NOVEL METHODS FOR EXTRACTING ILLUMINATION-INVARIANT IMAGES AND FAST CLASSIFICATION OF TEXTURES

by

Muntaseer Salahuddin
B. Sc., Independent University, Bangladesh, 2005

a Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Summer 2009

APPROVAL

Name: Muntaseer Salahuddin

Degree: Master of Science

Title of Thesis: Novel Methods for Extracting Illumination-Invariant Images and Fast Classification of Textures

Examining Committee: Dr. Brian Funt
Chair

____
Dr. Mark S. Drew, Co-Senior Supervisor

____
Dr. Ze-Nian Li, Co-Senior Supervisor

____
Dr. Greg Mori, SFU Examiner

Date Approved: March 31, 2009

Declaration of Partial Copyright Licence

Abstract

In this thesis we propose a standardized method for extracting illumination-invariant images and a novel approach for classifying textures. Experiments are also extended to include object classification using the proposed methods.

The illumination-invariant image is a useful intrinsic feature latent in color image data. Existing methods of extracting the invariant image are dependent upon the characteristics of cameras. Here, assuming that every image consists of data in a standardized sRGB color space, we develop a standardized method for extracting the illumination-invariant that is independent of camera characteristics.

Texture classification is an important aspect of Computer Vision. In this work, we greatly increase speed for texture classification while maintaining accuracy. Inspired by past work, we propose a new method for texture classification which is extremely fast due to the low dimensionality of our feature space.

Finally, we classify images of objects captured by varying the illumination angle.

To My Parents, My Sisters, and My Wife

"If God answers your prayers, He is increasing your faith; if He delays, He is increasing your patience; if He does not answer, He has something better for you."

— *Anonymous*

Acknowledgments

I would like to express the deepest gratitude towards my Senior Supervisors Professor Mark Drew and Professor Ze-Nian Li for enlightening me with their vast knowledge and experience in the field of Computer Vision and for their tremendous kindness in enduring through this thesis work.

Professor Drew, who is a genius when it comes to dealing with illumination in problems in Computer Vision, has opened so many new doors of knowledge for me and made me greatly appreciate the importance of appropriate processing of illumination in Vision problems through his patient and articulate nature in teaching and research. Professor Li's vast knowledge in the area of Computer Vision has helped me make my research work very well organized. It was a great pleasure going to him during various stages of this thesis work to verify my methodologies and to find out what other alternatives are there to consider for further improvements.

I would also like to convey heartfelt gratitude towards my thesis examiner Dr. Greg Mori for so many reasons. Not only he is the best teacher I have ever had, but is one of the friendliest and smartest people I have ever met. His wisdom in the field of Computer Vision and Machine Learning has made me want to explore these fields further. His patience to go through my thesis work and make valuable comments and corrections has improved this thesis work greatly. Also I would like to thank my thesis committee chair Professor Brian Funt, who is no less of a genius himself when it comes to dealing with illumination in Computer Vision, for sharing his thoughts on my thesis during the defence and help me improve my work and presentation.

It was my great fortune and an enormous joy to be able to work with so many great thinkers from the field of Computer Vision all at once. I am also particularly thankful to National Science and Engineering Research Council of Canada for the financial support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis we address two problems in computer vision namely Illumination-Invariant Image Formation and Texture Classification. We propose a standardized method for extracting illumination-invariant images and a novel approach for classifying textures. We begin with a discussion of the motivation behind addressing these problems.

1.1 Motivation for Illumination-Invariant Image Formation

The illumination-invariant image is a useful intrinsic feature latent in color image data. Since the inception of the term "Intrinsic Image" in [3], where any given camera input can be decomposed to its counter parts — reflectance image and illumination image as in Figure 1.1; one of the open challenges in computer vision has been to find a way to perform this decomposition with accuracy and intuitive sense. We want algorithms that aim to retrieve color values from sensor responses that only depend on the surface reflectance spectra [4], i.e. the effect of the illuminant is to be discounted. The reflectance image or the illumination-invariant image in particular would be of much use to any vision based machine learning algorithms, as it better represents the true object characteristics.

The illumination invariant image is formed from image data by taking the logarithm of band-ratio chromaticity colour coordinates, and then projecting in a certain direction [14]. The input colour data is 3-dimensional RGB, and the chromaticity is effectively 2D colour. Projecting in a 2-space direction generates a 1D, greyscale image. If the direction is chosen with care, the resulting greyscale image is quite independent of the lighting at each pixel, therefore forming an illumination invariant. The cleverness of the invariant is that it is

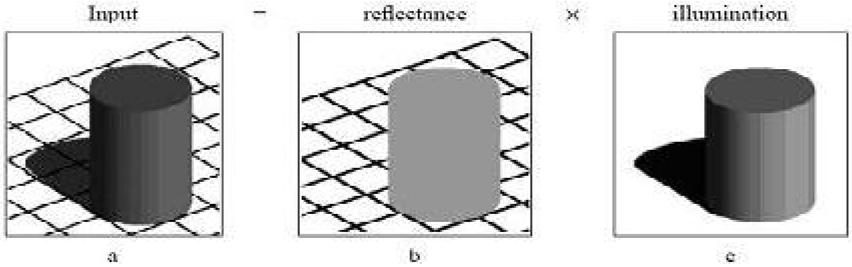

Figure 1.1: Intrinsic Image [3]

formed at each pixel independently, with no global image processing required.

The special direction for projection is that which is orthogonal to the direction that is followed along, in the 2-space, as the lighting changes (within a simplified model). Since lighting is thus removed, as a particular case shadows are also removed, or at least greatly attenuated [15, 13].

Since in fact we are projecting onto a line through the origin in a colour *2-space*, we need not think of the result of projection as merely a 1D, greyscale image: we do have as well, after all, a 2D coordinate position on that line, so we could state the projection answer as a 2D chromaticity [9]. Projection removes the lighting, but this can then be partially added back, by shifting the chromaticity projection line so as to make the chromaticity for bright pixels match that for bright pixels in the input image. So projection does not have to completely remove colour.

Once we indeed have an invariant image, we can go on to remove shadows by comparing edges in the original to edges in the invariant image. Removing or blending edges where these two edge maps disagree provides a strategy for re-integrating the edge map back into a full-colour, shadowless RGB image [15, 21].

1.2 Motivation for a Fast Texture Classification Method

Material surface texture classification is the process of determining the category of an unknown material from a set of known categories. This has a wide variety of use in fields such as multimedia information retrieval. For example, many image retrieval algorithms try to compute local variations of intensity to select the correct image containing a particular object(s) from a given database of images. Other important applications include defect detection in manufacturing processes, disease detection such as skin cancer, segmentation of satellite/aerial imagery, and document analysis. An important confounding problem is that in real life textured surfaces occur under variations of illumination and orientation, among other visual differences. These changes may make us perceive the same texture as different under different conditions. Until recently, most algorithms that tried to classify texture suffered from the effect of these variations of illumination and viewpoints.

Extracting meaningful properties from texture, and thus defining texture appropriately, is key to getting high accuracy rates in texture classification. The current state-of-the-art in texture classification [42] uses the definition that texture is composed of textons, the elementary building blocks of texture. We argue and show in this thesis that the process used for generating textons is subject to many experimental conditions and parameters and thus many textured surfaces (specially those under varying illumination and viewing angles) cannot be generalized by a universal set of textons. Moreover, the process is slow due to its complexity.

1.3 Contributions

For invariant-image extraction we would like to argue that it is possible to do a good enough job in finding the invariant image by simply assuming the input data to live in the standardized sRGB colour space and sharpening that space. In this way, we are not tied to finding the invariant projection for a particular camera, or using the data in a particular image, and can develop a standardized workflow that can be applied directly to any input image. Of course, deriving an invariant that is sensor- or image-adaptive instead, as originally conceived [15, 13], will likely work better than a one-size-fits-all approach, but here we show that results are indeed adequate using an approach applying the same transform to any image — e.g., shadows are principally attenuated, no matter what the input image.

For texture classification we try to learn the properties of texture from single images, making use of the insights as described in [24] to re-direct the foundation elements of the texton approach. The main argument vis-a-vis textons is that, given an image of a textured surface, we can apprehend various properties of it by convolving it with Gaussian derivative filters at various orientations and scale. The distributions of each of these filtered responses to Gaussian derivatives contain enough information which will help us classify the texture. We just have to extract this information efficiently and measure it meaningfully. The texton approach clusters vectors of filter responses and measures distance between histograms of these cluster centers. In this thesis we propose a novel approach for classifying texture, whereby we represent texture in a Weibull space. We then learn the information stored in each training image of a particular texture class by measuring its information entropy in this space. In the classification stage we choose for a test image its nearest neighbour in the Weibull space, i.e. the training image which has the closest amount of information as in the test image. The result is a much faster algorithm, somewhat similar to the texton approach but without the same level of complexity. We perform all our experiments on the CURET database [7], one of the most challenging databases available for texture images capturing variations in illumination and viewpoint.

1.4 Thesis Outline

In Chapter 2 we briefly review the literature in illumination-invariant image extraction and also in texture classification.

In Chapter 3 we propose our standardized method of illumination-invariant image extraction. In Section 3.1 we compare the strategy of sharpening XYZ data, for input nonlinear-sRGB images, to the new approach of sharpening the sRGB data directly, and show that a better invariant (more invariant to lighting change) arises from the latter approach. We extract the illumination invariant from measured, nonlinear input data for images of the Macbeth chart across 105 different illumination environments. Applying the standardized illumination invariant extraction scheme presented here produces images much more independent of lighting change. And in Section 3.3 we apply the new method to the problem of reducing or removing shadows from imagery, by generating an invariant image from input colour images, making use of the new standardized method.

In Chapter 4 we describe our approach to texture classification, with experiments in Section 4.6 and analysis and comparison of results with the state-of-the-art in Section 4.7.

Chapter 5 describes our experiments with object recognition using our method for extracting illumination-invariant object images without prior knowledge of camera characteristics, and then using our fast method for texture classification to classify object images in a low dimensional feature space.

We finish with some concluding remarks in Chapter 6.

Chapter 2

Previous Work

2.1 Background on Illumination-Invariant Image Formation

Many approaches have been proposed to find the invariant image from an input color image. While the idea for finding the invariant is fairly simple (as described in Section 1.1), carrying out finding the proper direction in which to project in a 2D log-chromaticity colour space is not necessarily as straightforward. In [15], the camera itself was calibrated, in this invariant image sense, by utilizing a set of images of a colour target, under different illuminants, to find the best 2D direction characterizing lighting change. In [13], evidence in the image itself was used to discover the correct direction orthogonal to the lighting change direction. There, it was argued that such a projection direction is best described as that leading to a minimum-entropy distribution in greyscale values.

Previously, in an approach inspiring ours sharpening in XYZ colour space was tried recognizing that the sRGB standard [5] contains not only a mapping from nonlinear to linearized colour values, but also a relationship from the sRGB gamut to corresponding XYZ values via matrixing. Thus it was proposed [18] that input images could be assumed to be in nonlinear sRGB colour space, linearized to linear-sRGB, and then transformed to XYZ. Then, in XYZ, the XYZ curves themselves could be sharpened. The results for shadow removal were indeed better than simply moving removing gamma-correction. However, the invariant direction still was found using the entropy method of [13], so a fully standardized data independent method was not developed.

We would argue, moreover, that going from linear-sRGB to XYZ is in itself counterintuitive, in that the sRGB colour-matching functions are close to sharpened colour-matching

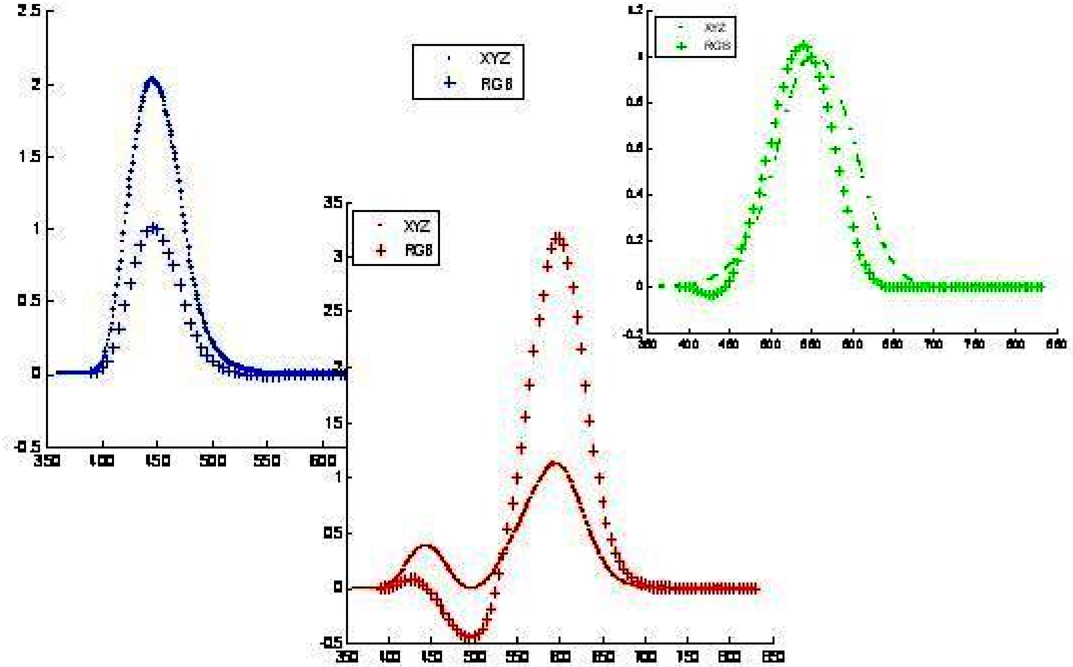

Figure 2.1: sRGB (+ curves) to XYZ (dotted curves) is a Broadening Transform

curves already [19], so that going over to XYZ curves is a kind of broadening transform, see Figure 2.1. Following this with a further sharpening of the XYZ curves is not as direct as simply sharpening the sRGBs themselves, and hence that approach is what we propose here. And, we show that the new idea, of sharpening the sRGB data, provides better performance for producing an illumination invariant.

