Modeling Scene Text and Texture by Decomposing into Component Images

Thesis submitted in partial fulfillment
of the requirements for the degree of

MS by Research
in
Computer Science

by

Siddharth Kherada
200702048
siddharth.kherada@research.iiit.ac.in

Centre for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
December 2013

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Modeling Scene Text and Texture by Decomposing into Component Images” by Siddharth Kherada, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Prof. Anoop M. Namboodiri

To My Parents and My Advisor

Acknowledgments

I would like to express my sincere gratitude to my advisor Prof. Anoop Namboodiri for his continuous guidance, support, patience and motivation. His immense knowledge in the area of research and his guidance has helped me throughout the course of this research. He stood by me and supported me in my good and bad times.

I was also fortunate enough to have friends and colleagues at CVIT who have played different roles at different times. They helped me out whenever I was stuck. I would like to thank - Abhinav Goel, Vinay Garg, Shrikant Baronia, Rohit Nigam, Harshit Sureka, Rohit Gautam, Shashank Mujumdar, Akhil Vij, Rohan Kulkarni, Srijan, and Harshit Agrawal. Many thanks to Prof. P.J. Narayanan, Prof. C. V. Jawahar and Prof. Jayanthi for their encouraging presence and for providing an environment conducive to learning of the finest quality at CVIT.

I would like to thank my friends Varun, Ayush, Ankur, Sankalp, Sachin, Gaurav, Ammar, Rahul, Rakshit for making this time worth remembering.

Finally, I would like to thank my parents and sister for their support in my academic and research pursuits. They are and will always be a continuous source of motivation and inspiration in all the ventures of my life. Many thanks to everyone else who affected my life in any way, and wasn't acknowledged personally above.

Abstract

Separation of images into its constituent components based on source of data, frequency distribution, or nature of data has been a widely used technique in the field of image processing and computer vision. Many problems are solved by partitioning images into components and working on each component separately. Common examples include breaking down of image into Red, Green and Blue channels/components for ease of representation or into Luminance and Chrominance for better compression. In this thesis, we explore the separation of natural images into appropriate components for the purpose of representation as well as recognition. We first introduce a framework where separation of images into direct and global components helps in modeling of 3D textures. These 3D textures are often described by parametric functions for each pixel, that models the variation in its appearance with respect to varying lighting direction. However, parametric models such as Polynomial Texture Maps (PTMs) tend to smoothen the changes in appearance. Therefore we propose a technique to effectively model natural material surfaces and their interactions with changing light conditions. We show that the direct and global components of an image have different characteristics, and when modeled separately, leads to a more accurate and compact model of the 3D surface texture. Direct component is mainly affected by structural properties of the surface and is therefore deals with phenomena like shadows and specularity, which are sharply varying functions. The global component is used to model overall luminance and color values, a smoothly varying function. For a given lighting position, both components are computed separately and combined to render a new image. This method models sharp shadows and specularities, while preserving the structural relief and surface color. Thus rendered image have enhanced photorealism as compared to images rendered by existing single pixel models such as PTMs.

We then look at separating an image based on its sources of illumination or albedo variations for the purpose of scene text segmentation. Extracting text from scene images is a challenging task due to the variations in color, size, and font of the text and the results are often affected by complex backgrounds, different lighting conditions, shadows and reflections. A robust solution to this problem can significantly enhance the accuracy of scene text recognition algorithms leading to a variety of applications such as scene understanding, automatic localization and navigation, and image retrieval. We propose a method to extract and binarize text from images that contains complex background. We use Independent Component Analysis (ICA) to map out the text region, which is inherently uniform in nature, while removing shadows, specularity and reflections, which are included in the background. The technique identifies the text regions from the components extracted by ICA using a simple global thresholding method to

isolate the foreground text. We show the results of our algorithm on some of the most complex word images from the ICDAR 2003 Robust Word Recognition Dataset and compare with previously reported methods.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A digital image is a discrete sampled representation of a two-dimensional function ($I_{ij} \in \mathcal{R}^d$) Each element, I_{ij} is referred to as a pixel, which takes a d-dimensional value depending on the function that it tries to model. In the case of a natural color image, each pixel captures the intensity and color of the light ray that travels from a specific world point to the camera center. A popular way to represent these properties of the light ray is to measure the intensity of the red, green and blue components of the light ray. One could then think of the complete image as composed of three component images: a red image, a green image and a blue image. Another way of representation may instead capture the Hue and Saturation of the color in two components and the intensity in a third component. Given a specific representation, one can often convert the image into other representations (See Figure 1.1).

A specific representation may be more appropriate in solving a specific problem, while not so well suited for others. For example, the $L^*a^*b^*$ representation is designed to accurately measure the perceptual differences between a given pair of colors, while the CMY component representation is better suited for printing of images. In this thesis, we look at decomposing an image into components based on the surface properties that is represented as well as the illumination of the surface. We show that appropriate decomposition can help in modeling the observed surface for enhanced photo-realistic rendering as well as the segmentation of scene text from shadows, reflections and complex backgrounds.

1.1 Analysis of Texture in Scene Images

Texture refers to a surface characteristic and appearance of an object given by its geometry, density and surface reflectance, and the stochastic variation of these parameters. It is a detailed pattern that is mapped into a multidimensional space. It is an important cue in trying to achieve photo realistic rendering of 3D models by adding surface details or color to an object or a scene. Some of the natural scene textures are shown in Fig 1.2

3D Texture modeling is an important area in computer graphics as it results in realistic rendering of natural material surfaces. The characterization of surface reflectance properties is important in achieving photorealism. The appearance of a surface in different lighting and viewing direction/conditions is

(a) Input Color Image

(b) Red

(c) Green

(d) Blue

(e) Hue

(f) Saturation

(g) Intensity

Figure 1.1 A natural color image, that is represented by various decompositions into component images. (b)-(d) RGB components of the image (e)-(g) HSV components of the image

affected by its reflectance properties. 3D texture actually models the relation between surface reflectance properties and illumination direction.

Mapping 2D textures or images on to a 3D surface is the most common method used, which is efficient for most 3D models and scenes, especially where the lighting conditions remain constant. They look best when the object is viewed in similar lighting conditions as when the texture is captured. They appear flat and smooth. In practice, the real world surfaces are characterized by phenomena such as inter-reflection, self-shadowing, subsurface scattering, specularity, etc. These properties interact with different lighting directions and therefore the same surface appears different under different lighting condition Fig 1.3. 2D texture fails to capture these complex reflectance properties of a surface and therefore a rendered surface looks highly unrealistic in case the lighting conditions are changed. In order to produce a realistic rendering it is necessary to capture and model the interaction of the material surface with different lighting conditions. [30] investigates the problem of representation, recognition, synthesis of natural materials and their rendering under arbitrary viewing/lighting conditions.

3D textures are a way to model this relation between surface reflectance properties and illumination/viewing conditions. The use of 3D texture modeling results in enhanced realism of the scene. Reflectance texture maps are one of the techniques that can be used to compactly represent the 3D tex-

Figure 1.2 Natural Scene Textures

tures. These maps are generated using image re-lighting techniques [3, 33, 10] in which multiple images are captured under different lighting conditions.

Image based modeling techniques [5, 52, 58] have emerged as an effective approach for realistic rendering of 3D objects, where multi-view geometry is utilized in directly synthesizing an unseen view of an object from nearby views without explicit surface reconstruction. The traditional object models capture the shape information in the meshes, while the reflectance and the surface properties are relegated in the textures. 3D models such as Polynomial Texture Maps (PTM) capture the surface properties more faithfully, including the effect of small scale height variation on the surface.

Polynomial Texture Maps [33] belong to the class of UTFs (Uni-directional Texture Function). It is a pixel based technique that concisely models the surface reflectance properties using a polynomial model for the reflectance, dependent on two angular parameters of the lighting direction (l_u and l_v). PTMs reconstruct the color of the surface under varying lighting conditions and models real world phenomenon such as self-shadowing, inter-reflection and sub-surface scattering. They thus introduce enhanced photorealism in texture mapping. Polynomial Texture Mapping is applied in a wide range of archaeological contexts [12]. It also offers advantages over traditional raking light photography for examining and documenting the surface texture and shape of paintings [46]. Recently, PTMs have been used in cultural heritage field to document and virtually inspect several sets of small objects, such as cuneiform tablets and coins.

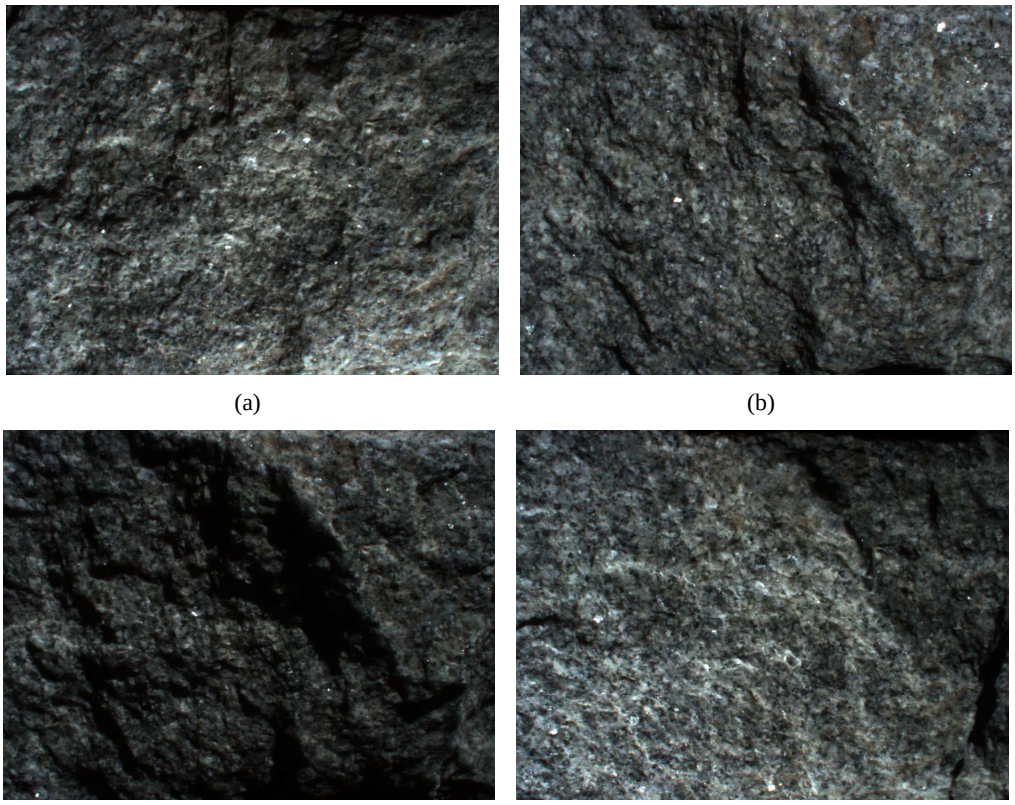


Figure 1.3 Variation in appearance of the same surface patch, when illuminated from different lighting directions

