1996.
3. D. Berwick and S. W. Lee, *A chromaticity space for specularity for specularity-, illumination color- and illumination pose-invariant 3-d object recognition*. ICCV98, Session 2.3, 1998.
4. G. Buchsbaum, A spatial processor model for object colour perception. *Journal of the Franklin Institute*, **310**:1-26, 1980.
5. F. Chatelin. *Eigenvalues of Matrices*. Wiley, 1993.
6. G. D. Finlayson, *Colour Object Recognition*. Simon Fraser Univ. School of Computing Science, 1992.
7. G. D. Finlayson, Color in perspective. *IEEE Trans. PSMI*, **18**(10):1034-1038, 1996.
8. G. D. Finlayson, Bernt Schiele, and James L. Crowley, Comprehensive Colour Image Normalisation. In *5th European Conference on Computer Vision*, page 475- 490, 1998.
9. G. D. Finlayson, M. S. Drew, and B. V. Funt, Spectral sharpening: Sensor transformations for improved color

constancy. *J. Optical Society of America A*, **11**(5), 1553-1563, 1994.

10. M. D. Zmura and G. Iverson, Color constancy, basic theory of two-stage linear recovery of spectral description for lights and surfaces. *JOSA*, **10**:2148-2165, 1993.
11. G. D. Finlayson and G. Schaefer, Colour indexing across devices and viewing conditions, *2nd Int. Workshop on Content-Based Multimedia Indexing, Brescia*, pages 215–221, 2001.
12. B. V. Funt, K. Barnard, and L. Martin, Is machine colour constancy good enough, in *The Fifth European Conference on Computer Vision (Vol II)*, pages 445–459. European Vision Society, 1998.
13. Ron Gershon, Allan D. Jepson, and John K. Tsotsos, From [R,G,B] to Surface Reflectance: Computing Color Constant Descriptors in Images. In *International Joint Conference on Artificial Intelligence*, pages 755-758, 1987.
14. G. Gevers and A. W. M. Smeulders, Color based object recognition, *Pattern Recognition*, **32**:453-464, 1999.
15. G. H. Golub and C. F. van Loan, *Matrix Computations*, John Hopkins U. Press, 1983.
16. E. H. Land, The retinex theory of color constancy, *Scientific American*, pages 108-129, 1977.
17. M. J. Swain and D. H. Ballard, Color indexing, *International Journal of Computer Vision*, **7**(11), 11-32, 1991.
18. S. Lin and S. W. Lee, Using chromaticity distributions and eigen space analysis for pose-, illumination and specularity-invariant recognition of 3d object, in *CVPR97*, pages 426-431, 1997.
19. W. Niblack and R. Barber, The QBIC project:Querying images by content using color, texture and shape, in *Storage and Retrieval for Image and Video Databases I*, volume **1908** of SPIE Proceedings Series, 1993.
20. M. J. Swain and D. H. Ballard, Color indexing, *International Journal of Computer Vision*, **7**(11), 11–32, 1991.
21. B. V. Funt, K. Barnard, and L. Martin, Is colour constancy good enough? In *5th European Conference on Computer Vision*, pages 445-459, 1998.
22. R. W. G. Hunt *The Reproduction of Color*, Fountain Press, 5th edition, 1995.
23. J. A. Worthey and M.H. Brill, Heuristic analysis of von Kries color constancy. *Journal of The Optical Society of America A*, 3(10), 1708–1712, 1986.