Moreover, once we decide to assume that all input data consists of sRGB values, and we provide a sharpening transform for sRGB, we can in fact find the best projection direction simply using synthetic data and then apply the same transform and projection once and for all to any input image. We show that this simple strategy produces reasonable results, within the application of shadow removal.

2.2 Background on Texture Classification

Over the past 30 years textures analysis have been widely studied and numerous methods have been proposed for describing and classifying image texture. Texture classification methods were divided by [41] into four categories: statistical, structural, model-based, and signal processing. A brief description of these approaches is provided in this section. For detailed study we point the reader to the following surveys on texture analysis methods: [37, 36, 41, 28].

Statistical methods aim to extract pixel-level features by deriving a set of statistics from the local gray values. Popular methods in this category include autocorrelation functions [26], transforms, edgeness, random field models and gray level differences [43], which have inspired a variety of modifications later on.

The building blocks of texture are sought for by Geometrical methods. Edges are considered to be the primitive elements of texture by a large number of authors. Geometrical methods based on edges look to detect them through Laplacian-of-Gaussian or difference-of-Gaussian filter [32], or by mathematical morphology [34].

Model based methods are based on placement rules of texture elements which may be detrministic or random [28]. Texture elements may be defined in terms of gray level, shape, or homogeneity oin size and orientation. Deterministic placement rules include adjacency, closest distance and random placement rules measure properties such as edge density and run lengths of maximally connected texture elements.

Signal processing methods introduced the notion of spatial filters to the texture analysis process. These filters measure frequency information and common techniques include the use of masks that are designed for edge detection (e.g. Roberts' and Sobel's operators [38]).

Most of the earlier work assumed constant imaging conditions and therefore are limited in terms of performance when such variations are added to the image. An excellent example of a database of textures that incorporate variation in lighting and viewpoint is CURET [7]. This is one of the most challenging and largest databases for texture. Figure 2.2 depicts some of the challenges presented by the CURET dataset.

Originally proposed by [Julesz, 1981 [29]], [Leung and Malik, 2001 [30]] provided the first working version of textons, “elementary particles” that constitute texture. The work of [30] (denoted LM) produced notable classification results on the CURET dataset. In the texton approach, filter responses are first generated by convolving each training image with a bank of filters (48 are used in LM) that include first and second derivatives of Gaussians at multiple scales and orientations, Laplacian of Gaussians, and Gaussians. A 2D texton is defined as the cluster centers in the filter response space, where each (sampled) pixel has a 48-vector of responses. However, images had to be carefully registered during the learning

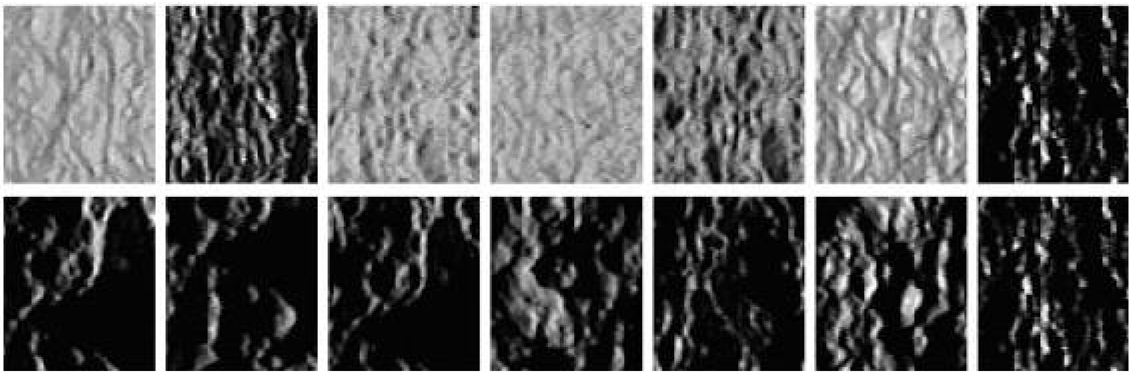

Figure 2.2: Challenges in the CURET Dataset. Top Row: Texture #30 under Constant Viewing Angle and Varying Illumination. Bottom Row: Texture #30 under Constant Illumination and Varying Viewing Angle.

stage and then mapped to a 48 dimensional filter response space, effectively generating 3D textons.

The same set of filters as LM were used by [6] (Cula and Dana, denoted "CD"), but without the registration process and generating textons from single images, instead of image stacks as proposed in LM.

Rotationally invariant set of Gabor-like filters were proposed by [Schmid, 2001 [40]] (denoted "S") that also achieved good classification performance on the CURET dataset.

To date, the state of the art in terms of classification accuracy is provided by the work of [42]. They introduce the idea of Maximum Response Filters (MR8 and MR4) which are a collapsed sub-set of the "Root Filter Set" (RFS). The RFS filters containing both isotropic and anisotropic filters are similar to the LM filters, but there are 38 of these instead of 48 (as in LM). The filter responses from the 38 filters are collapsed by keeping the maximum response across orientations, thus reducing the number of filter responses to 8 and 4 for MR8 and MR4 respectively, for each image. This is done to extract the strongest response across filters thus generating meaningful features even from textured images that are at acute angles. The traditional rotationally invariant features, such as the S set, fail to extract features from anisotropic textures [42]. Moreover, in the Maximum Response set the dimensionality of the feature space is reduced which makes the clustering process simpler. Finally, they propose a greedy algorithm which tries to reduce the number of models required to represent a class of texture without affecting classification accuracy.

A different line of research, such as in [24], is concerned with other properties of textured surfaces. In [23], they provide the notion of a sequential fragmentation process. Here a textured surface is perceived to be the result of an object that has been spatially fragmented. The fragmentation process is stochastic in nature for almost all textures, especially those present in the CURET database. They propose the Weibull distribution as suitable to measure the distribution of such textures as a function of orientation. As a result, the (two) Weibull parameters that characterize a probability distribution function are capable of characterizing the spatial layout of stochastically ergodic textures. In [24] they move on to extract properties of texture (such as regularity, coarseness) based on the Weibull parameters. We take insight from their work and, using Weibull parameters, define our own feature space which has a much reduced dimensionality than other texture classification methods discussed in this section.

We compare our method's classification accuracy and dimensionality with all four methods S, LM, MR8, and MR4 in Section 4.7.

Chapter 3

Illumination-Invariant Image Extraction

In this chapter we derive a standardized method for extracting Illumination-Invariant images that is independent of camera characteristics and also do not depend on input image data. The idea in forming an illumination invariant is to post-process input image data by forming a logarithm of a set of chromaticity coordinates, and then project the resulting 2-dimensional data in a direction orthogonal to a special direction, characteristic of each camera, that best describes the effect of lighting change. Lighting change is approximately simply a straight line in the log-chromaticity domain; thus, forming a grayscale projection orthogonal to this line generates an image which is approximately independent of the illuminant, at every pixel. But a problem, addressed here, is that the direction in which to project is camera-dependent and we may not have information on the camera. So here we take a simpler approach and assume that every input image consists of data in the standardized sRGB color space. Previously, this assumption has led to the suggestion that the built-in mapping of sRGB to XYZ tri-stimulus values could be used by going on to sharpen the resulting XYZ and then seeking for an invariant. Instead, here we sharpen the sRGB directly and show that performance is substantially improved this way. This approach leads to a standardized sharpening matrix for any input image and a fixed projection angle as well. Results are shown to be satisfactory, without any knowledge of camera characteristics.

3.1 Sharpening XYZ versus Sharpening sRGB

3.1.1 Sharpening XYZ

The first approach to a standardized sharpening and projection scheme was to sharpen XYZ values arising from input nonlinear-sRGB images [18]. Here we propose re-examining this approach as going from RGB to XYZ is seemingly a broadening transform, so we sharpen colour-patch data directly rather than sharpen the XYZ colour-matching curves — that is, we take a maximum-prescience approach rather than a maximum-ignorance one. But what colour data should we utilize? As a set of fairly generic inputs, suppose we simply use the 24 patches of a Macbeth ColorChecker [35], with synthetic values for tristimulus values under Planckian lights [44]. However, here we are aiming at the idea of starting with sRGB data; therefore we first transform the resulting XYZ values back to linear-sRGB colour space, and thence back to XYZ again. The thought here is that the transform from XYZ to sRGB [5] may involve clipping to the range [0,1], and we wish to take that into account. Therefore we generate a set of synthetic images of the Macbeth chart, formed under 9 Planckian lights for temperatures T=2,500°–10,500° in 1,000° intervals. We define the synthetic data in XYZ coordinates rather than in sRGB so that we have meaning and generality for the data. Taking the resulting XYZ triples to linear-sRGB colour space does turn out to involve some clipping. Then we take the data back to XYZ space.

Finally, we wish to consider an invariant in a colour 2-space, and here we make use of log-chromaticities formed as the logarithm of ratios of the XYZ to their geometric mean [11]:

$$\log x_k \;=\; \log \left[\{X, Y, Z\}/(X \cdot Y \cdot Z)^{1/3}\right] \tag{3.1}$$

This generates 3-vector quantities but, in fact, in the log space every such 3-vector lies in the plane orthogonal to the unit vector $\boldsymbol{u} = (1/\sqrt{3})(1,1,1)^T$; thus only two coordinates are independent. We can rotate into that plane (cf. [9]) by forming a 2-vector $\boldsymbol{\chi}$ by making use of the 2×3 rotation matrix $\boldsymbol{U}^T$ equal to the orthogonal matrix factorizing the projector onto the subspace perpendicular to $\boldsymbol{u}$:

$$\begin{aligned} P^{\perp} &= I - \boldsymbol{u}\,\boldsymbol{u}^T, \\ P^{\perp} &= \boldsymbol{U}\,\boldsymbol{U}^T, \quad \boldsymbol{U} \text{ is } 3 \times 2 \\ \boldsymbol{\chi} &= \boldsymbol{U}^T \log \boldsymbol{x} \end{aligned} \tag{3.2}$$

A plot of the resulting 2D colour coordinates in Figure 3.1(a) shows that, rather than forming straight lines as expected, we see some curvature in the plots as lighting changes. If we center the data by subtracting the mean $\boldsymbol{\chi}$ vector for each colour patch, we would like to see as close as possible to a single straight line through the origin, for the purposes of forming a lighting invariant: a single straight line would indicate that, in Figure 3.1(a) we could simply project in a direction orthogonal to the direction of that line and effectively eliminate the influence of lighting on the feature. I.e., we could generate a 1D greyscale illumination invariant.

However, in Figure 3.1(b), for mean-subtracted data, we instead see that the data is fairly spread out. We can discount the effect of outliers to a degree by finding the best slope using a robust statistical method [39], but still, we find the data has a correlation coefficient R of only 0.605 — not an excellent indicator of straight-line behaviour.

Therefore we consider *sharpening* the colour-patch data [12], in order to make the illumination-invariant image formation model [14] more applicable, since the theory behind the model requires quite sharp camera sensors. We thus make use of the data-based sharpening method [12] to determine a sharpening matrix T; we choose the synthesized data under the most red and the most blue lights, and find the best least-squares matrix transforming one into the other. The sharpening matrix T is the set of eigenvectors of the least-squares transform. Figure 3.1(c) shows that sharpening does indeed straighten out the log-chromaticity plots; for mean-subtracted data in Figure 3.1(d), we now find a correlation coefficient R=0.764, a much improved value.

3.1.2 Sharpening sRGB

The objective of this research is to determine whether sharpening sRGB values themselves can produce a better illumination invariant than can sharpening XYZ values. Therefore now we compare how sRGB log-chromaticities fare under lighting change — can we sharpen analogues to eq. (3.1) constructed from linear-sRGB values and arrive at a better invariant?

Firstly, we examine how sRGB itself does in forming an invariant. We plot linear-sRGB for the synthetic images under 9 different lights, with results shown in Figure 3.2(a). We see that sRGB coordinates do indeed form straighter lines than do XYZ coordinates (in Figure 3.1(a)). For mean-subtracted values, in Figure 3.2(b), we find a correlation coefficient R=0.837, already better than sharpened XYZ notwithstanding outliers created by clipped values. (We use a generalized logarithm [18], not a logarithm, to avoid the log of zero.)