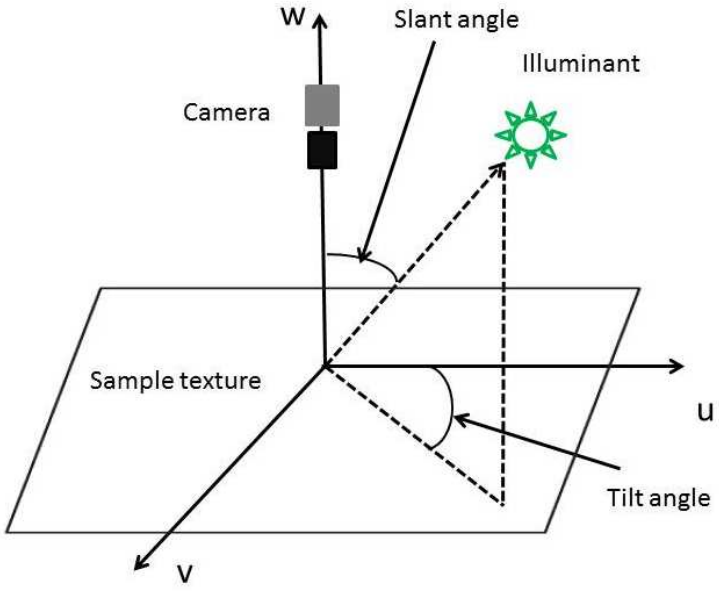


Figure 1.4 Experimental setup

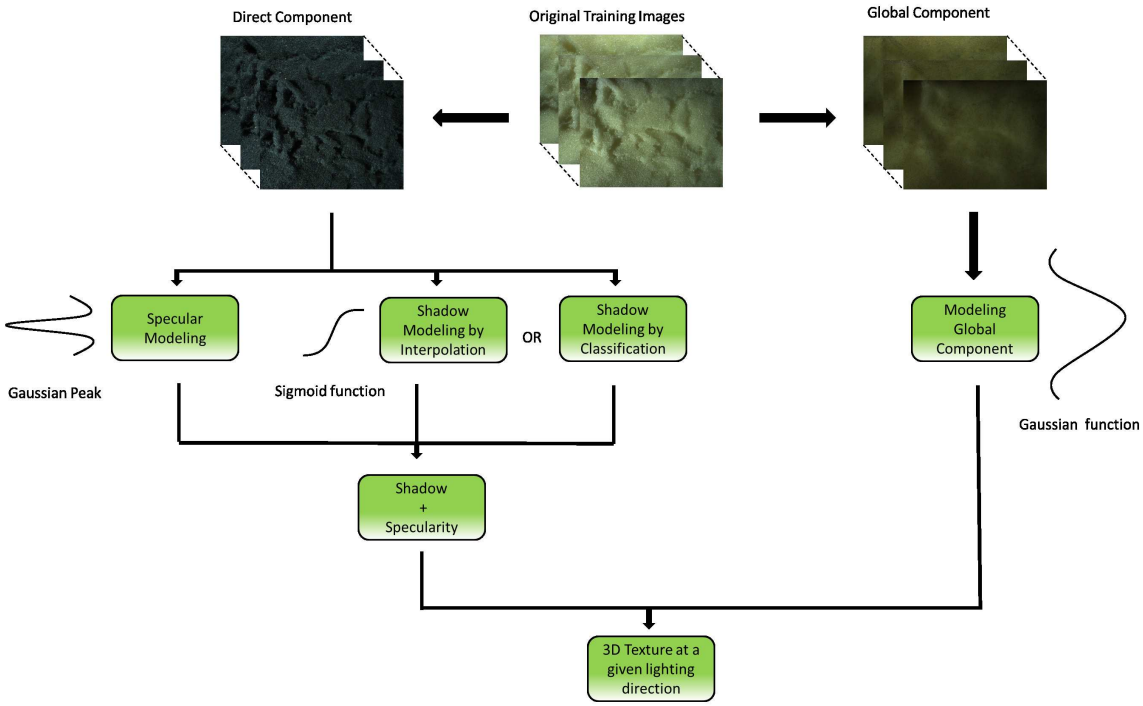


Figure 1.5 Component Based Modeling (CBM)

1.1.1 Problem

Texture mapping is an important area in computer graphics which adds realism to three dimensional models. 2D texture mapping fails to capture the surface variation and reflectance properties under varying lighting and viewing direction. They appear good only when viewed from similar lighting direction in which they were captured and fails to provide the information required for rendering other than the original illumination condition. But 3D textures correctly models the relation between surface reflectance properties and illumination conditions. The use of 3D texture modeling results in enhanced realism of the scene.

But PTM technique causes overall smoothening of light which dampens the effect of specularity and softens sharp shadows. The effect of point light source is reduced and the appearance is always similar to a diffused light source. Therefore we improve upon the PTM model to overcome the above limitations and generate a complete 3D Texture model that can be evaluated at individual pixels. We propose an approach to image-based lighting interpolation that is based on estimates of geometry and shading from a set of input images.

1.1.2 Approach

We capture multiple images of a static object with a static camera under varying lighting conditions. The scene is illuminated using a high frequency checkerboard pattern using the projector. The projec-

tor is moved to different lighting positions for the purpose of obtaining images with different lighting directions. Experimental setup is shown in Figure 1.4.

When we separate the image of the texture into direct and the global part, we find that the shadows and the specularity appear very strongly in the direct part, as these are phenomena that involve light that reaches the surface point directly from the light source. The fine details and the structure of the material are very prominently visible in the direct part as they are observed primarily through shadows. On changing the lighting direction, the change in the luminance of the direct part is minimal as long as the surface point is directly illuminated. The variations are introduced, primarily by self-shadowing and specularity, both of which are abrupt changes as the lighting direction changes. The global component contains the lighting of a surface point from other parts in a scene, and hence it captures the overall illumination as well as color variations of a surface with lighting direction. As the lighting direction changes, the luminance value of the global part varies significantly.

Both direct and global components are separately analyzed to derive the corresponding models and parameters. Given a new lighting direction, we use the two models separately to generate the corresponding components, and combine them to get the final image. Then for a new lighting direction, we can readily interpolate both specular content as well as shadows. The main contributions of this work are (1) Direct and Global modeling characterized by shadows, specularity and luminance, (2) separate modeling and hence better capturing of shadows and specularity and (3) per pixel function model to achieve real-time rendering of enhanced 3D textures on GPU. The method is shown to indeed generate better results for non-observed lighting directions. A complete flowchart of our model is shown in Fig 1.5

1.2 Analysis of Text in Scene Images

Scene text is textual content that is captured by a camera in an image. Reading text captured in images provides valuable information and is used in many content-based image applications such as content based web image search, information retrieval and mobile based text analysis and recognition.

In the recent years, content-based image analysis techniques have received more attention with the advent of various digital image capture devices. Applications of text localization and recognition in real-world images ranges widely. It is used for indexing large image databases by their textual content, sign recognition for foreigners, automatic license plate recognition, assisting the elderly and visually impaired in reading labels.

The images captured by these devices can vary dramatically depending on lighting conditions, reflections, shadows and specularity. Some natural scene text images are shown in Fig 1.6. These images contain numerous degradations such as uneven lighting, complex background, multiple colors, blur etc. We propose a method which removes reflections, shadows and specularity from natural scene text images and binarize the text from a single image. Binarization method is one of the important preprocessing steps in document image analysis system. It directly affects the performance of the subse-

Figure 1.6 Natural Scene Text Images

quent step which is text recognition. Binarization of text can be defined as classifying individual pixels as foreground (text) or background.

There are many algorithms that aim at extracting foreground text from background in images but thresholding remains one of the oldest form that is used in many image processing applications. Many sophisticated approaches often have thresholding as a pre-processing step. It is often used to segment images consisting of bright objects against dark backgrounds or vice versa [17, 54, 47]. It typically works well for images where the foreground and background are clearly defined. For color thresholding images, most algorithms convert the RGB image into grayscale but here we will make use of the RGB channels as three different sources/components.

Traditional thresholding based binarization can be grouped into two categories: the one which uses global threshold for the given images like Otsu [45], Kittler *et al.* [25] and the one with local thresholds like Sauvola [56], Niblack [42]. In global thresholding methods [45, 53], global thresholds are used for all pixels in image. These methods are fast and robust as they use a single threshold based on the global histogram of the gray-value pixels of the image. But they are not suitable for complex and degraded scene images. Also selecting the right threshold for the whole image is usually a challenge because it is difficult for the thresholding algorithm to differentiate foreground text from complex background.

On the other hand, local or adaptive binarization [6] methods changes the threshold over the image according to local region properties. Adaptive thresholding addresses variations in local intensities throughout the image. In these methods, a per-pixel threshold is computed based on a local window

Figure 1.7 Scene text images containing complex background

around each pixel. Thus, different threshold values are used for different parts of the image. These methods are proposed to overcome global binarization drawbacks but they can be sensitive to image artifacts found in natural scene text images like shadows, specularities and reflections. On the other hand, we propose a method that removes shadows, specularity and reflections and thus produces a clean binary images even for the images with complex background.

1.2.1 Problem

The primary issue related to segmenting text from scene images is the presence of complex/textured background. Some of the challenges are described below:

- **Noise:** Sensor noise in hand held camera devices is usually high.
- **Viewing Angle:** While capturing images, the text and the camera device may not be parallel.
- **Blur:** Some motion blur can appear or be created by a moving object. Camera shake with hand-held shots can result in blurry images. Camera not equipped with auto-focus also blurs the image.
- **Resolution:** Depending on the capture device, resolutions of the images can vary from low to high.

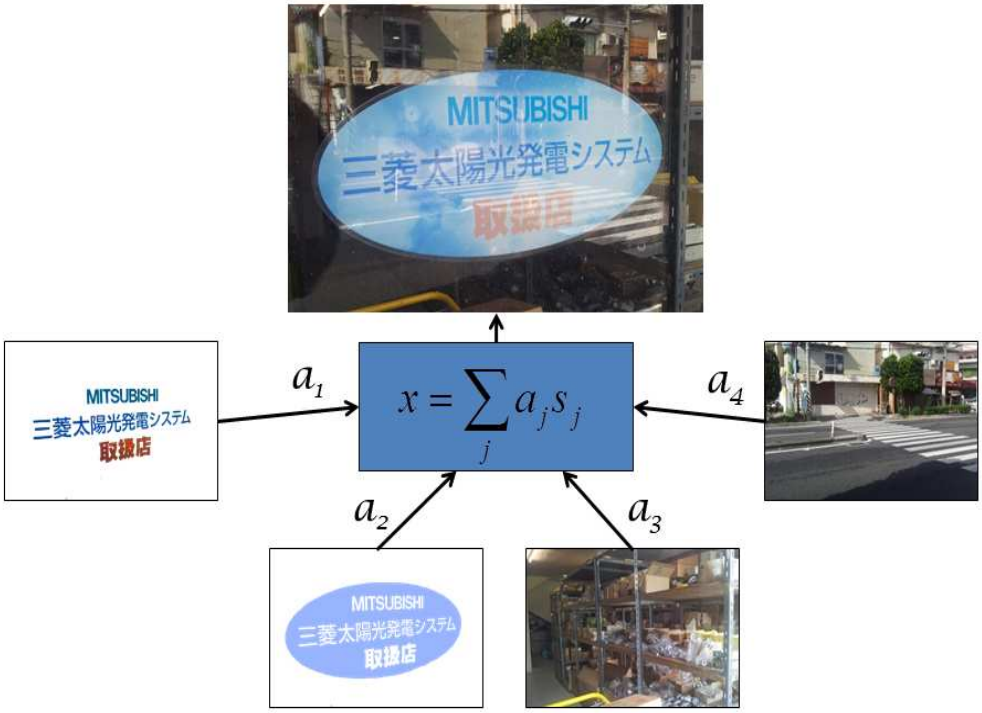


Figure 1.8 ICA model applied on images

- **Non-planar surfaces:** Text written on non-planar objects like bottles can suffer from deformation.
- **Low contrast:** It is difficult to extract text when there is not much color difference between the foreground text and the background.
- **Lighting:** Images containing uneven lighting, reflections, shadowing, highlights make colors of the text vary drastically and thus decreases analysis performance. Both the physical environment and uneven response/artificial lighting from the camera devices are responsible for complexing the task of text extraction.