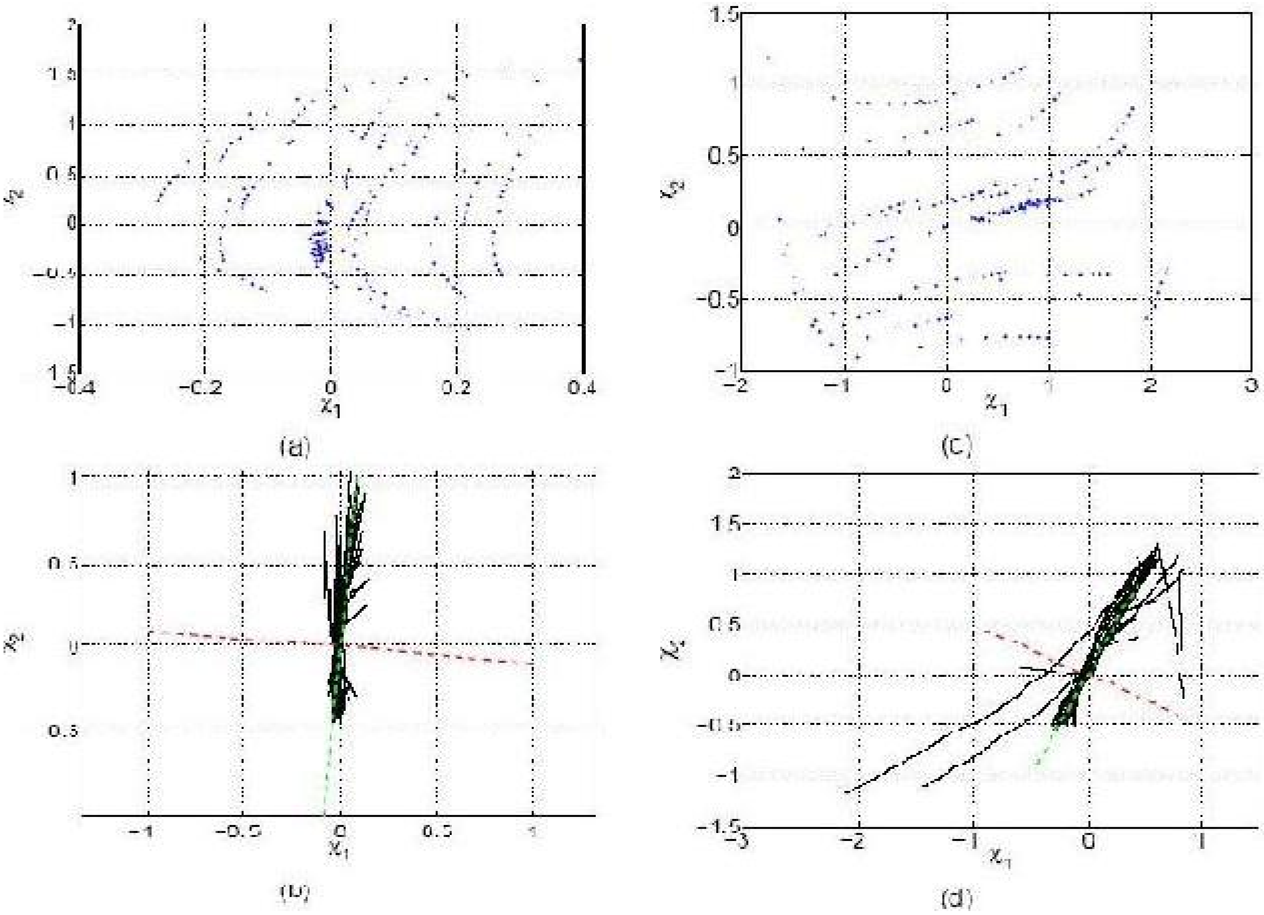

Figure 3.1: Log-chromaticity XYZ coordinates for Macbeth patches, as light changes. (a): $\boldsymbol{\chi}$ vectors; (b): Mean-subtracted values: best (robust) direction in green, orthogonal direction in red, R=0.605. Lines joining data points are for each colour patch, as lighting changes. (c): Sharpened XYZ; (d): Sharpened, mean-subtracted: R=0.764.

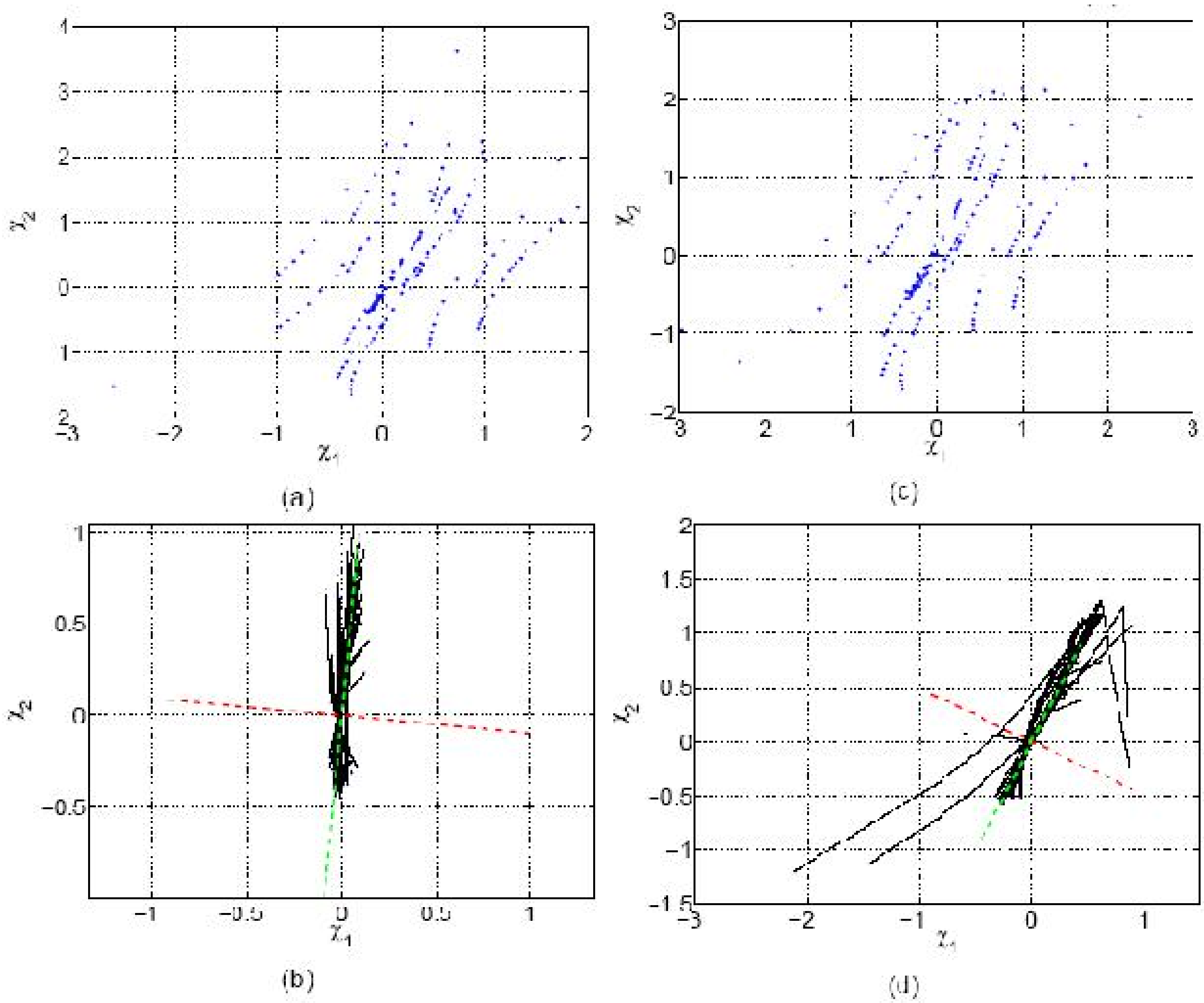

Figure 3.2: Log-chromaticity sRGB coordinates for Macbeth patches, as light changes. (a): $\boldsymbol{\chi}$ vectors; (b): Mean-subtracted values: R=0.837. (c): Sharpened sRGB; (d): Sharpened, mean-subtracted: R=0.877.

Data-based sharpening in this case actually makes the correlation coefficient worse: Table 3.1 shows that applying sharpening results in R=0.630 (the mean-subtracted data is somewhat spread out).

However, if we make use of a white-point preserving data-based sharpening [10], then R is improved: R=0.877, the highest value found so far.

3.1.3 Optimized sRGB Sharpening Transform

The result above is encouraging, since it indicates that sharpening sRGB does indeed produce the best illumination invariant, the result we argue for in this paper. However, while the result is good, it could be better as shown in this section.

Sensor sharpening simply has the objective of concentrating energy in each sensor in its associated colour band. However, here we have a specific objective: producing the best invariant coordinate. Therefore we adopt the optimization strategy in [8], which aims specifically at finding the best sensor transform $\boldsymbol{T}$ the minimizes the spread of the lines plotted in a mean-subtracted log-chromaticity space. The optimization also insists on non-negative results, after applying the colour transform $\boldsymbol{T}$.

Applied to the sRGB data, we find the following transform

$$\boldsymbol{T} = \begin{pmatrix} 0.9968 & 0.0228 & 0.0015 \\ -0.0071 & 0.9933 & 0.0146 \\ 0.0103 & -0.0161 & 0.9839 \end{pmatrix} \tag{3.3}$$

The sRGB data forms quite straight lines, now, in the transformed space, as shown in Figure 3.3(a). For the mean-subtracted data in Figure 3.3(b), we now find the improved correlation coefficient value: 0.920. Thus we suggest adopting the linear-sRGB colour space transform matrix (3.3) as a *standard* colour transform. The direction for orthogonal projection found by a robust regression, shown in red in Figure 3.3(b), is given by the 2D vector $\boldsymbol{e}^{\perp}$ orthogonal to the lighting-change direction:

$$\boldsymbol{e}^{\perp} = (0.9326, -0.3609)^T \tag{3.4}$$

Thus, overall, we argue here that as a standardized workflow for producing an illumination invariant image from an input colour image we proceed as follows:

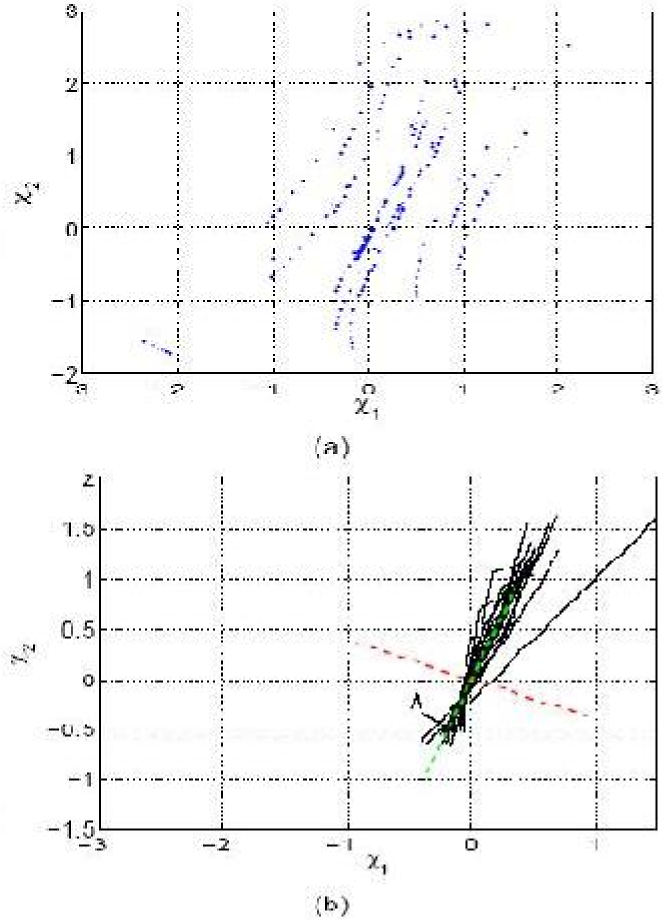

Figure 3.3: Optimized log-chromaticity sRGB coordinates. (a): $\boldsymbol{\chi}$ vectors; (b): Mean-subtracted values: R=0.920.

Table 3.1: Correlation coefficient values R for projection of mean-subtracted log-chromaticity, formed according to the method in columns "Scheme": from XYZ coordinates, and from sharpened XYZ; from sRGB, from sharpened sRGB, and from white-point preserving sharpened sRGB; and finally using an optimized tranform T from eq.(3.3) on sRGB coordinates.

Scheme	R
XYZ	0.605
XYZ$^{\#}$	0.764
sRGB	0.837
sRGB$^{\#}$	0.630
sRGB$^{\#}_{WPP}$	0.877
sRGB$_{T\_OPT}$	0.920

```
Transform input image nonlinear sRGB to linear-sRGB.
Transform to sharpened colour space. I.e., if linear sRGB
    values are $\boldsymbol{\rho}$ , then $\boldsymbol{\rho}^{\#} = \boldsymbol{T}\,\boldsymbol{\rho}$ , where $\boldsymbol{T}$ is given by (3.3).
Form 2D log-chromaticity coordinates $\boldsymbol{\chi}$ as in eq. (3.2) for
    sRGB values.
        I.e., form 2-vectors $\boldsymbol{r}$ via
        $\boldsymbol{r} = \log\boldsymbol{\rho} - (1/3)\sum_{k=1}^{3}\log\rho_k$,
        $\boldsymbol{\chi} = U^T\boldsymbol{r}$ , using sharpened $\boldsymbol{\rho}^{\#}$ values.
        E.g., use $\boldsymbol{U} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{6} \\ 1/\sqrt{2} & 1/\sqrt{6} \\ 0 & -2/\sqrt{6} \end{pmatrix}$
Project onto line perpendicular to lighting-change direction,
    using vector $\boldsymbol{e}^{\perp}$ in (3.4).
Form 2D-colour from projected point by rotating back
    to a 3-vector using $3 \times 2$ matrix $\boldsymbol{U}$ .
Exponentiate to go back to non-log coordinates.
Move to chromaticity in an $L_1$ norm by dividing
    by $(R + G + B)$.
```

The above algorithm generates 3D colour, but only from values projected onto the

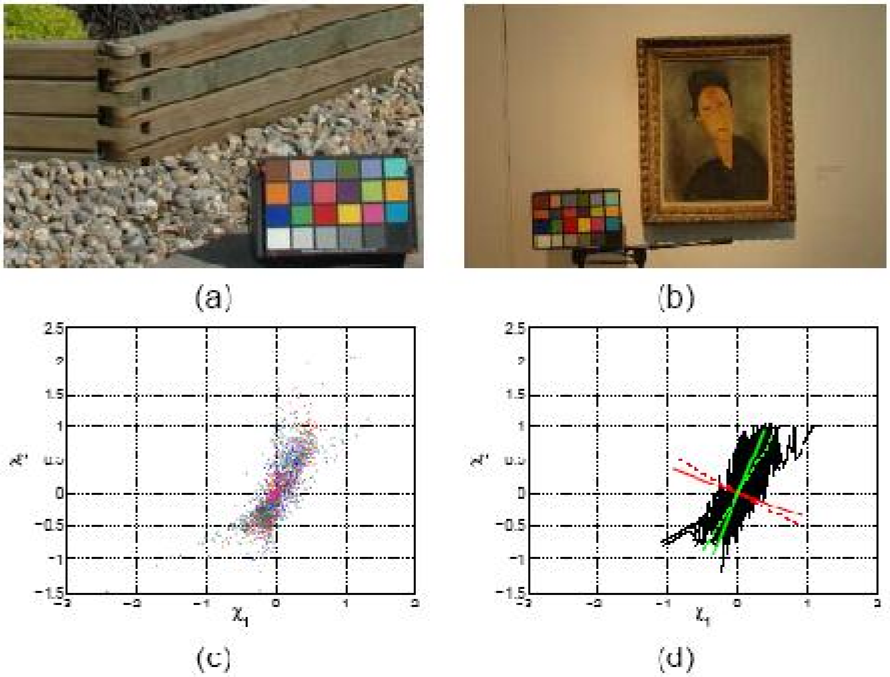

Figure 3.4: Log-chromaticity sRGB coordinates for measured empirical Macbeth patches, for 105 different lighting conditions. (a,b): Samples of images. (c): Mean-subtracted values: R=0.775. (d): Sharpened, mean-subtracted: R=0.809.

projection line, so effectively 2D. Nonetheless, the 2D colour information can still be useful [9].

In the next section, we apply this algorithm to empirical images of the Macbeth chart, situated in various illumination environments, and show the efficacy of such a generic sharpening plus projection scheme.

3.2 Invariant from Measured Chart Data

A set of various images that include a Macbeth chart in the scene were acquired under 105 different lighting conditions. [1] Figure 3.4(a,b) shows two of these images, which we treat as nonlinear-sRGB. Forming the mean-subtracted log-chromaticities, we find that without any colour space transform the correlation coefficient is only R=0.775. Thuswe would not

[1] These images are due to Prof. Graham Finlayson and Dr. Clément Fredembach. The images are nonlinear; the camera used was a Nikon D70.

expect to achieve a reliable illumination invariant without transforming the colour space.

Now if we apply the algorithm given above, applying transform $\boldsymbol{T}$, we then achieve an R value of R=0.809. I.e., while an optimization applied to this data would do better, the pre-defined transform derived from synthetic data already does quite well. In Figure 3.4(d), we show the pre-determined projection line as a solid line, and the best-fit one for the actual data in a dashed line — the two are not far apart.

In the next section, we apply the standardized algorithm to ordinary images, with a view to testing the efficacy with respect to shadow removal in an invariant image.

3.3 Experiments and Critical Analysis

Here we apply the standardized algorithm to a set of images acquired under a variety of illumination environments. We form the 2D chromaticity, both without and then with invariant image processing applied as described above. If the standardized approach to extraction of an illumination invariant does indeed work, we expect shadows to be attenuated, compared to in the original chromaticity image.

Figure 3.5(a,d,g,j,m) shows several images that contain shadows. The effect in every case, over the cameras utilized, is to remove or at least reduce the effect of shadows. This is shown by displaying the chromiticity images with their edge-map overlaid: edges for shadows appear in the original Figure 3.5(b,e,h,k,n), but not in the invariant version of the chromaticity Figure 3.5(c,f,i,l,o). This can then be used to go on to remove shadows from images (see [17] and [21] for approaches to this task). Just as long as shadow-edges are indeed eliminated, we can go on to remove shadows in the original RGB images.

Figure 3.6 further shows that the standardized method does in fact produce a usable invariant. Here the 24 Macbeth patches were imaged under 14 different daylights using a HP912 camera (Figure 3.6(a)). In Figure 3.6(b) we have the illumination-invariant formed by calibrating the camera. For this image the average standard deviation across illuminants for the macbeth patches is 4.42%. Figure 3.6(c) demonstrates the illumination-invariant image formed by applying the proposed stadardized method. In this image the average standard deviation across illuminants for the macbeth patches is 6.11%. Certainly comparing to the best possible invariant (that of Figure 3.6(b) by calibrating the camera) this is not better, but usable nevertheless.

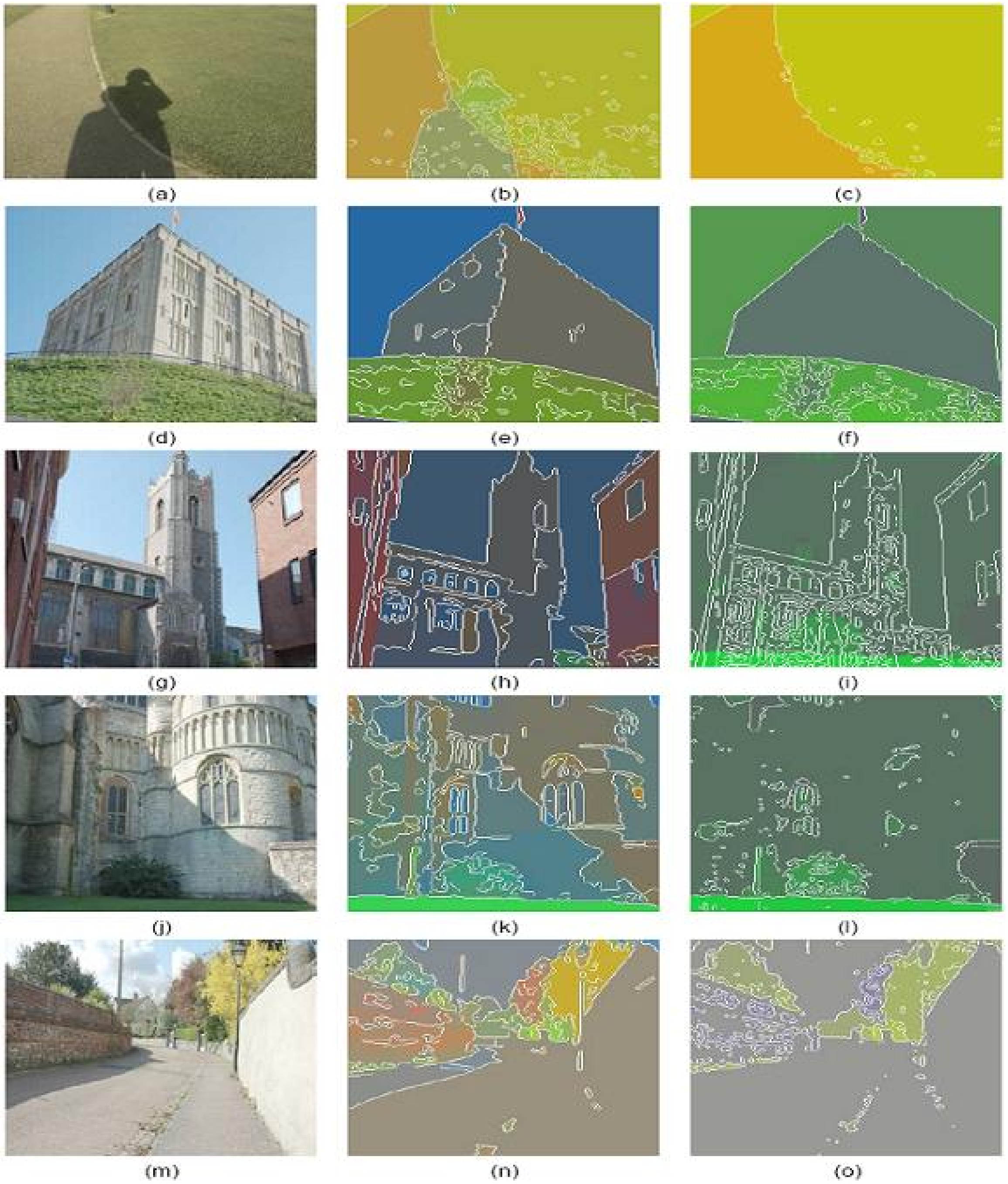

Figure 3.5: Input colour images (a,d,g,j,m), their chromaticity (b,e,h,k,n), and the chromaticity images for an extracted illumination invariant (c,f,i,l,o). Here, the Mean-Shift algorithm has been applied to generate a cleaner image, and edge-detection overlaid — the illumination invariant has fewer edges on shadow boundaries. Cameras used were an HP 912 and a Nikon D70.

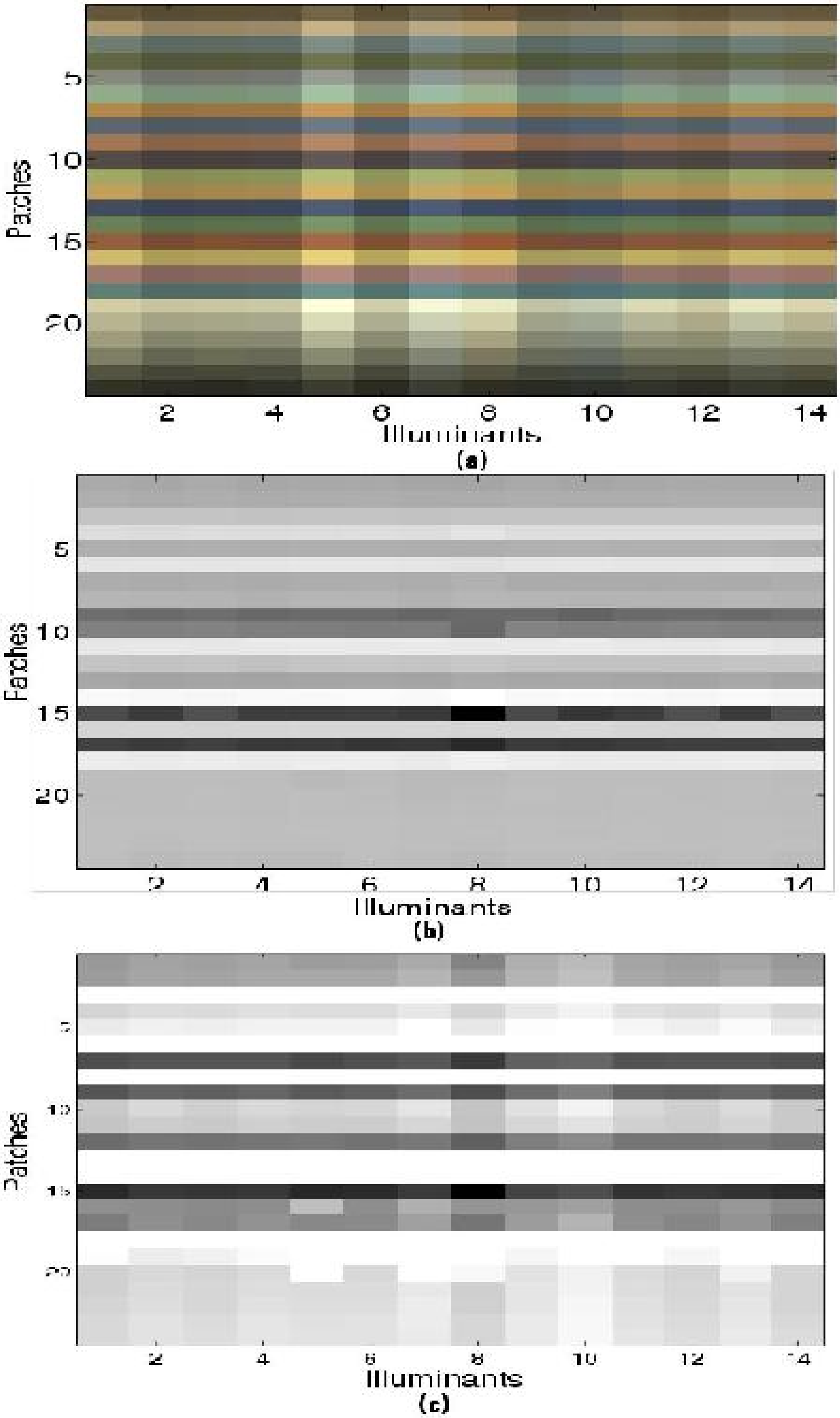

Figure 3.6: (a) Macbeth chart under 14 different daylights. (b) Invariant image formed by calibrating camera. (Av. Std. Dev. across illuminants = 4.42%). (c) Invariant image formed by applying the proposed standardized method. (Av. Std. Dev. across illuminants = 6.11%)

3.4 Summary

In summary to our contributions, we have proposed an algorithm, independent of camera characteristics and image data, that extracts the invariant image from an input sRGB image. Through various experiments we have shown that our method produces a usable invariant and reduces the effect of illumination greatly. Such an algorithm can be utilized as a preprocessing step to many vision algorithms, performance of which suffer due to inconsistent illumination effects in input image data.