Some of the challenges that we encounter while segmenting natural scene text are shown in Fig 1.7.

1.2.2 Approach

We apply an Independent Component Analysis (ICA) model on natural scene images. It is one of the most important methods of blind source separation and has received attention in the field of signal processing, pattern recognition, data compression and image analyzing. Using ICA, we can extract the source signals from the observations based on the stochastic property of the input signals/images even without any information of the original source signals. ICA method obtains features that present the

data through a set of components that are statistically independent and characterizes the data in a natural way. ICA based decomposition enables us to separate text from complex backgrounds containing, reflections, shadows and specularities. Fig 1.8 shows the basic ICA model applied on images. For binarization, we apply a global thresholding method on the independent components of the image and that with maximum textual properties is used for extracting the foreground text. Binarization results show significant improvement in the extraction of text over other reported methods.

1.3 Outline

In this chapter, we briefly analyzed textures and text in scene images. The rest of this thesis is organized as follows: We will survey techniques related to texture mapping and scene text understanding in Chapter 2. In Chapter 3 component based texture modeling will be discussed where we will show how each component is modeled differently to achieve photorealism. In Chapter 4, we will discuss component based text segmentation from natural scene images. Finally we conclude the thesis summarizing our main contributions and discuss future work in Chapter 5.

Chapter 2

Background and Related Work

2.1 Texture in Scene Images

In computer graphics, texture mapping is a powerful tool which adds the surface detail to an object by wrapping the color information from a digitized image. Texture is applied on top of a polygon or 3D model to obtain a realistic rendering of it. This makes rendering of objects more realistic than those without surface texture. Scene images are extensively used as source of textures as they are able to capture visual and structural information of the real world. They are also able to capture a high level detail of object properties. Generally an image is used as a texture map on a planar surface.

There are several texture mapping methods that avoids modeling of the complex surface details. One of them is image-based rendering technique which is frequently used in computer graphics in the form of texture mapping. The images are used to represent the appearance of a complex object. It is a technique where images are used in place of complex geometry and material properties. Images are a quick way to achieve photorealism as our primary objective is to get a realistic rendering. This is because they are able to capture complex light interactions such as inter-reflections, self-shadowing and sub-surface scattering present in the real world. The main disadvantage of using images as rendering source is that the image captures the appearance of the object from a single viewpoint and under a fixed lighting condition/direction. Therefore this method fails if the lighting conditions of the synthetic environment are different from the lighting conditions of the texture image. This is usually the case when we map 2D textures onto 3D models under varying lighting directions.

In 2D texture modeling, the reflectance and the structural properties of natural surfaces are not captured. They fail to capture the variations in surfaces for different lighting and viewing directions. The texture which is mapped onto a 3D model has the lighting direction from which it was captured. So if we want to see how the texture looks from different lighting direction, this mapping will give poor results when viewed from different lighting direction apart from the direction from which it was captured. In general, real world objects are not flat and smooth in nature. They show different types of structural variation across their surface each having different reflectance properties. These properties causes effects like shadows, specularity, sub-surface scattering, inter-reflection etc. Hence 3D texture

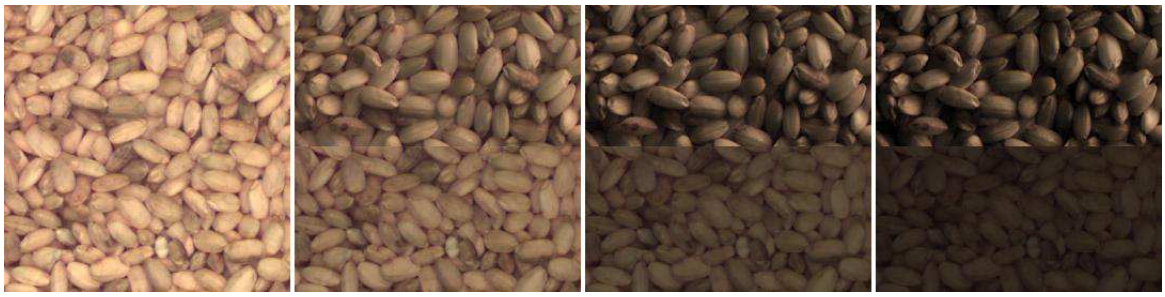


Figure 2.1 3D vs 2D texture map: The upper part of the images shows the visual appearance of a 3D texture map while the bottom part shows the conventional 2D texture map. We can see that the bottom part suffers from unrealistic lighting and shadows.

mapping is required for realistic modeling of real objects. It results in realistic rendering of natural material surfaces. The characterization of surface reflectance properties is important in achieving enhanced realism of the scene. The appearance of a surface in different lighting and viewing direction/ conditions is affected by its reflectance properties. 3D textures are a way to model the relation between surface reflectance properties and illumination/viewing conditions. Fig 2.1 shows the difference between a 3D texture map and a 2D texture map. To solve this problem, techniques based on reflectance texture maps have been proposed.

Reflectance texture maps are one of the techniques that can be used to compactly represent the 3D textures. 3D textures actually models the relation between surface reflectance properties and illumination direction. Hence they can be represented by Reflectance Texture Maps. The maps are generated using image re-lighting techniques which model the surface reflectance properties of object. These maps are created by capturing multiple images under different lighting/viewing conditions.

Reflectance map required in 3D texture can be modeled by Bidirectional Reflectance Distribution Function (BRDF) [43] technique which defines spectral and spatial reflectance characteristic of a surface. Bidirectional Reflectance Distribution Function is defined as the ratio of reflected radiance to incident irradiance, given the lighting and viewing directions.

Various techniques have been developed to compactly represent BRDF [57, 26, 51, 18] BRDF was extended to Bidirectional Texture Function (BTF) [10] by allowing BRDF to vary spatially across planar texture co-ordinate (u,v).

BTF effectively captures view point dependent phenomenon such as specularity along with other physical phenomenon such as shadow, sub-surface scattering, inter-reflection, etc. However, the capture of BTF requires careful camera calibration and capturing numerous images for sampling. Generating reflectance map from BTF is very complex. Because of the high dimensionality of the BTF and high storage requirement, Unidirectional Texture Functions (UTF) were introduced in which viewing point is not taken into account while modeling the surface reflectance properties. They model these properties only in relation to different lighting conditions. As the visual appearance of the surface is mostly independent of the viewing direction, the model provides a reasonable approximation of the surface with

Figure 2.2 Natural textures mapped onto a 3D model of teapot under varying lighting directions

significantly lower complexity of the model. Fig 2.2 shows the 3D texture modeling. 3D textures are mapped onto a 3D teapot model.

Polynomial Texture Maps [33] belong to the class of UTFs. It is a pixel based technique that concisely models the surface reflectance properties using a polynomial model for the reflectance, dependent on two angular parameters of the lighting direction. It is used to model luminance against changing lighting direction and requires no modeling of complex geometry. Only a set of images is required of the desired scene to be used as a texture which are taken from different known lighting direction. PTMs reconstruct the color of the surface/texture under varying lighting conditions and models real world phenomenon such as self-shadowing, inter-reflection and sub-surface scattering. When a surface is rendered with a PTM, it takes on different illumination characteristics depending on the direction of the light source. They thus introduce enhanced photorealism in the texture mapping process.

PTM model uses a set of input images captured from a fixed camera, where each image is illuminated from a specific known lighting direction. It uses a biquadratic polynomial function with 6 coefficients per pixel for modeling the reflectance. These coefficients are estimated from the set of input images(30 to 40), where the lighting direction is resolved into two components i.e l_u,l_v by projecting it on the image plane. These two components are used as variables in the biquadratic function. Once the coefficients

are estimated fitting the model to the observed values, they are used to render images from any given lighting direction. The function used in PTMs is:

$$L(l_u, l_v) = al_u^2 + bl_v^2 + cl_u l_v + dl_u + el_v + f \tag{2.1}$$

But the PTM technique causes overall smoothening of light which dampens the effect of specularity and softens sharp shadows. The effect of point light source is reduced and the appearance is always similar to a diffused light source. The current state-of-the art in the field of PTM involves robust method for interpolation of shadow and specularity, Drew *et al.* [11]. But they do not model natural material surfaces and their interactions with changing light conditions. Moreover the number of per pixel parameters are too high for real-time rendering

We improve upon the PTM model to overcome the above limitations and generate a complete 3D Texture model that can be evaluated at individual pixels. We propose an approach to image-based lighting interpolation that is based on estimates of geometry and shading from a set of input images. We decompose images captured at different lighting conditions into intrinsic image components; i.e, the *direct* and *global* image components. Each of these components is then further separated to obtain different physical phenomena such as shadows, specularity and luminance. A final image is obtained by combining the the individual models together.

2.2 Text in Scene Images

The aim of scene text localization and recognition is to find all areas in an image that can be considered as text, mark boundaries of the areas (rectangular bounding boxes) and output the word meanings of the detected content. As the digital imaging devices are becoming cheaper and widely available, the size of the available digital image content is increasing rapidly. The textual information present in images is very useful and needs to be extracted. Therefore there is a growing demand to analyze, process and retrieve information from multimedia content in an efficient way. Some of the natural scene text images are shown in Fig 2.3.

Recognizing text in scene images is a challenging task. This is because the quality and content of images are rather unpredictable. There are many variations in backgrounds (non-uniform), textures, fonts and lighting conditions that are present in images. To build a full end-to-end text recognition system, we need to develop models which are robust to these variations. They should read all the text within the image and extract information.

Most of the work on scene text recognition tends to focus on a sub-component of the full end-to-end text recognition system. The general problem of end-to-end text recognition consists of three primary components:

- **Text detection:** Is there any text and where is it in the image? Locate it.

Figure 2.3 Some natural scenes containing text

- **Text Segmentation:** How to handle all the variations in background to extract foreground text as properly as possible.
- **Text recognition:** What is the meaning of the word images?

In text localization, the goal is to locate individual words or lines of text in the image. The next step i.e text segmentation job is to extract the text properly so that it is better recognized afterwards. The extracted text goes to OCR for recognition. Finally we identify the actual word meanings and lines of the text. The recognized text is then used in various applications. An example of the end-to-end text recognition model is presented in Fig 2.4

2.2.1 Text Detection and Recognition

As mentioned above, the goal of text detection or localization is to identify regions of text in a given scene image. The task of detection is to identify rectangle bounding box for each word or each line of text in the image. There are methods which focus on general text localization problem. It can be categorized into two groups - (a) methods based on a sliding window and (b) methods based on grouping of regions.

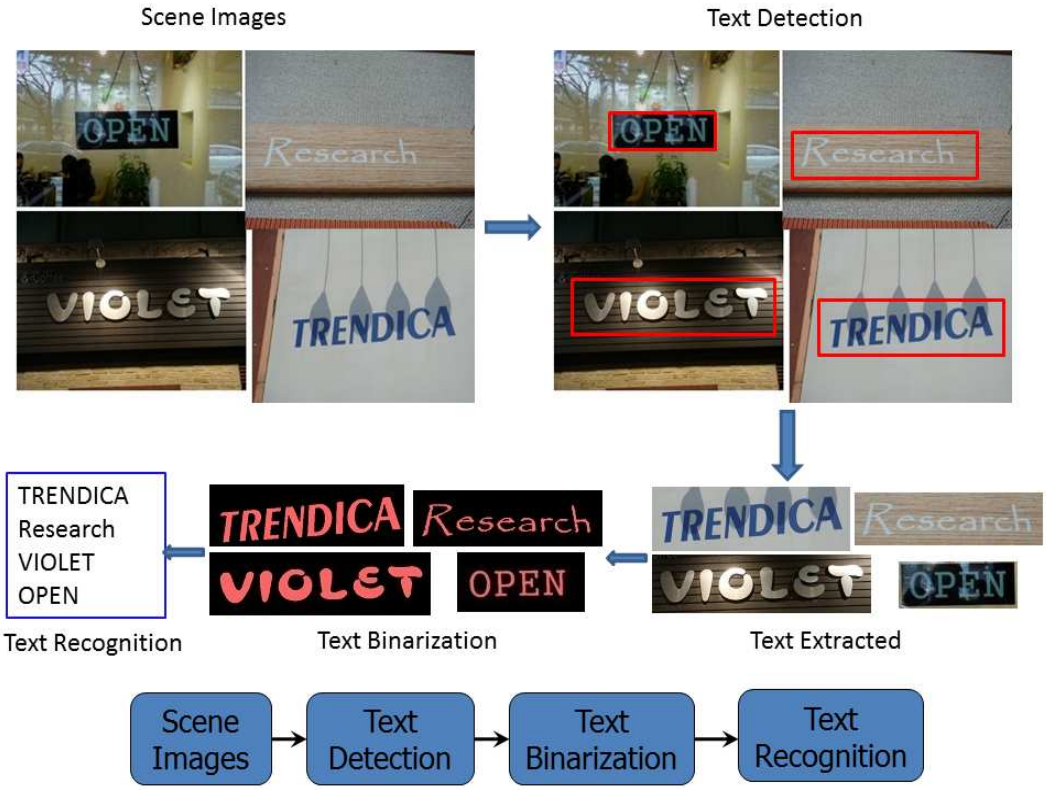


Figure 2.4 An end-to-end text recognition model

Methods in the first group [8, 49, 28] use a texture-based approach for text detection. A window is moved over the whole image and the position of text is estimated on the basis of local image features. These methods are robust to noise. But the computational complexity of these methods is high because we need to search the text in image with many rectangles of different sizes and aspect ratios. Also if the text is slanted or perspectively distorted, then the sliding window methods do not produce accurate text segmentation results. Chen *et al.* [8] uses AdaBoost classifier having intensity variance features, histogram features, mean intensity features, derivative features and edge linking features. They use a variant of Niblack's adaptive binarization algorithm for segmentation. The method is computationally expensive. It also requires manual segmentation of many sub-windows for training purposes and it seems to over estimate the area of text in the image. The above method was improved by Pan *et al.* [49] by adding a combination multi scale local binary pattern and histogram of gradient features in the text detection stage. A Markov Random Field (MRF) is used to group segmented characters into words. The method claims better localization performance but it still suffers from high computational complexity. Recently, Lee *et al.* [28] further improved the approach by adding more computationally expensive features which slightly improved the text localization performance but it took longer time for processing. Lienhart *et al.* [32] uses a complex-valued multilayer feed-forward network which is

Figure 2.5 Text detected in Scene Images

trained to detect text at a fixed scale and position. Coates *et al.* [9] used unsupervised machine learning techniques for character detection and recognition. A 32x32 pixel window is shifted over the entire image in multiple scales and each patch is classified using a linear SVM classifier as text or non-text. A variant of K-means method is used to generate features during training which are then used by the classifier. However, the method does not provide end-to-end text recognition.

Recently published methods are based on the second category i.e region grouping [50, 65, 39, 13, 36]. In these methods, certain local features are computed for each pixel in the image and then pixels with similar feature values are grouped together using connected component analysis to form characters. But these methods are sensitive to noisy and low-resolution images. Ohya *et al.* [44] was the first method which was based on character localization. In this method, they apply a local adaptive thresholding on grayscale images to detect candidate regions. The regions which have sufficient contrast are termed as characters. Li *et al.* [16] [31] apply thresholding in a quantized color space. Then they group individual characters into text blocks by some simple alignment rules. Both of the above methods assume that the background is uniform and characters are upright i.e without any rotation. But this is not the case for general text localization. Kim *et al.*[24] combine three independent detection channels i.e edge detection, color continuity and color variance to find regions of possible text. These regions are grouped into blocks by size and position. Each block is divided into overlapping 16x16 pixel sub-blocks, which are verified by a trained SVM classifier using wavelet transform to generate features. The block is

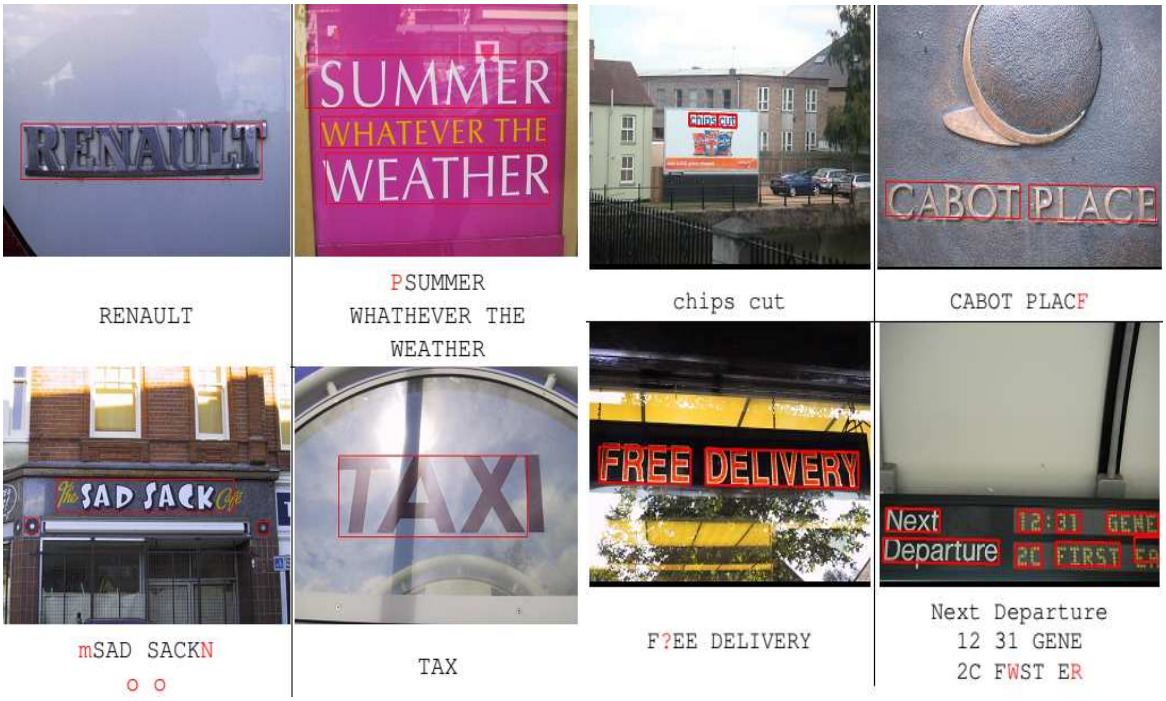


Figure 2.6 Text recognized in Scene Images

marked as text if the ratio of sub-blocks marked as text is higher than a pre-defined threshold. The method is not scale-invariant as the size of sub-blocks is constant. Pan *et al.* [50] uses a Waldboost classifier with histogram of gradient as features. They create a text confidence map on a grayscale image pyramid. Possible text regions are detected independently on a grayscale image using Niblack's binarization algorithm. A Conditional Random Field (CRF) is used to label regions as text or non-text. To form block of texts, a simple gradient graph energy minimization approach is applied. Epshtein *et al.* [13] introduced an image operator: Stroke Width Transform (SWT). The SWT method uses Canny edge detector to find edges and estimates stroke width for each pixel in the image. A Connected component algorithm is then applied to form pixels with similar stroke width into character candidates. These are merged into text blocks using several heuristic rules. This method depends on the success of edge detection which normally fails on noisy, blurred and low-contrast images. Yao *et al.*[65] further improved this method. They replaced the heuristic rules for character candidate detection and text block formation by trained classifiers with rotationally invariant features. Fig 2.5 shows text detected in scene images.

The methods listed above are focused only on text localization. They estimate the location of the text but do not provide the contents meaning. Mishra *et al.* [36] proposed an effective method to recognize scene text. Their model combines bottom-up cues from character detections and top-down cues from lexica. Neumann and Matas [39] proposed an end-to-end method for scene text localization and recognition. They used hypotheses-verification framework for processing multiple text line. They

Figure 2.7 A comparison of scene text segmentation results. From left to right (a) Text Image (b) kittler (c) Niblack (d) Otsu (e) Sauvola

use synthetic fonts to train the algorithm thus eliminating the need for time consuming acquisition and labeling of real-world training data. They also use Maximally Stable Extremal Regions (MSER) which are robust to geometric and lighting conditions. A real time text localization and recognition was proposed by Neumann and Matas [40]. This is achieved by posing the character detection problem as an efficient sequential selection from the set of Extremal Regions (ERs). The ER detector is robust to illumination, blur, color and texture variation and also handles low-contrast text. Neumann and Matas [41] presented an unconstrained end-to-end text localization and recognition method. The method introduces a novel approach for character detection and recognition which combines the advantages of sliding-window and connected component methods. Fig 2.6 shows text recognized in scene images.

2.2.2 Text Segmentation

The aim of text segmentation is to split an image into regions of foreground text and background. If the text is extracted properly it can be better recognized afterwards. OCR is able to recognize text if the input image is well formated and binarized. Therefore we want to extract clean binary text images from complex backgrounds.