Chapter 4

A Fast Method for Classifying Surface Textures

In this chapter we describe a novel approach for classifying texture under varying conditions of illumination and viewpoint, whereby we represent texture in a Weibull space. We then measure the information stored in each training image of a particular texture class by measuring its information entropy in this space. In the classification stage we classify a test image by choosing its nearest neighbour in the Weibull space, i.e. the training image which has the closest amount of information as in the test image. The result is a much faster algorithm which we compare with the state-of-the-art in terms of speed and accuracy. We perform all our experiments on the CURET dataset [7].

4.1 Preprocessing Steps

Identical to [42] the following pre-processing steps are applied before going ahead with any learning or classification.

We use the modified version of the CURET dataset which can be found in [25]. All processing is done on the cropped regions in this dataset (see Figure 4.1) and they are converted to grey scale and intensity-normalized to have zero mean and unit standard deviation. This normalization gives invariance to global affine transformations in the illumination intensity. Second, filter banks are L_1 normalized, so that the responses of each filter lie roughly in the same range. In more detail, each filter F_i in the filter bank is divided by $|F_i|_1$ so that the

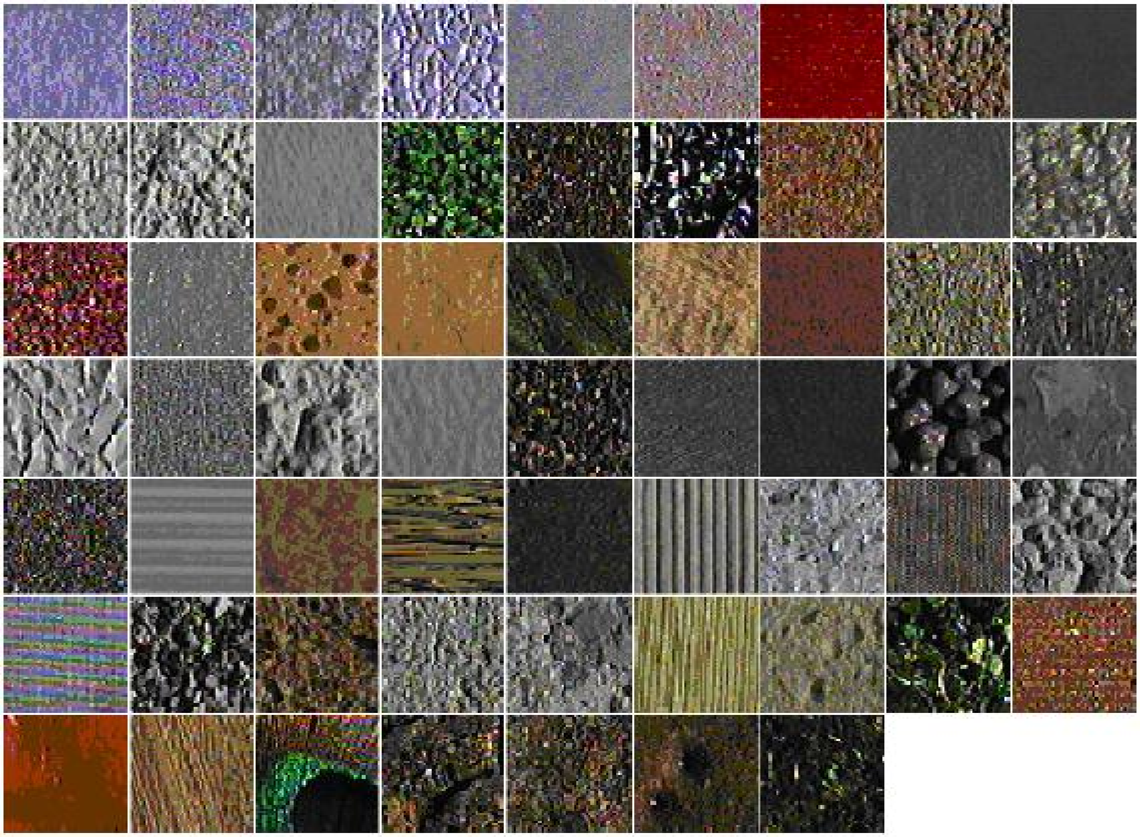

Figure 4.1: Textures in the CURET Dataset

filter has unit L_1 norm. This is to make the scaling for each of the filter response axes the same [31].

To meaningfully compare filter responses of different images (through Histogramming see Section 4.3 and Mapping to Weibull Space see Section 4.4) they are contrast normalized so that they lie in the same range. Let $|F(x)|$ be the L_2 norm of the filter responses at pixel x. We normalize the filter responses by equation 4.1:

$$F(x) \leftarrow F(x) \times \frac{\log\left(1 + \frac{|F(x)|}{0.03}\right)}{|F(x)|} \tag{4.1}$$

Finally, although proposed in [20, 31, 42], we do not contrast-normalize the image since from our experiments it seems to enhance noise more than signal, thus affecting classification performance. This has also been noted by [33].

4.2 Root Filter Set (RFS)

RFS consists of 38 filters [42], partitioned as follows: first and second derivatives of Gaussians at 6 orientations and 3 scales making a total of 36, and 1 Gaussian and 1 Laplacian of Gaussian filter. The Gaussian and Laplacian of Gaussian both have scale $\sigma = 10$ pixels (these filters have rotational symmetry). The bar (first derivative) and edge (second derivative) filters both include 3 scales: $(\sigma_x,\sigma_y) = \{(1,3),(2,6),(4,12)\}$. These filters are oriented at 6 orientations: (0^o, 30^o, 60^o, 90^o, 120^o, 150^o). Sample filters and their corresponding filter responses on a textured surface are displayed in Figure 4.2 (* denotes convolution).

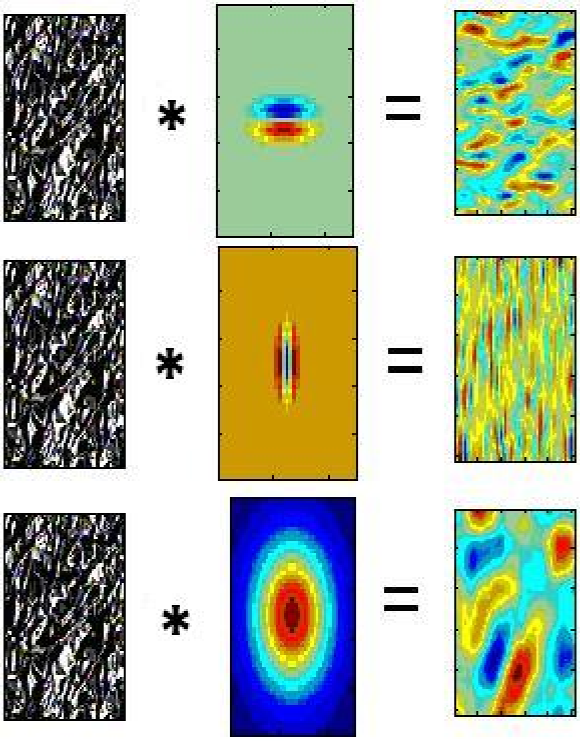

Figure 4.2: Sample Filter Responses. An Example Image (1st column), * denotes Convolution, Medium Scale Gaussian First Derivative at 0^o (2nd column, 1st row), Medium Scale Gaussian Second Drivitive at 90^o (2nd column, 2nd row), Rotationally Symmetric Gaussian at Scale = 10 (2nd column, 3rd row), Corresponding Filter Responses (3rd column)

4.3 Histogramming

After applying RFS to each training image, we obtain a set of 38 filter responses for each image. We histogram each of these filter responses individually to speed up the process of Weibull parameter estimation for a particular filter response (cf.[24]). Each histogram consists of 1001 equally spaced bins according to the range of data within each filter response. However, this property (number of bins/bin size) will be analyzed in Section 4.7 and some interesting insights will be revealed. Figure 4.3 shows a generated histogram (in red) with Weibull Probability Density Function (see section 4.4) fitted to it (in black)

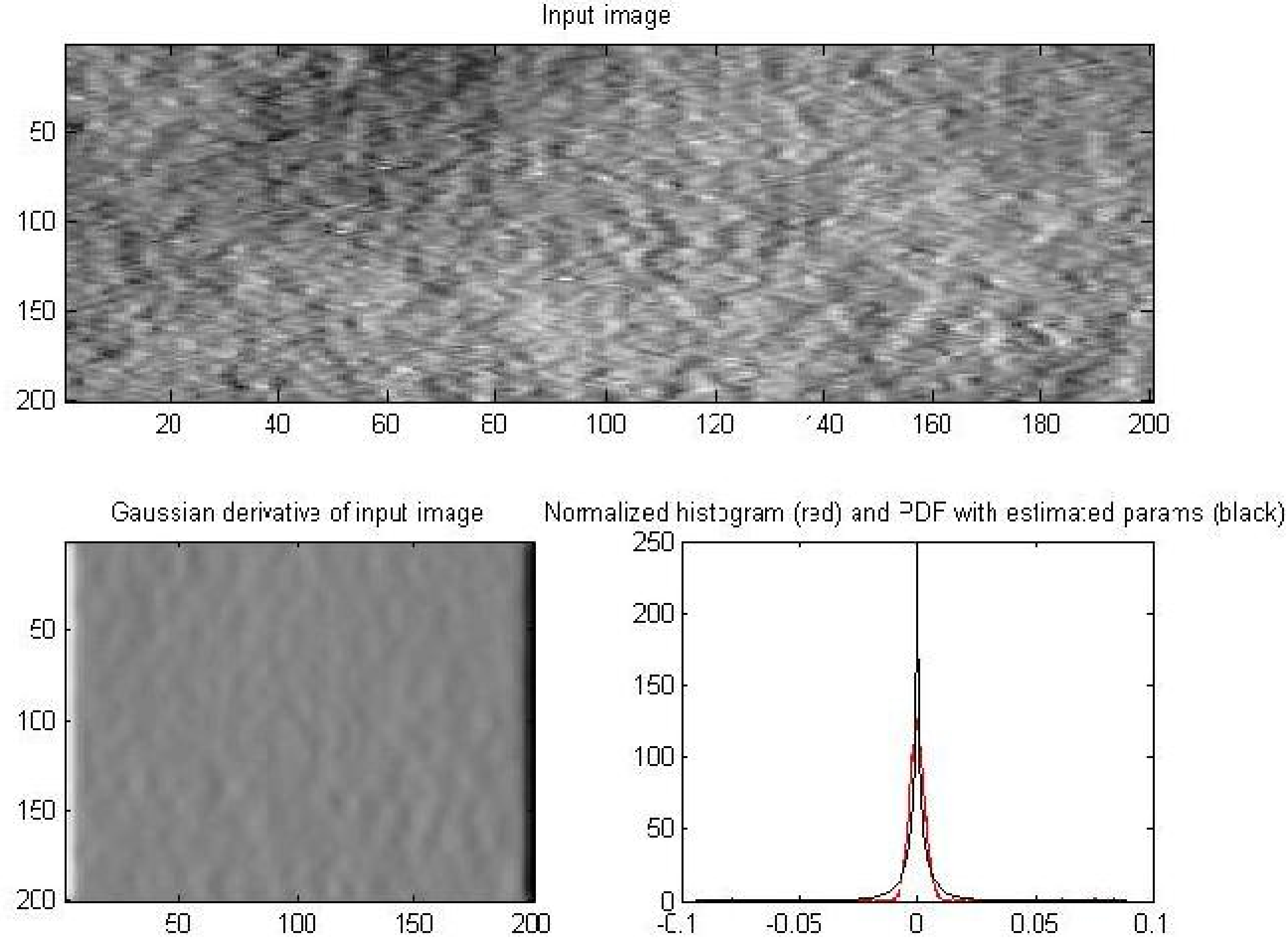

Figure 4.3: A Sample Input Image (top row), The Gaussian Derivative of that Image (bottom row left), and Generated Histogram (in red) with Weibull Probability Density Function fitted (in black) (bottom row right)

4.4 Mapping to the Weibull Space

At this point we observe the nature of textured surfaces proposed in [23], and therefore move to fit a 2-parameter Weibull distribution to each of the histograms we generated in the previous step. As suggested in [24] Weibull parameters completely characterize stochastically ergodic textures. And [24] imply that almost all textures in the CURET dataset are of such nature. Therefore, we map a filter response to its corresponding location in the Weibull space. The Weibull distribution has the probability density function as in equation 4.2:

$$f(x,k,\lambda) = \frac{k}{\lambda}(x/\lambda)^{k-1}e^{-(x/\lambda)^k} \tag{4.2}$$

for $x > 0$ and $f(x;k,\lambda) = 0$ for $x \leq 0$ where $k > 0$ is the shape parameter and $\lambda > 0$ is the scale parameter of the distribution.

Many approaches have been proposed to estimate the parameters of such 2-parameter Weibull distribution. We opt for the maximum likelihood estimation technique where a likelihood function is defined and values for shape and scale are obtained by trying to maximize this function. We utilize the standard method where the partial derivatives of the likelihood function are taken with respect to the parameters. Setting these two equations equal to zero we can now solve for the parameters simultaneously. The two equations for shape and scale, respectively, are as follows (equations 4.3 and 4.4):

$$\frac{\sum \log(x)x^k}{\sum x^k} - \frac{1}{n}\sum \log(x) - \frac{1}{k} = 0 \tag{4.3}$$

$$\lambda = \left(\frac{\sum x^k}{n}\right)^{1/k} \tag{4.4}$$

As in [24], equation 4.3 (estimating shape of the distribution) is solved using the Newton-Raphson method. The precision of the Newton-Raphson method is set to 0.01, and shape is initialized at 0.01. We allow a maximum of 30 iterations for the solution to converge. In practice, almost 100% of the time convergence occurs within 5 iterations (see Section 4.7).

4.5 The Final Model

We store the shape and scale parameters for each filter response for each image. This is our model for an image; i.e. every image is represented by a 76-vector (38 values for scale and

38 values for shape). And that is the complete, very simple model used here. Each image in the database has 2 Weibull parameters associated with each filter response. Algorithms such as k-Means, which compute joint statistics of filter responses, are expensive and slow down the model generation phase and as will be shown in Section 4.7 are dependent upon numerous experimental parameters.

In comparison, for the texton approach there are several variants however in [42], e.g. for the MR8 approach, a subset (13) of training images is selected with, for example 10^4 sample pixels, from each image. Each pixel has 8 filter response values (after having reduced RFS to MR8), so textons for a class are the k-Means clusters resulting from clustering 130,000 8-vectors, a daunting task. If we use global texton set and just 10 textons per class, then each pixel's 8-vector is associated to the closest texton. In the case where there are 20 possible texture categories and 10 textons per class, each texture is finally represented as a histogram over the total 200×8 texton set. So in this case a training image is represented as a 200-vector. Test images have their pixels each associated to a closest texton, and the resulting histogram is again over 200 bins. Classification results are obtained by comparing the histogram for the test image with all those for training images in all classes, categorizing according to the closest histogram.

In the case when there are 40 possible texture classes, the histogram would have 400 bins, and 610 bins for the 61 class case (again using 10 textons per class). Altogether, the texton algorithm has high complexity (a further analysis is presented in Section 4.7), whereas our approach is relatively simple. Moreover, in the texton method, the process of getting to these vectors is quite time consuming indeed and is subject to numerous constraints, where as our approach is not. Further discussion and comparative analysis will follow in Section 4.7.

In sum, our model for each class is represented by the 2 Weibull parameters for each of 38 filter responses, for 46 training images (see Section 4.6.2) within that class.

4.6 Classification Method

4.6.1 Distance in the Weibull Space

We note that the information entropy for a Weibull distribution is defined as in equation 4.5:

$$H = \lambda \left(1 - \frac{1}{k}\right) + \log\left(\frac{\lambda}{k}\right) + 1 \tag{4.5}$$

where $k > 0$ is the shape parameter, $\lambda > 0$ is the scale parameter of the distribution, and γ is the Euler-Mascheroni constant with numerical value 0.577 (to 3 decimal places) in equation 4.5 [27].

The information entropy measurement captures the information stored in a Weibull distribution represented by a pair of {shape, scale} parameters. Over the range of shape and scale parameters actually determined, for modeling the 38 filter-response distributions by using Weibull probability distribution functions, we find experimentally that the entropy H is monotonic with Weibull pairs {shape,scale}. We found this distance measure to actually produce more accurate classification results and therefore we adopt it here (it would indeed seem that this particular distance metric has not been used previously and is new to this research). Therefore, we use equation 4.5 as a distance measure between two images in our Weibull space. I.e., for every image we have a 38-vector of entropies (one for each shape, scale pair). Now for every test image we measure the L_2 distance between its vector of entropies and that of a training image. We classify the test image according to the class of its nearest neighbour amongst the training images.

4.6.2 Experimental Setup

We follow the experimental setup of [42] in order to compare our results with theirs and other previous results of texture classification (S,LM).

We perform three experiments to assess texture classification rates over 92 images for each of 20, 40 and 61 texture classes respectively. The first experiment, where we classify images from 20 textures, corresponds to the setup employed by [6] which is also used by [42]. The second experiment, where 40 textures are classified, is modelled on the setup of [30] also used by [42]. In the third experiment, we classify all 61 textures present in the Columbia-Utrecht database which corresponds to the setup employed by [42]. The 92 images are selected as follows: for each texture in the database, there are 118 images where the viewing angle θ_v is less than 60 degrees. Out of these, only those 92 are chosen for which a sufficiently large region could be cropped across all texture classes. The resultant modified CURET dataset could be found at [25].

Each experiment consists of two stages: generating a model for the class, with texture models learnt from training images, and classification of novel images.

The 92 images for each texture are partitioned into two, disjoint sets. Images in the first (training) set are used to generate the final model for the class and classification accuracy is only assessed on the 46 images for each texture in the second (test) set. Both sets of images sample the variations in illumination and viewpoint.

Each of the 46 training images per texture defines a model for that class as follows: the image is mapped (through convolving with RFS, then histogramming, and Weibull fitting) to its location in the Weibull space. Thus, each texture class is represented by a set of 46x76 vectors of Weibull parameters shape and scale.

An image from the test set is classified by first mapping it to the Weibull space and then choosing its nearest neighbour in this space from the training set. The distance function used to define closest is that based on an entropy measure, as explained in the previous section.

In the three experiments, we form our Weibull space from 20 textures, 40 textures, and 61 textures respectively.

In the first experiment, 20 textures are chosen (see fig. 19a in [6] for a list of the novel textures) and 20x46 = 920 novel images are classified in all. In the second experiment, the 40 textures specified in fig. 7 of [30] are chosen and a total of 40x46 = 1840 novel images classified. Finally, in the third experiment, all 61 textures in the Columbia-Utrecht database are classified using the same procedure.

To compare the run-time of our algorithm with that of [42] we conducted our experiments on a Windows based system with Intel 2.2GHz processor, 2GB of RAM running Matlab 7.1. We selected a set of 480 training and test samples and ran the classification procedure multiple times under consistent experimental environment to generate the average run times per texture for the algorithms (see Table 4.2).

4.7 Results and Critical Analysis

The results (percentage accuracy of classifying test images) of all three experiments are presented in Table 4.1. The first point we note from Table 4.1 is that in case of 20 texture classes our method, achieves classification accuracy rates very close to that of S, and LM, notwithstanding its simplicity and much faster speed (see Table 4.2). It is better than the MR4 approach in all cases and only slightly (2%) worse than MR8 for the 20 class case.

However, for the case with all 61 classes in the database our method is some 5% worse than MR8. We will come back to this point but first we present the execution times per texture in Table 4.2.

Table 4.1: Comparison of Classification Accuracy Percentages for Varying Number of Texture Classes

	# of Texture Classes		
Approach	20	40	61
S	96.30	95.27	94.62
LM	96.08	93.75	93.44
MR4 (200 Textons)	94.13	92.07	90.73
MR8 (200 Textons)	97.83	96.41	96.40
MR8 (610 Textons)	-	-	96.93
Our Weibull based	95.98	92.28	91.52

Table 4.2: Execution Times Per Texture

Approach	Model Generation	Classification
Our Weibull based	2.7s	8×10^{-4}s
MR8 (610 Textons)	26s	4×10^{-3}s
MR8 (200 Textons)	22s	1.4×10^{-3}s

The method proposed here is almost 5 times as fast as that of [42] (the MR8 approach) though when classifying all 61 texture classes our method does lose 5% accuracy. Of course, when using a smaller number of texture classes (for example, 20) our accuracy is very close to that of MR8 but our method is 10 folds faster. We analyze the dimensionality and complexity of MR8 further and compare it to ours.

Up until the point of generating filter responses to the 38 filters from RFS both our approach and MR8 have identical complexity. Let ρ be the number of pixels in each filter response.

In MR8, the authors contrast-normalize the filter responses but we do not as, from our experiments, this step reduces the classification accuracy. The contrast-normalized filter responses are reduced to the maximum response set in MR8. But we keep all the filter responses. Both these steps add a complexity of $O(\rho)$ to the MR8 method.

At the next step, MR8 clusters the filter responses using a standard k-Means technique. There are a number of reliability issues related to this step. In the first experiment, the authors in MR8 (and also CD, LM) select a set of 20 classes from which they generate their texton dictionary through k-Means clustering. This particular choice of 20 classes has provided excellent accuracy for the CURET dataset but doubts remain. It is unreasonable to assume that, given a novel texture that is not present in this database, textons generated

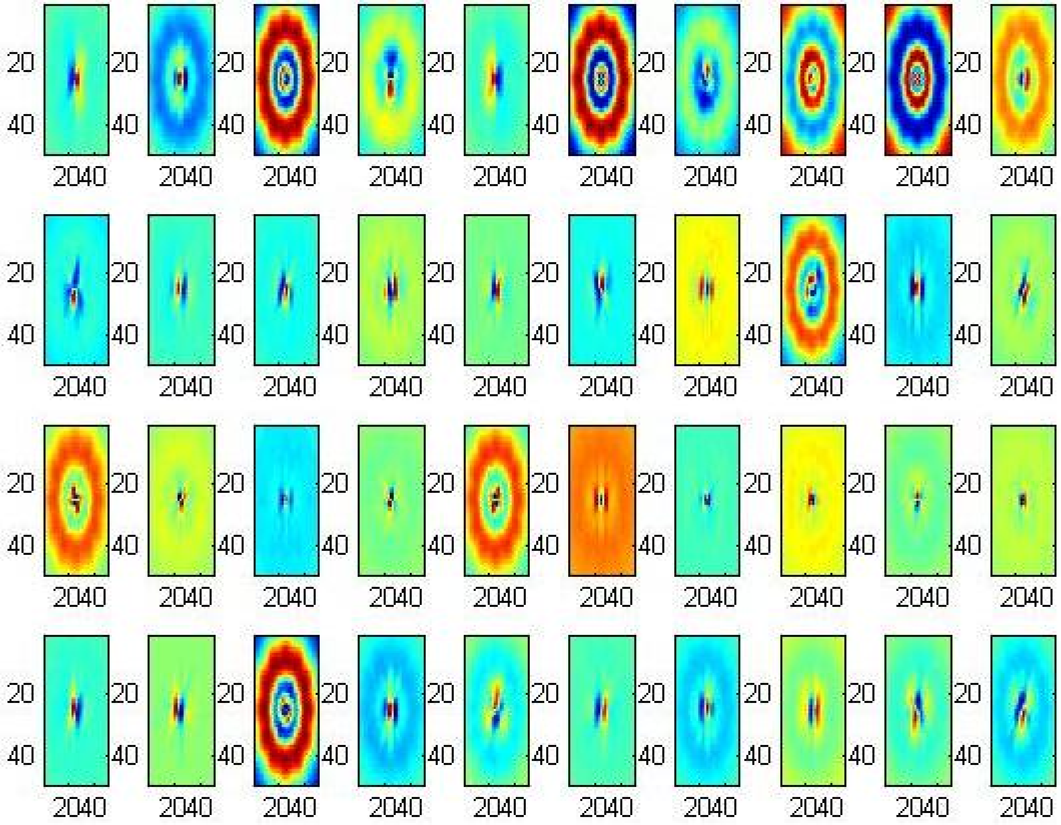

Figure 4.4: 10 Textons generated from Felt (first row), Polyester (second row), Terry Cloth (third row), and Rough Plastic (fourth row) by the MR8 method. Each texton is of dimension 49 × 49.

from these 20 classes will produce similar levels of accuracy on the new texture as well. Moreover, the process of generating the textons through the use of a standard k-Means technique is possibly problematic. A standard k-Means technique is not guaranteed to converge in polynomial time. It has been shown by [2] that with high probability the k-Means algorithm may converge in super-polynomial time. They also go on to prove that the worse case complexity of k-Means on n data points is $2^{\Omega(\sqrt{n})}$. We have that n for the MR8 approach is $13\rho/N$ where N determines the number of sample pixels kept from each filter response (and 13 sample images are chosen at random from each class). This would generate 10 textons for a particular class and the process has to be repeated for all 20 texture classes and for different samples, further adding to the complexity. Given that the clustering method is not guaranteed to converge, especially under such high dimensions, the ordering and initialization of the data becomes very critical. For example a simple experiment where we re-order the initial neighbours, such as pixels from the filter response, shows that the generation process of textons/cluster centres are adversely effected. Different initial ordering of data points will generate different cluster centres. If the kMeans algorithm converges to a global minimum then the initial ordering of data points has no effect. Of course, the authors set all these parameters through empirical evidence. Figure 4.4 shows 10 textons generated from 4 different textures (felt, polyester, terry cloth, rough plastic) from the CURET dataset.