Thresholding is the basic method used for segmentation. [60] does a survey of thresholding based methods. Fig 2.7 shows thresholding based binarization results. Recently, several methods for natural scene text binarization have been proposed. Wakahara *et al.* [63] proposes a technique which is composed of three parts: They first generate tentative binarized images via dichotomization of k clusters obtained by k-means clustering in the HSI color space. Then with the help of support vector machine (SVM), they determine determine the degree of "character-likeness" of each tentatively binarized image. Finally the binarized image having maximal degree of "character-likeness" is selected as optimal binarization result. To segment text information from camera-based images Thillou *et al.* [62] develops an automatic color thresholding method based on wavelet denoising and color clustering with K-means.

Zhou *et al.* [66] proposed a new text segmentation method based on inverse rendering (decomposition of an input image into basic rendering elements). The method uses iterative optimization to solve the rendering parameters like material properties (for example diffuse/specular reflectance) , blur kernel size and light source. Field *et al.* [14] describes a novel technique for text segmentation that models smooth color changes across image. They use bilateral regression technique for segmentation of text from complex background. Milyaev *et al.* [34] proposed a new binarization method that works well for text in natural scene images. The method embeds local binarization into a global optimization framework. No information about the position and size of the text in an image is required. It can also be used for text localization and for recognition of the cropped text. Mishra *et al.* [35] has formulated the problem of binarization of text as an MRF optimization problem. The method shows superior performance over traditional binarization methods on many images and we use it as the basis for our comparisons.

2.3 Summary

In this chapter, we introduced the problem of texture modeling. We saw how scene images can be used as source of textures as they are able to capture visual and structural information of the real world. We showed the difference between the traditional 2D and 3D texture maps. We looked at how 2D textures results in unrealistic and incorrect rendering while 3D textures models the relation between surface reflectance properties and illumination/viewing conditions thus resulting in realistic rendering. We described the techniques like Bidirectional Reflectance Distribution Function (BRDF) and Polynomial Texture Maps (PTM) used for modeling 3D textures. We showed the disadvantages of using PTM model and how we propose to improve it. We also looked into end-to-end text recognition model. We saw how text detection helps in identifying regions of text in a given scene image. A word recognizer then identifies the segmented words and finds the underlying word meaning. We surveyed recent techniques applied to text detection, text segmentation and text recognition stages of the recognition model. In the next chapter, we propose a new framework where separation of images into *direct* and *global* components helps us in better modeling of 3D textures.

Chapter 3

Component Based Texture Modeling

The appearance of the texture in a given lighting condition is characterized by shadows, specularity and overall luminance. The luminance is affected by subsurface scattering and inter-reflection properties of the surface. PTM does not separately take these properties into account and models them together using a biquadratic function. However, the nature of variation of the reflected light is significantly different for these phenomena. In our method, we analyze each of these phenomena separately and capture the results using appropriate models.

3.1 Separation into components

We first separate the images into two components: one is the direct part, which is controlled by the reflectance of a surface point and the structural properties of its neighborhood, while the second is the global part that captures overall luminance. Separation of a scene into global and direct part can be done by illuminating the scene with a high frequency binary pattern [38]. The direct part captures the light that is directly reflected by the surface point from the source whereas the global part is due to the illumination of the point from all other points of the scene (see Fig 3.1). The direct and global components of the scene are shown in Fig 3.2.

In our experiments, we used checkerboard pattern that were 10x10 pixels in size and was shifted by 5 times (by 3 pixels each time) in each of the two dimensions to capture a total of 25 images. The separation step is given in Fig 3.3.

If we separate the image of the texture into direct and the global part, we find that the shadows and the specularities appear very strongly in the direct part, as these are phenomena that involve light that reaches the surface point directly from the light source. The fine details and the structure of the material are very prominently visible in the direct part as they are observed primarily through shadows. On changing the lighting direction, the change in the luminance of the direct part is minimal as long as the surface point is directly illuminated. The variations are introduced, primarily by self-shadowing and specularity, both of which are abrupt changes as the lighting direction changes. The surface variations in rough surfaces or textures leads to interesting self shadowing effects.

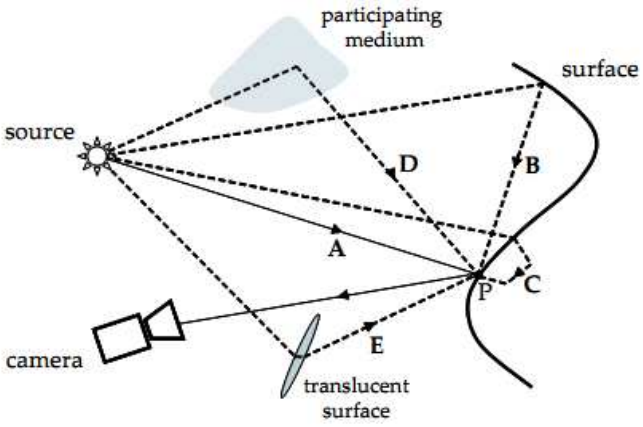


Figure 3.1 The luminance of scene point is due to direct illumination of the point by the source (A) and global illumination due to other points in the scene which is mainly due to inter-reflections (B), subsurface scattering (C), volumetric scattering (D) and translucency (E) [38]

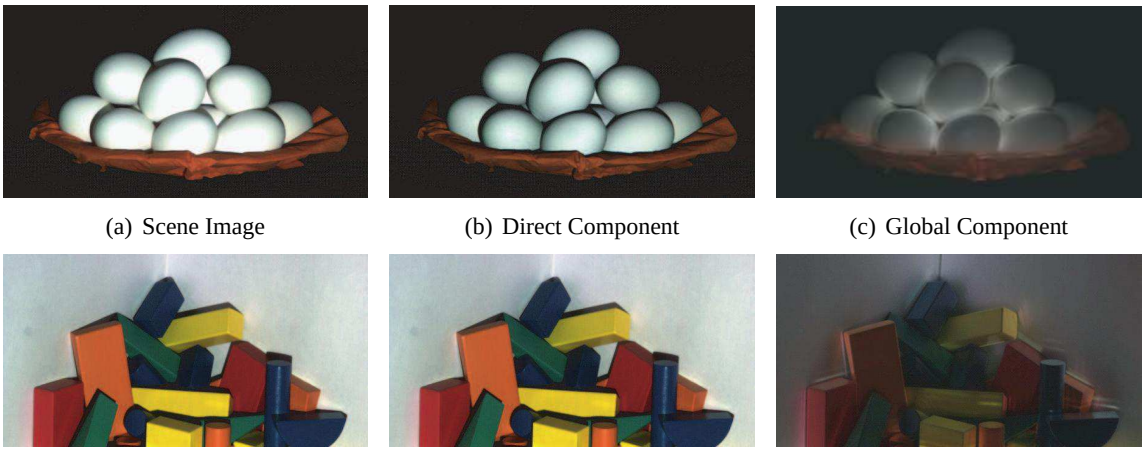
The global component contains the lighting of a surface point from other parts in a scene, and hence the captures the overall illumination as well as color variations of a surface with lighting direction. As the lighting direction changes, the luminance value of the global part varies significantly.

Both direct and global components are separately analyzed to derive the corresponding models and parameters. Given a new lighting direction, we use the two models separately to generate the corresponding components, and combine them to get the final image.Then for a new lighting direction, we can readily interpolate both specular content as well as shadows.

3.2 Modeling Direct Component

As noted before, the direct component is affected by the phenomena of self-shadowing and specularity, in addition to the lambertian reflectance of the surface point. Shadows are the points that receive no direct light from the primary source. However, their luminance value is not completely zero. This is because they get some light from the neighboring pixels because of inter-reflections. However, when the image is decomposed into direct and global component, the luminance value of shadow region (due to inter-reflections) appear in global part and thus direct part is left with dark prominent shadow regions whose value is near to zero (see Fig 3.5). These dark shadow regions can easily be separated out using thresholding.

A major difficulty in lighting interpolation is the realistic generation of shadows. To compute shadow masks for real scenes, our approach first infers shadow pixels from the illumination intrinsic image. The intensities in an illumination intrinsic image represent magnitudes of incident irradiance, so image areas with low values indicate shadowed regions.



(a) Scene Image (b) Direct Component (c) Global Component

(d) Scene Image (e) Direct Component (f) Global Component

Figure 3.2 Direct and global components of a scene [38]

A variety of shadow detection techniques have been proposed in the past to capture this phenomenon in a realistic fashion [55, 23, 22, 29]. In our approach, as the direct component receives only light directly from the source, a simple thresholding is quite effective in addition to being efficient. More details on various shadow algorithms and their complexities can be found in [64]. Therefore a simple thresholding is done to separate out the shadow part. A suitable threshold is determined using Otsu's method [45]. Once the shadow regions are detected in each of the images, we proceed to capture the variations of it with lighting direction. We discuss two distinct approaches for this purpose, each with its own merits and short-comings, and show how they can be combined to derive a good shadow model.

3.2.1 Shadow Modeling by Interpolation

Consider a pair of images of a surface captured from the same view point, but by moving the light source through a short distance. We note that each pixel (surface point) belong to one of the three categories:

1. Pixels that are not in shadow in either image.
2. Pixels that are in shadow in both images.
3. Pixels that are in shadow in only one of the images.

For the first two types of pixels, the behavior of the pixels remains same as the lighting direction changes from one image to other. i.e, the pixels that are in shadow continue to remain in shadow and that illuminated remain illuminated, provided the distance between the two images being interpolated is small. The values of these pixels change smoothly.

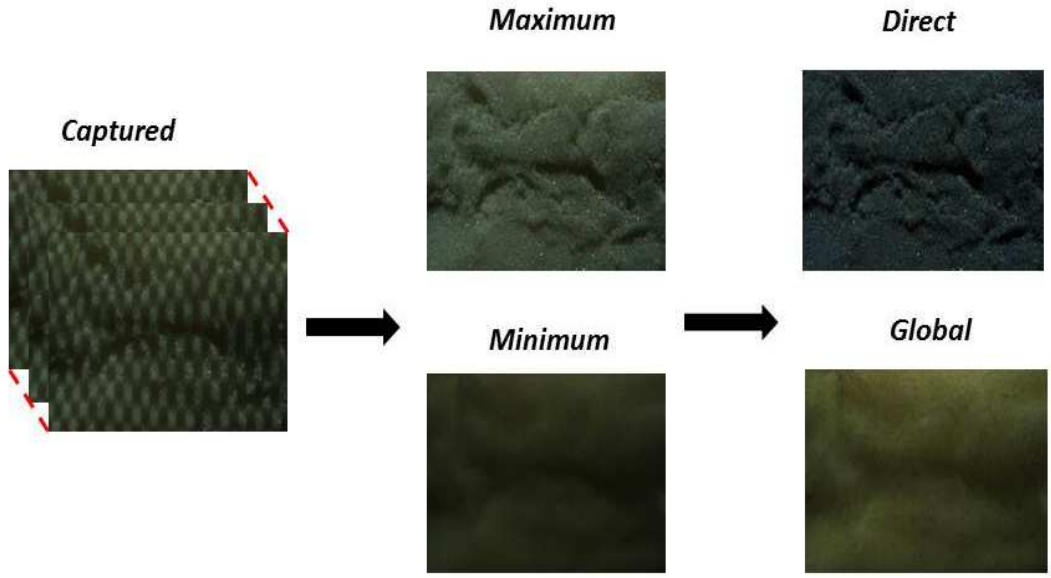


Figure 3.3 The steps involved in the computation of direct and global images using a set of shifted checkerboard illumination patterns

Modeling this behavior gives good results on all the datasets that we considered. However, it is possible that this may not hold for high-frequency textures as small shadows might appear and disappear quickly at a point with changes in lighting direction. Therefore, a denser sampling of images will always provide a better estimate. Fig 3.4 shows shadow modeling by interpolation. The luminance values of the first two types of pixels are directly calculated by linear interpolation between the values of the corresponding pixels in the given two images. One could also attempt higher order interpolation techniques, given more images of the same type. In practice, linear interpolation works well, as the variations are often very limited for the first two types. Given the luminance, L_1 and L_2 from the corresponding pixels of images taken from lighting directions p_1 and p_2, the luminance value of the interpolated pixel(k), L, at lighting position p_0 is given by:

$$L(k, p_0) = \frac{\omega_2 L_1(k, p_1) + \omega_1 L_2(k, p_2)}{\omega_1 + \omega_2}, \tag{3.1}$$

where $\omega_i = |D(p_0, p_i)|$; $D(p_a, p_b)$ gives distance between lighting directions at p_a and p_b.