In contradistinction to these problems, our method extracts meaningful information from each filter response without being dependent on so many parameters. The transformation of a filter response to the Weibull space involves histogramming the data first. This gives rise to the question of how many bins should there be in the histogram and what should be the size of each of them? Interestingly, in a fairly exhaustive set of experiments we found that it does not matter what the number of bins are as long as they are above a certain threshold (in our case this happens to be 1001). This is primarily due to the information present in the filter responses from specific filters and the Weibull fit process (essentially a least squares type estimate). Increasing the number of bins does not improve or reduce classification accuracy. Even decreasing the number of bins to as low as 201 only slightly reduces the classification accuracy (for 61 classes the accuracy drops by 0.04% only). So our algorithm is independent of the number of bins in the histogram. This can also be noticed from Figure 4.5. In this figure, the key thing to note is that for various number of bins (201,1001,5001) the estimated Weibull parameters, shape and scale, are exactly the same (shape=1.204, scale=0.025). Each histogram represents responses to a particular filter which captures edge distributions at a particular orientation and scale. In order to facilitate comparison of shape and scale of different distributions caused by different filters each histogram is shifted so that it is centered on zero. Hence each histogram is peaked at zero.

The second important criterion for our proposed approach is the convergence of the Newton-Raphson method while estimating the shape parameter of the Weibull distribution. Although we allow a maximum number of iteration of 30, in practice for 99.73% cases of the 213,256 filter responses (61 classes, 92 images from each class, 38 filter responses for each image) present in the dataset, the Newton-Raphson method converges in 5 iterations or less. Compare this to the convergence problems associated with k-Means. Moreover, this

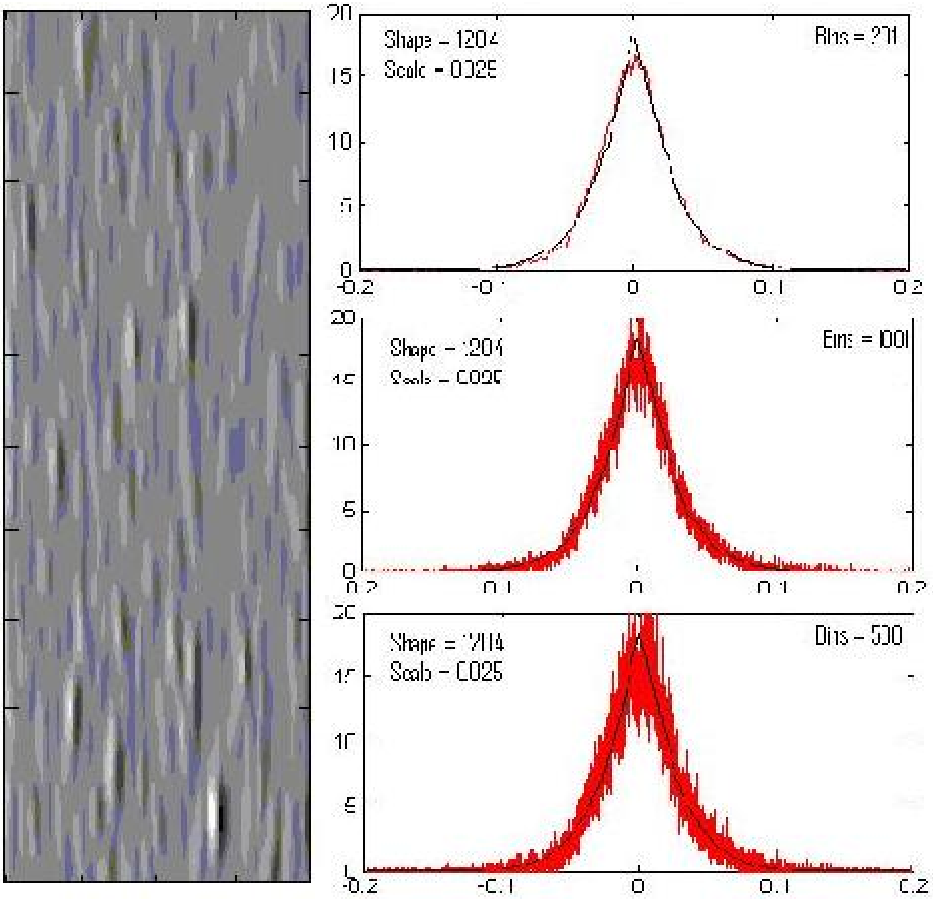

Figure 4.5: Invariance to Number of Bins. The 1st column shows a sample Filter Response, the 2nd column is Demonstrating the Probability Density Function (the black line) generated from the Estimated Weibull Parameters for the Filter Response with Varying Number of Bins

simply reiterates the point made in [24] that the textured surfaces present in CURET are Weibull distributed.

Some other techniques are also proposed by [42] to further reduce the dimensionality of their problem by trying to reduce the number of models representing each texture class using a greedy algorithm. We omit further discussion about these steps primarily because it adds to the complexity of their approach while benefiting categorization results only marginally.

4.8 Summary

We have presented in this chapter an algorithm that models the edge distributions of a texture surface for classification purposes. The model generated, due to its simplistic nature, is very fast to compute and improves classification speed significantly with very small reduction in accuracy compared to the state-of-the-art.

Chapter 5

Experiments on Object Recognition

5.1 Motivation

In this chapter we extend our experiments to the domain of object recognition. We use for purposes of our experiments object images from Amsterdam Library of Object Images (ALOI) [22]. Particular reason for using this dataset is that it captures objects under varying illumination angle, thus causing objects to be visible only partially from certain camera positions.

Using the ALOI dataset we further prove the utility of illumination-invariant images by observing that parts of objects that are normally invisible due to the position of the illuminating device become visible again in such images. Figure 5.1(a),(b), and (c) demonstrates the effect of our standardized method which reduces the effect of illumination to a great extent without any prior knowledge about the camera or imaging conditions.

However, as Figure 5.1(d) demonstrates, the grayscale image generated by converting the illumination-invariant image into grayscale using the following standard equation 5.1 (as in Matlab 7.1) generates a very poor image indeed in terms of capturing the features of the original object.

$$I(x,y) = 0.2989 \times R(x,y) + 0.5870 \times G(x,y) + 0.1140 \times B(x,y) \tag{5.1}$$

where $I(x,y)$ is the output grayscale value at pixel location x,y. $R(x,y)$, $G(x,y)$ and $B(x,y)$ are the red, green, and blue channel values respectively at pixel location x,y. So we use

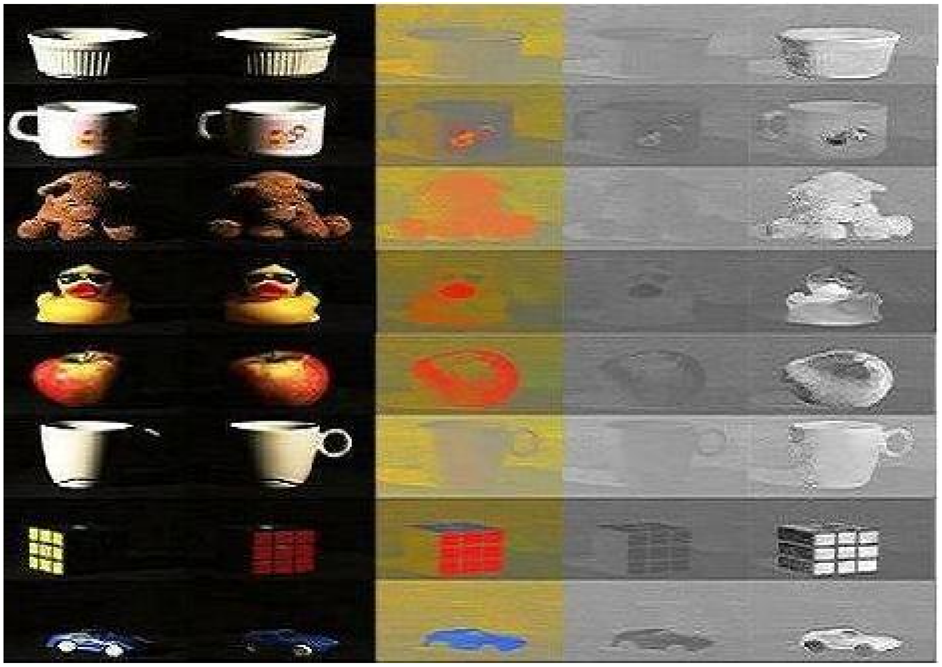

Figure 5.1: Objects 32,35,36,62,92,125,200,482 from ALOI. (a) In col 1 Objects Illuminated from the Left. (b) In col 2 Objects Illuminated from the Right. (c) The Illumination-invariant image obtained from image in col 2. (d) Ordinary Grayscale of the Invariant Image. (e)Improved Grayscale of the Invariant Image.

the method proposed in [1] to generate the greyscale image from the invariant image. This simple method based on recognizing the maximum gradient over color channels produces a much improved greyscale image (see Figure 5.1(e)) and is very fast too.

We also note that our method of texture classification classifies textured surfaces based on information about edge distributions at various orientations and scale. At a fixed pose, objects which are from the same class should have similar distributions of edges, and objects which are from different classes should be dissimilar on the same measure given that we can observe sufficient parts of the object. Therefore, the Weibull parameters generated must be similar for objects of the same class and dissimilar otherwise.

Figure 5.2 demonstrates this. We see that in Figure 5.2(b) the plots of the Weibull shape parameters are very similar for the illumination-invariant image of objects within the same class. But in Figure 5.2(c) we see that without applying the standardized method to the object image and computing the Weibull shape parameters from the greyscale image cause the plots to be quite different even for objects from the same class.

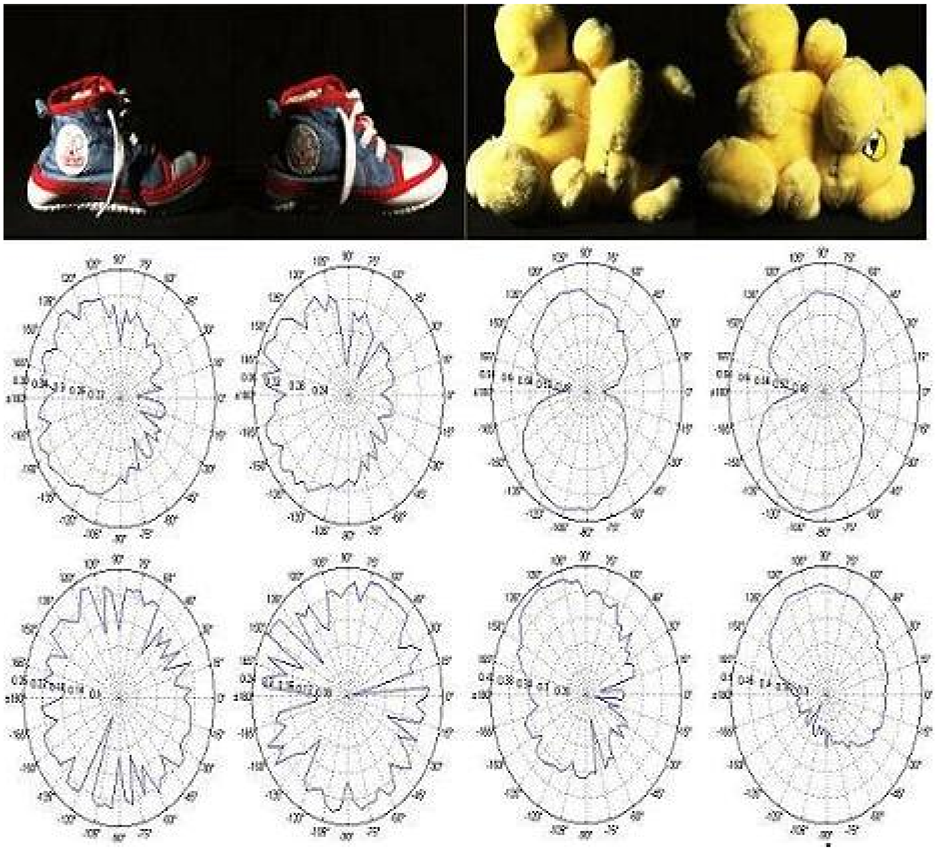

Figure 5.2: Weibull Shape Parameter Plotted at 72 different Orientations for the Bar Filter at Scale = 1. (a)row 1 has the same object with illumination from left and right respectively. (b)row 2 has the Weibull Shape Parameters of the Illumination-invariant Image of the Object in row 1. (c)row 3 has the Weibull Shape Parameters of the Greyscale Image (nonillumination-invariant) of the Object in row 1.

5.2 Experiments and Critical Analysis

ALOI has 1000 classes of objects, each captured under various illumination angles among other illumination conditions. Figure 5.3 and Figure 5.4 demonstrate this.

We use images from camera 1 in our experiments. Our model for each class consists of only one of these 8 images for that class as the training image and the other 7 images are put into the test set for each class to test the classification accuracy of our methods.

The RFS set of filters are used, filter banks are L_1 normalized and filter responses are not contrast normalized. Filter responses are histogrammed into histograms with 1001 equally spaced bins. Weibull parameters are then estimated from these histograms.

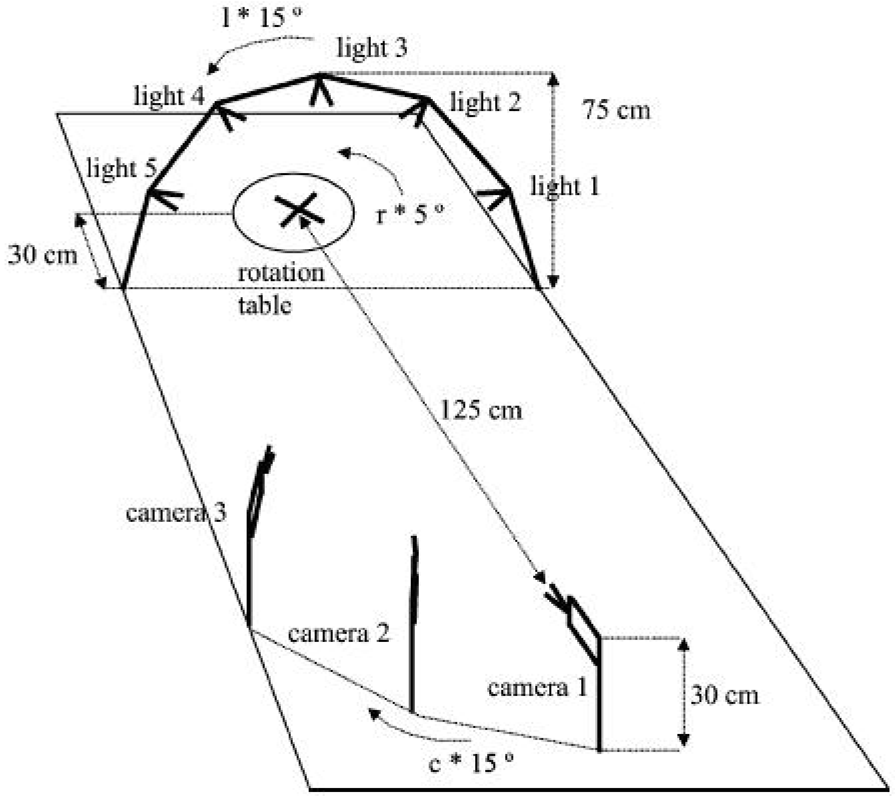

Figure 5.3: Experimental setup for capturing the ALOI collection [22].

The training model image for a class is mapped to its location in the Weibull space. Thus, each object class is represented by a 76 vector of Weibull parameters (38 shape and 38 scale parameters for the 38 filter responses).

An image from the test set is classified by first mapping it to the Weibull space and then choosing its nearest neighbour in this space from the training set. The distance function used to define closest is that based on an entropy measure, as explained in Chapter 4.6.

In each experiment we classify $1000 \times 7 = 7000$ images. We use L8 from camera 1 in Figure 5.4 as our model and put the other 7 images from camera 1 in our test set.

In the first experiment, we operate in a non-illumination-invariant environment, that is an image is converted to grayscale using equation 5.1 and all processing is done on this. In the second experiment, we apply our standardized method to each image and operate in an illumination-invariant environment now. However, the grayscale is still generated from the invariant image using equation 5.1. In our third and final experiment we use the method

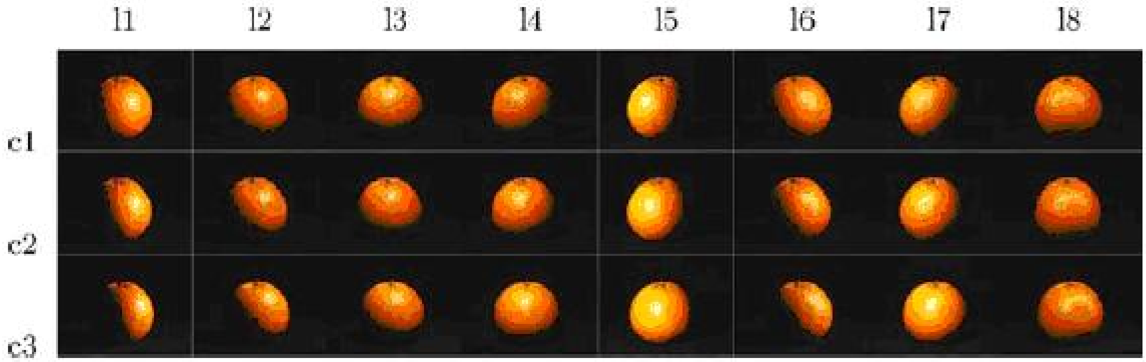

Figure 5.4: Example object from ALOI viewed under 24 different illumination directions. Each row shows the recorded view by one of the three cameras. The columns represent the different lighting conditions used to illuminate the object [22].

proposed in [1] to generate the grayscale image from the invariant image.

Table 5.1: Classification Accuracy of Experiments on Object Recognition

	Approach
Experiments	Our Weibull based
Experiment 1 (No intrinsic image with grayscale from equation 5.1)	27.65%
Experiment 2 (Intrinsic image with grayscale from equation 5.1)	32.03%
Experiment 3 (Intrinsic image with grayscale from [1])	42.87%

The results are presented in Table 5.1. We observe that the use of illumination-invariant images improve the classification accuracy of our Weibull based approach. The intrinsic images we form make more of the object visible and thus recover more of the general shape of the object which in turn helps our Weibull based approach in classifying with higher accuracy.

Chapter 6

Conclusion

We have outlined a simple, standardized method to generate an illumination invariant, from input colour images. The method is based on the idea of simply treating every input image as inhabiting sRGB colour space, and transforming that space. The transformation is found by optimizing the lighting invariance for generic, synthetic data when taken to log-chromaticity space and projected into a 1D invariant. The invariant image itself can be understood as a 2D-colour chromaticity image. Experiments show that applying the standardized invariant extraction method generates reasonable independence to lighting, across conditions and cameras. The approach set out here may be usefully employed in place of a more rigorous, camera- and image-dependent method for extraction of an illumination invariant.

We have also set out a new texture categorization method that unifies the texton approach with an approach that recognizes that distributions are often well represented by the Weibull distribution. Consequently, we can dispense with a good deal of the complexity of the texton approach while maintaining comparable classification accuracy, notwithstanding a substantial speedup in the algorithm. This texture recognizer can easily be incorporated in any multimedia search and retrieval system that utilizes a texture component. The new entropy-based similarity measure has not been suggested before for judging nearness of distributions.

Finally we have tested the performance of our illumination-invariant images and our classification approach in the domain of object recognition and have observed that classification performance improves through the use of illumination-invariant images.

Bibliography

[1] Ali Alsam and Mark S. Drew. Fast colour2grey. In *16th Color Imaging Conference: Color, Science, Systems and Applications.*, pages 342–346. Society for Imaging Science & Technology (IS&T)/Society for Information Display (SID) joint conference, 2008.

[2] David Arthur and Sergei Vassilvitskii. How slow is the k-means method? In *SCG '06: Proceedings of the Twenty-Second Annual Symposium on Computational Geometry*, pages 144–153, New York, NY, USA, 2006. ACM.

[3] H.G. Barrow and J.M. Tenenbaum. Recovering intrinsic scene characteristics from images. In *Computer Vision Systems*, pages 3–26, 1978.

[4] David H. Brainard and Brian A. Wandell. Analysis of the retinex theory of color vision. *J. Opt. Soc. Am. A*, 3(10):1651–1661, 1986.

[5] International Electrotechnical Commission. Multimedia systems and equipment – colour measurement and management – part 2-1: Colour management – default RGB colour space – sRGB. IEC 61966-2-1:1999.

[6] Oana G. Cula and Kristin J. Dana. 3d texture recognition using bidirectional feature histograms. *International Journal of Computer Vision*, 59(1):33–60, 2004.

[7] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, 1999.

[8] Mark S. Drew, Chao Chen, Steven D. Hordley, and Graham D. Finlayson. Sensor transforms for invariant image enhancement. In *Tenth Color Imaging Conference: Color, Science, Systems and Applications.*, pages 325–329. Society for Imaging Science & Technology (IS&T)/Society for Information Display (SID) joint conference, 2002.

[9] Mark S. Drew, Graham D. Finlayson, and Steven D. Hordley. Recovery of chromaticity image free from shadows via illumination invariance. In *IEEE Workshop on Color and Photometric Methods in Computer Vision, ICCV'03*, pages 32–39, 2003.

[10] Graham D. Finlayson and Mark S. Drew. Constrained least–squares regression in color spaces. *Journal of Electronic Imaging*, 6:484–493, 1997.

[11] Graham D. Finlayson and Mark S. Drew. 4-sensor camera calibration for image representation invariant to shading, shadows, lighting, and specularities. In *ICCV'01: International Conference on Computer Vision*, pages II: 473–480. IEEE, 2001.

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

[13] Graham D. Finlayson, Mark S. Drew, and Cheng Lu. Intrinsic images by entropy minimization. In *ECCV 2004: European Conference on Computer Vision*, pages 582–595, 2004. Lecture Notes in Computer Science Vol. 3023.

[14] Graham D. Finlayson and Steven D. Hordley. Colour constancy at a pixel. *J. Opt. Soc. Am. A*, 18(2):253–264, Feb. 2001. Also, UK Patent #2360660, "Colour signal processing which removes illuminant colour temperature dependency".

[15] Graham D. Finlayson, Steven D. Hordley, and Mark S. Drew. Removing shadows from images. In *ECCV 2002: European Conference on Computer Vision*, pages 4:823–836, 2002. Lecture Notes in Computer Science Vol. 2353.

[16] Graham D. Finlayson, Steven D. Hordley, and Mark S. Drew. Removing shadows from images using Retinex. In *Tenth Color Imaging Conference: Color, Science, Systems and Applications.* Society for Imaging Science & Technology (IS&T)/Society for Information Display (SID) joint conference, 2002.

[17] Graham D. Finlayson, Steven D. Hordley, Cheng Lu, and Mark S. Drew. On the removal of shadows from images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 28:59–68, 2006.

[18] Graham D. Finlayson, Cheng Lu, and Mark S. Drew. Invariant image improvement by sRGB colour space sharpening. In *AIC 2005: The Tenth Congress of the International Colour Association*, 2005.

[19] Graham D. Finlayson and Sabine Süsstrunk. Performance of a chromatic adaptation transform based on spectral sharpening. In *Eigth Color Imaging Conference: Color, Science, Systems and Applications.*, pages 49–55. Society for Imaging Science & Technology (IS&T)/Society for Information Display (SID) joint conference, 2000.

[20] Charles Fowlkes, David Martin, Xiaofeng Ren, and Jitendra Malik. Detecting and localizing boundaries in natural images. Technical report, University of California at Berkeley, 2002.

[21] Clement Fredembach and Graham D. Finlayson. Hamiltonian path based shadow removal. In *16th British Machine Vision Conf, BMVC2005*, pages 970–980, 2005.

[22] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The Amsterdam library of object images. *Int. J. Comput. Vis.*, 61(1):103–112, 2005.

[23] Jan-Mark Geusebroek and Arnold W.M. Smeulders. Fragmentation in the vision of scenes. *International Conference on Computer Vision*, 1:130–135, 2003.

[24] Jan-Mark Geusebroek and Arnold W.M. Smeulders. A six-stimulus theory for stochastic texture. *International Journal of Computer Vision: Special Issue on Texture Analysis and Synthesis*, 62(1-2):7–16, 2005.

[25] Visual Geometry Group. Modified curet color textures. Downloaded from http://www.robots.ox.ac.uk/ vgg/research/texclass/index.html, 2008.

[26] Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(6):610–621, 1973.

[27] Julian Havil. *Gamma: Exploring Euler's Constant.* Princeton University Press, 2003.

[28] Anil K. Jain. *Fundamentals of Image Processing.* Pearson Education, 2004.

[29] Bela Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, March 1981.

[30] Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Comput. Vision*, 43(1):29–44, June 2001.

[31] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *Int. J. Comput. Vision*, 43(1):7–27, 2001.

[32] David Marr. *Vision.* W.H. Freeman, 1982.

[33] David R. Martin, Charles C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(5):530–549, 2004.

[34] G. Matheron. Eléments pour une théorie des milieux poreux. 1967.

[35] C.S. McCamy, H. Marcus, and J.G. Davidson. A color-rendition chart. *J. App. Photog. Eng.*, 2:95–99, 1976.

[36] Majid Mirmehdi, Xianghua Xie, and Jasjit Suri. *Handbook of Texture Analysis.* World Scientific Publishing, 2008.

[37] Maria Petrou. *Image Processing: Dealing with Texture.* Wiley, 2006.

[38] Azriel Rosenfeld and Avinash C. Kak. *Digital Picture Processing.* Academic Press, 1976.

[39] Peter J. Rousseeuw and Annick M. Leroy. *Robust Regression and Outlier Detection.* Wiley, 1987.

[40] Cordelia Schmid. Constructing models for content-based image retrieval. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–39–II–45 vol.2, 2001.

[41] Mihran Tuceryan and Anil K. Jain. *Handbook of Pattern Recognition and Computer Vision.* World Scientific Publishing, 1993.

[42] Manik Varma and Andrew Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision: Special Issue on Texture Analysis and Synthesis*, 62(1-2):61–81, 2005.

[43] J.S. Weszka, C.R. Dyer, and A. Rosenfeld. A comparative study of texture measures for terrain classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 6(4):269–286, April 1976.

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