In case of a pixel that transitions from shadow to light (or the reverse), the transition is quick, though not instantaneous. We model this behavior using a sigmoid function. As the light source moves from the position of the first image(p_1) to the second(p_2), there is a point p_x around which the pixel quickly emerges out of the shadow and then remains illuminated for the rest of the light motion. The transition would be abrupt except for the diffraction of light around the edge causing the shadow. Given the illuminations of the shadow (L_s) and non shadow (L_{ns}) pixels, and the position p_x at which the transition occurs, the illumination at position p_0 can be approximated by a sigmoid of the form:

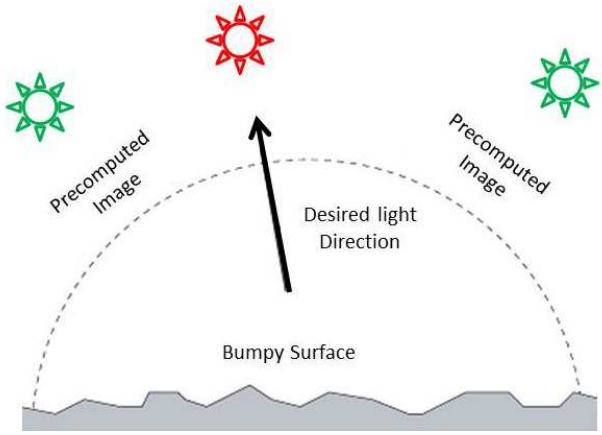
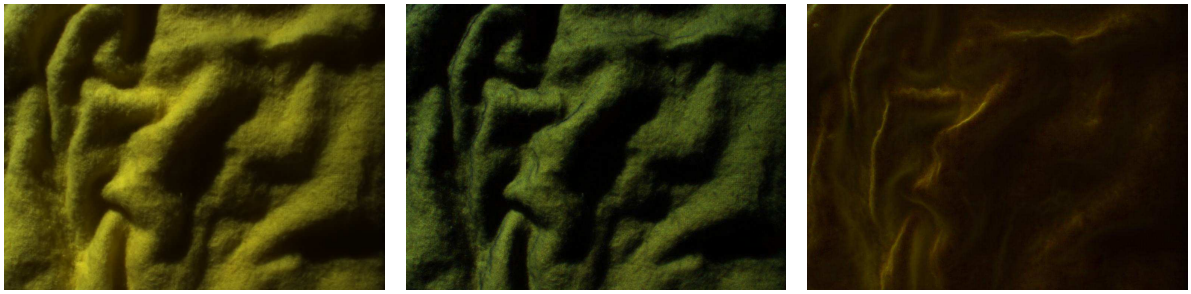


Figure 3.4 Shadow modeling by interpolation



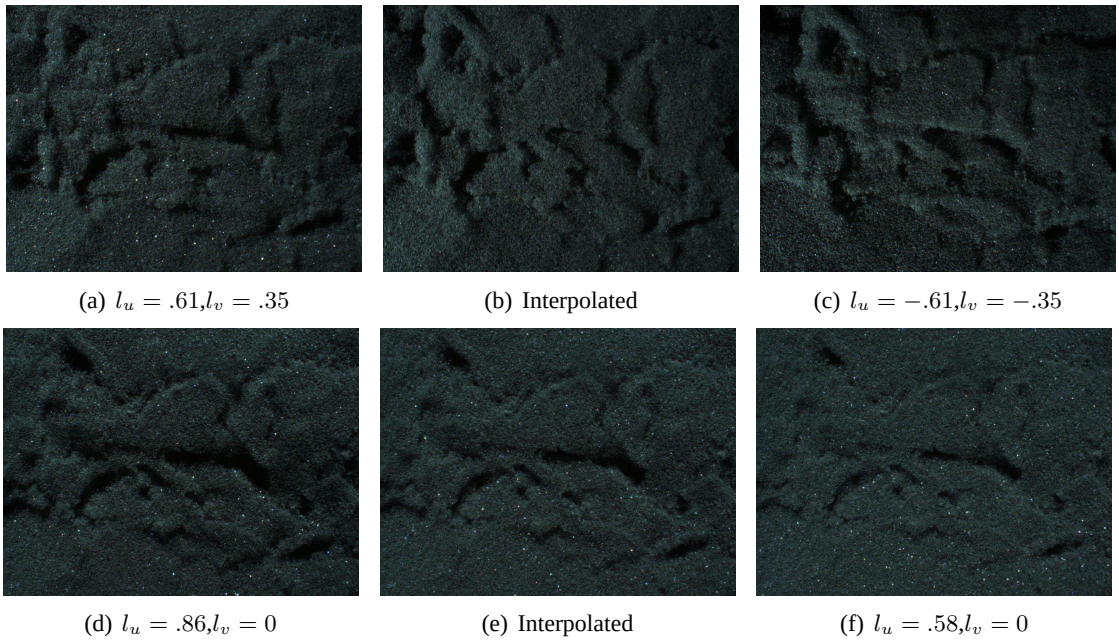
(a) (b) (c)

Figure 3.5 Components of a cloth image for a specific lighting direction: a) Original image, b) Direct component, c) Global component.

$$L(k, p_0) = L_s(k, p_1) + \frac{L_{ns}(k, p_2) - L_s(k, p_1)}{1 + \chi e^{-d}}, \tag{3.2}$$

where $d = p_0 - p_x$. The slope of the sigmoid function controls the transient behavior of pixels from shadow to non-shadow region, controlled by the parameter χ.

This sigmoid function will exhibit different behavior for different pixels. For example, the pixels that are at the edge of shadows say in the first image, will come out of the shadow quickly, whereas the pixels that are at the center of the shadow will continue to remain in shadow region for a longer time as the light source is moved from p_1 to p_2. The parameter χ varies slightly depending on the nature of the surface, but can be treated constant for all practical purposes, provided the light positions are angular measurements. The only unknown in carrying out the interpolation is the position p_x at which the transition occurs. A quick approximation may be obtained by counting the number of pixels around the pixel under consideration that are in shadow and not. Consider a pixel k that is in shadow at light position p_1. Let χ_s be the fraction of neighboring pixels of k that are in shadow in the first image, and χ_{ns} be the fraction of neighboring pixels of k that are not in shadow in the second image. We compute



(a) $l_u = .61$,$l_v = .35$ (b) Interpolated (c) $l_u = -.61$,$l_v = -.35$

(d) $l_u = .86$,$l_v = 0$ (e) Interpolated (f) $l_u = .58$,$l_v = 0$

Figure 3.6 Shadow interpolation in two directions: a,c) images with horizontally varying lighting directions, b) interpolated direct image between the two; d,f) images with vertically varying lighting directions, e) interpolated direct image between the two.

these fractions by taking masks of increasing sizes until the $0 < \chi_x < 1$. If χ_{ns} and χ_s are almost equal, then the transition, p_x occurs around midway between positions p_1 and p_2. If $\chi_{ns} >> \chi_s$, then p_x is close to p_1, and $\chi_s >> \chi_{ns}$ indicates that p_x is far from p_1 and close to p_2. We define p_x as:

$$p_x = \frac{\chi_{ns} p_1 + \chi_s p_2}{\chi_{ns} + \chi_s} \tag{3.3}$$

The advantage of interpolation is that the physical structure of the material is taken into account while interpolating, leading to realistic estimations of shadows. This is implicitly used while considering the neighborhood information of a pixel. However approach is both memory and compute intensive as one need to store input images for interpolation, and the computation of each pixel of the shadow mask involves searching an increasing neighborhood of pixels. An alternate method is to decide whether a given pixels falls in shadow or not, independently as a function of just the lighting position. Fig 3.6 shows the input images and the interpolated image at two different lighting positions.

3.2.2 Shadow Modeling by Classification

In our experiments, we note that most pixels fall under shadow from the effect of at most two neighboring structures. Hence, a biquadratic classifier boundary is adequate to decide whether for a given lighting direction, the pixel will be in shadow or not. The direct component of input images are binarized using thresholding and used as training data. After the classification, each pixel in new image is labeled as shadow or non-shadow.

(a) (b) (c)

Figure 3.7 a) Direct component of an image computed using bilinear interpolation, b) after multiplying (a) by the shadow mask, and c) after adding specularity.

$$Ya = b \tag{3.4}$$

$$\underbrace{\begin{bmatrix} {y_1}^{(0)} & {y_1}^{(1)} & \ldots\ldots & {y_1}^{(5)} \\ {y_2}^{(0)} & {y_2}^{(1)} & \ldots\ldots & {y_2}^{(5)} \\ \vdots & & & \vdots \\ {y_n}^{(0)} & {y_n}^{(1)} & \ldots\ldots & {y_n}^{(5)} \end{bmatrix}}_{Y} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_5 \end{bmatrix}}_{a} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{b} \tag{3.5}$$

where y_i=$[{l_u}^2 \;\; {l_v}^2 \;\; l_u l_v \;\; l_u \;\; l_v \;\; 1]$ and 'n' is the total number of training images.

However, the direct computation of the classifier from the input images often results in incoherence between neighboring pixels in an image. To improve this, we first interpolate the input images to obtain the shadow masks at a larger number of intermediate positions. These values are then used to train the classifier. The resulting shadow regions estimated by classifier are very close to original and more accurate than the images that are directly interpolated from the input images as shown in Fig 3.7.

Once the classifier is trained, the distance of a point classified as shadow from the decision boundary can be thought of as the distance from the point of transition from light to shadow. We use this distance to decide the darkness of a shadow pixel. The pixel which lies in the region of strong shadows will have a greater value of absolute distance from the decision boundary than the pixel which is in a region of diffused shadow or is at the edge of a shadow (See Fig 3.8).

We use this binary image to make a mask where each non-shadow pixel is given a value of 1 and shadow pixels are given values between 0 to 1 based on their distance from the hyperplane. The greater the distance, the farther the pixel is in shadow and thus smaller is the value. Now, since the direct component is devoid of color variation the change in chrominance value of direct image is very minimal. Thus using a bilinear function

$$L(l_u, l_v) = \alpha l_u + \beta l_v + \gamma, \tag{3.6}$$

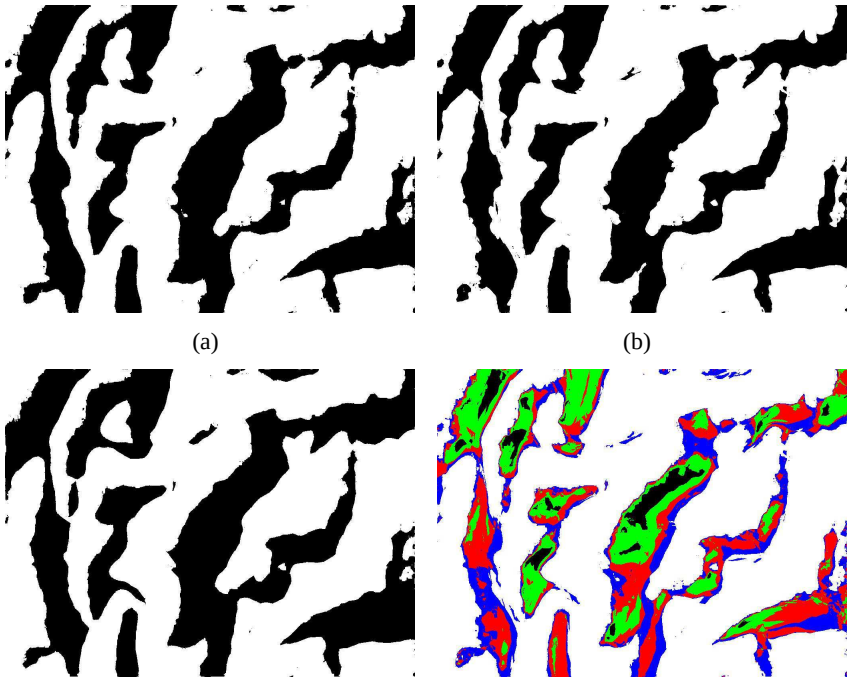


Figure 3.8 a) Binarized image of cloth shadow, b) binary image as rendered by classification technique, c) binary image obtained using interpolation, d) distance image of pixels from classifier boundary. Blue pixels are closest to the hyperplane and include pixels at the edge of a shadow or pixels present in the region of diffused shadow. Black color pixels are the farthest from the hyperplane and represent region of strong and dense shadow.

an interpolated image is generated (Fig 3.7 (a)). This image is then multiplied with shadow mask. Biquadratic function can also be used for interpolation. Using either of the function for interpolation leads to the smoothening of shadows. Therefore we multiply this image with a mask described above to get the shadowed new image (Fig 3.7 (b)). The value of non-shadow pixels are not affected but the values of the shadow pixels are attenuated by the multiplication with the shadow mask. The classification technique thus enables us to render each pixel independently, increasing the speed of rendering and making it suitable for processing on the GPUs. The pseudo code for shadow modeling by classification is given in Algorithm 1.

3.2.3 Modeling the Specularity

Specularity is the visible appearance of specular reflections. It determines the brightness and location of the specular highlights, given a lighting direction. In case of PTMs, the ability to model the specularity is sacrificed due to fixed viewing direction. PTMs use a biquadratic interpolation model due to which the intermittent highlights, inherently present in many texture surfaces, are completely washed out.

Algorithm 1 Shadow modeling by classification

Require: Binarized direct component training images

1: Take shadow pixels as negative samples and non-shadow pixels as positive samples.
2: Learn the hyper plane (Ya=b) per pixel using pseudo inverse technique.
3: Classify pixels for a given lighting direction

$$Img(i,j) = \begin{cases} 1 & \text{if } Ya > 0 \\ 0 & \text{otherwise} \end{cases}$$

Ensure: shadow mask image

We model the specular highlights separately from the base reflection and shadowing in the direct component. The value of pixels showing these highlights fall off very sharply as lighting direction is changed. One could use any sharp falling function such as a Gaussian with very small variance or an exponential to model it. We model the specular highlights, S, as:

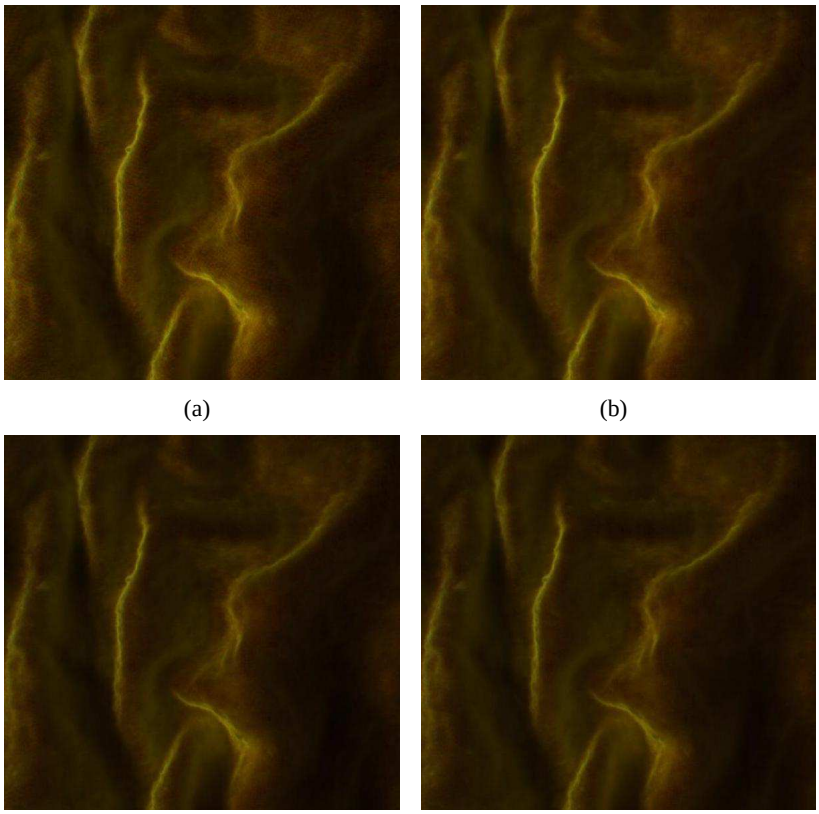
$$S = \eta exp - \left[\frac{(l_u - \mu_x)^2 + (l_v - \mu_y)^2}{\delta} \right], \tag{3.7}$$

where μ_x and μ_y are the lighting direction coordinates at which specularity is maximum, l_u and l_v are the current lighting directions. η and δ are the parameters that control the magnitude and fall-off of this function. The highlights also can have a tint based on the nature of reflection. In this case, one can multiply the above function with a single chrominance value to achieve realistic estimation.

We note that the modeling of highlights is tricky as one can observe highlights only if one of the original images contain it. Hence the highlights estimated are often inaccurate, although realistic. Fig 3.7 (c) shows the final image after multiplying the bilinearly interpolated image with shadow mask and adding specularity.

3.3 Modeling Global Component

The global component of the image is characterized by subsurface scattering, secondary illumination, diffuse inter-reflections, volumetric scattering and translucency. These are not sharply varying phenomena and therefore the variation of luminance can be modeled using appropriate function. However, the inherent interaction between different parts of the surface in global illumination means that the chrominance of a point can change with change in lighting direction. The global part of the image is responsible for the feeling of depth and life in many surfaces. As we separate the modeling of global component, the color values of the image rendered are closer in value to the original image and better than the images generated by PTM. From our experiments on various surfaces, the global component of illumination tends to be maximal when the illumination is perpendicular to the surface, and drops off in a symmetric fashion. We experiment with the following function for modeling the global component: a) Gaussian b) biquadratic polynomial, and c) paraboloid.



(a) (b)

(c) (d)

Figure 3.9 (a) Original Global Image (b) Global Image modeled by Gaussian function (c) Global Image modeled by biquadratic (d) Global image modeled by Parabola.

In Gaussian function we model the luminance as a gaussian function of lighting direction:

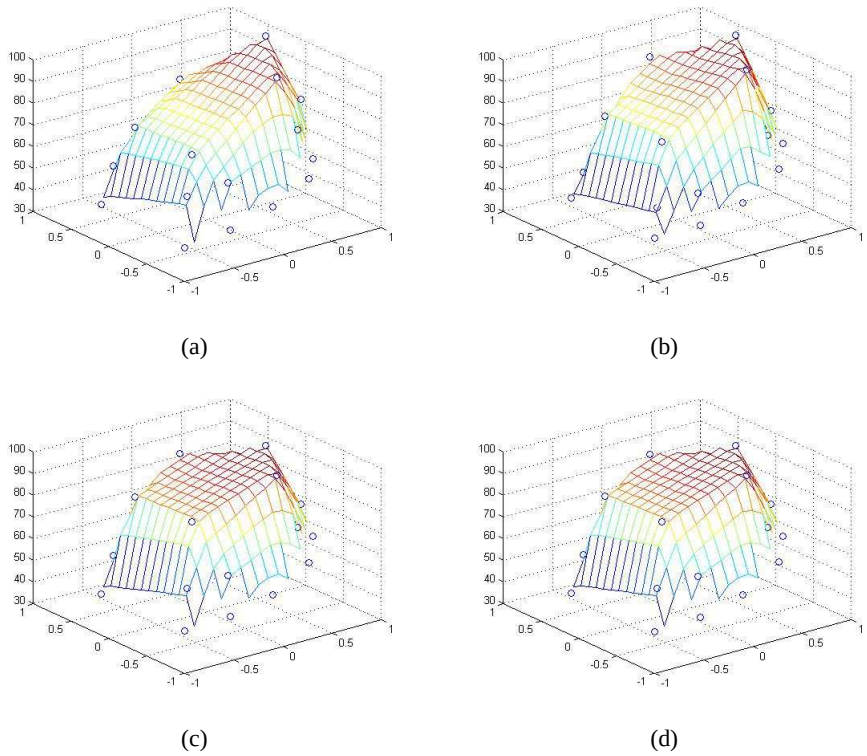
$$L(l_u, l_v) = K \exp -(al_u^2 + bl_v^2 + cl_u l_v + dl_u + el_v + f). \tag{3.8}$$

The equation may be rewritten as:

$$al_u^2 + bl_v^2 + cl_u l_v + dl_u + el_v - k = -\ln(L(l_u, l_v)), \tag{3.9}$$

resulting in the following system of linear equations for parameter estimation:

$$\begin{bmatrix} l_{u1}^2 & l_{v1}^2 & l_{u1}l_{v1} & l_{u1} & l_{v1} & -1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ l_{un}^2 & l_{vn}^2 & l_{un}l_{vn} & l_{un} & l_{vn} & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ k \end{bmatrix} = \begin{bmatrix} -\ln(L_1) \\ \dots \\ \dots \\ \dots \\ \dots \\ -\ln(L_n) \end{bmatrix}$$



(a) (b)

(c) (d)

Figure 3.10 Comparison of luminance at a pixel as modeled by different functions: a) original function plot at that pixel b) By Gaussian c) By Biquadratic d) By Parabola.

The above system of equations can be solved using SVD and the coefficients a, b, c, d, e, and k, can be estimated per pixel. Biquadratic polynomial, also used in modeling PTMs [33], can be a good choice here because of the absence of sharply varying features. The function is given by:

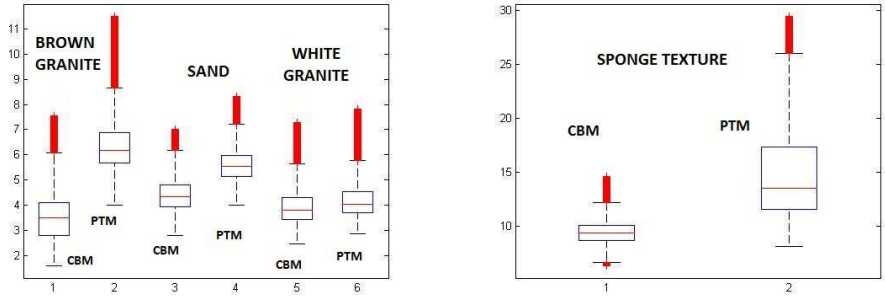
$$L(l_u, l_v) = al_u^2 + bl_v^2 + cl_u l_v + dl_u + el_v + f \tag{3.10}$$

The paraboloid may not be as accurate as above functions and can lead to some smoothening but they are computationally efficient with 5 coefficients per pixel

$$L(l_u, l_v) = al_u^2 + bl_v^2 + cl_u + dl_v + e \tag{3.11}$$

Fig 3.9 shows the global component as modeled by each of the above functions. We see that the gaussian model provides the most accurate estimation of global component although all three models are similar in performance to visual inspection. One could hence use the paraboloid for purposes of efficiency and storage.

The graphs shown in Fig 3.10(a) shows how the luminance of a given pixel varies with different lighting directions. Fig 3.10(b)-(d) shows the luminance of this pixel as a function of lighting direction when modeled using the different functions mentioned above. It is clear from the plots that gaussian is more accurate than the other two functions especially at the peak value. The mean squared error over a



(a) (b)

Figure 3.11 Error comparison between CBM and PTM over different surface textures. Red bars indicate outliers. The red line in the box is the mean and the blue lines are the 25th and 75th percentile.

sampled set of points from different surfaces is shown for comparison in table 3.1. Experimentally and visually, the Gaussian model best fits the observations.

Table 3.1 Root Mean Square Error Comparison

Dataset	Gaussian	Biquadratic	Parabolic
Sponge	2.70	3.64	3.26
Cloth	3.60	6.01	4.41
Granite	2.81	3.99	3.25
Sand	2.66	4.09	3.97

3.4 Data Acquisition

The setup required to capture input images include projector and a camera. We collect multiple images of a static object with a static camera under varying lighting conditions. The camera is mounted vertically above a table that holds the surface. Since the camera is fixed, we avoid the need for any camera caliberation. The scene is illuminated using a high frequency checkerboard pattern with the help of the projector. The projector (light source) is moved to different lighting positions for the purpose of obtaining images with different lighting directions. The distance of the projector from the scene remains fixed, and only its height and position is changed. This enables us to capture multiple images with varying light source direction from a hemispherical set of world coordinates. We capture images from 30 different lighting directions and for each lighting direction, using component separation technique described in [38], we separate the image into its global and direct components. We capture 5-6 additional images which are used as benchmark images for comparing results.

3.5 Experimental Results and Analysis

The component based modeling proposed in this paper has been applied on various natural material textures. We present qualitative and quantitative assessment on texture images as obtained from our method and that obtained from the PTM over different natural material surfaces. Figure 3.12(a)-(o) shows qualitative results where we compare the ground truth images with images obtained from our technique and with PTMs. The texture of sand when modeled using CBM technique, preserves prominent shadow regions where as these regions are significantly washed out in PTM images(Figure 3.12(a)-(c)) The sponge texture (Fig 3.12(d)-(f)) shows a very noticeable difference between the two techniques.

In the PTM image, there are no sharp shadows, the specularities are washed out and surface relief is smoothened to some extent whereas in CBM, structural details are preserved making it look more photo realistic. In Figure 3.12(g)-(i),(m)-(o) two different granite surfaces are modeled. Once again PTM smoothens sharp shadows, while they are preserved in the CBM rendered images.

For quantitative comparison, we capture additional images from known lighting directions during the capture phase. Generic measures such as PSNR only gives the average differences, and are not visually significant. We compute the absolute differences between each pixel values and analyze the distribution of these values. The differences between the original image and the image rendered using CBM and PTM are also plotted as a boxplot (see Figure 3.11).

It is clear from the the figure that the average per pixel error and the number of outlier points are less in the image rendered using component based modeling technique as compared to those rendered using PTM. However, one should note that the PTM based models miss the specularities completely, while CBM is possibly rendering some of the specularities at incorrect positions. This would result in a higher quantitative error for CBM, while the visual appearance is improved.

In case of brown granite texture (Fig 3.11a-1,2), one can observe that the number of outliers are quite high in case of PTM. This is because the texture has specularities which PTM fails to captures and also the sharpness of edges in shadow regions are lost. The root mean square error in case of CBM is 3.5 whereas in case of PTM its 6.2.

If outliers are included then rms becomes 4.9 for CBM and shoots upto 10.1 for PTM. In case of white granite (Fig 3.11a-5,6), there is not much difference in the average errors of the two techniques. For CBM rms error is 3.8 and 4.2 for PTM. As the texture does not have specularity, also there is not much structural variation and the shadow regions are small, therefore PTM is able to model it quite well with lesser errors. However, if we consider sponge texture, the PTM performs quite badly with average error of around 14 and 75th percentile at 17 whereas average error of CBM is around 8 with 75th percentile at 10(Figure 3.11(b)). Sponge is a highly textured surface with specularities and prominent shadow regions and therefore PTM produces very bad results as it tends to smoothen out the surface relief. But component based modeling accurately captures all structural details and thus rendered image is closer to the original. Modeling the shadows and specularity separately in CBM also allows us to make rendered images with multiple light sources, more realistic. Consider Figure 3.13, where the

sponge texture is illuminated at 10° (from the top of the image), and 180° (bottom). Areas that are in shadows for both lighting directions are preserved as shadow in the resultant image and the specularities have added up. However, with PTM based rendering, shadows tend to become more washed out with multiple light sources and rendered image is void of specularities.

But this improvement is achieved at the cost of more coefficients per pixel as compared to 6 coefficients in PTM. CBM uses a total of 19 coefficients as compared to [11] which requires the estimation of parameters- α (a scalar), β (3-vector) and γ (n-vector). These parameters are estimated per pixel, separately for shadow and specularity modeling. Since 'n' is generally 40-50, therefore total coefficients are very high as compared to ours.

3.6 Summary

In this chapter, we introduced a framework where separation of images into *direct* and *global* components helps us in better modeling of 3D textures. We improved upon the Polynomial Texture Maps (PTMs) which tend to smoothen the changes in appearance. First, we captured multiple images of a static scene with a static camera under varying lighting conditions. The scene is illuminated using a high frequency checkerboard pattern using the projector. We showed that the direct and global components of the image have different nature, and when modeled separately, leads to a more accurate and compact model of the 3D surface texture. Direct component is mainly affected by structural properties of the surface and deals with phenomena like shadows and specularity, which are sharply varying functions. Therefore, we used direct component for shadow and specularity modeling. On the other hand, global component is used to model overall luminance and color values, a smoothly varying function. So for a given lighting position, both components are computed separately and combined to render a new image. Thus rendered image have enhanced photorealism as compared to images rendered by existing single pixel models such as PTMs. In the next chapter, we propose a new technique which segments scene text from complex background using a component based model.

(a) Original Sand Texture (b) Modelled using CBM (c) Modelled using PTM

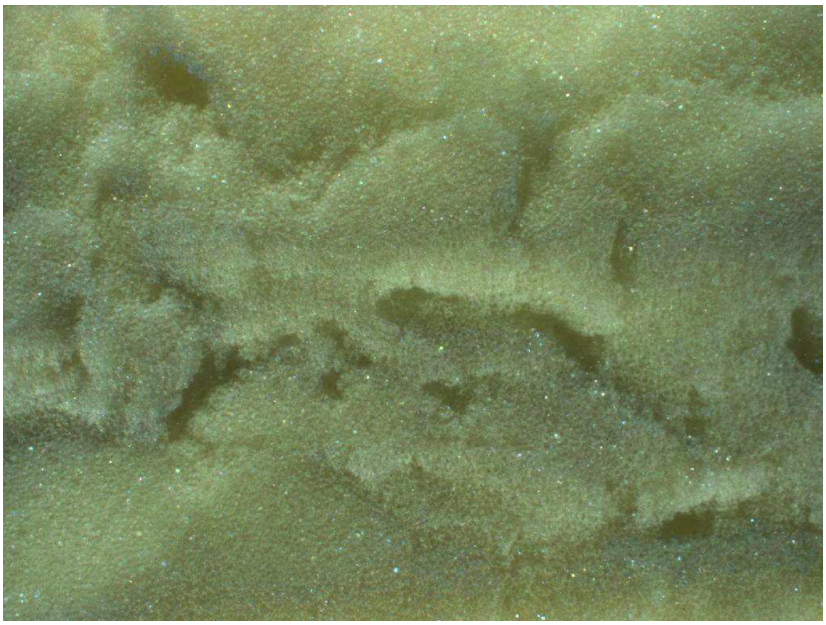
(d) Original Sponge Texture (e) Modelled using CBM (f) Modelled using PTM

(g) Brown Granite (h) Modelled using CBM (i) Modelled using PTM

(j) Original Cloth Texture (k) Modeled using CBM (l) Modeled using PTM

(m) Zoomed part of White Granite (n) Modelled using CBM (o) Modelled using PTM

Figure 3.12 Comparison of rendering results from Component Based Modeling and PTM techniques. CBM images have sharp shadows and specularity and also preserve the appearance of surface relief.



(a) CBM based output image

(b) PTM based output Image

Figure 3.13 Multiple simultaneous Light Sources effect. For (a) and (b) light sources are placed at top(10°) and bottom(180°) side of the texture.

Chapter 4

Component Based Text Segmentation

4.1 Independent Component Analysis (ICA) Model

Independent Component Analysis (ICA) has been an active research topic because of its potential applications in signal processing. And recently it has received attention in image processing tasks. In ICA model, more than one observation signals are needed to achieve the analysis. Currently when ICA is applied in image processing, images are divided into blocks [4, 19, 2] with size of 8x8 or 16x16. These blocks are taken as the observations of ICA model. But here we divide the image into sub-components i.e Red, Green and blue channels which are taken as the observation of ICA model (Fig 4.1).

The goal of ICA is to separate independent source signals from the observed signals, which is assumed to be the linear mixtures of independent source components. The mathematical model of ICA is formulated by mixture processing and an explicit decomposition processing. Assume there exists a set of 'n' unknown source signals $S = \{s_1, s_2, ..., s_n\}$. The assumptions of the components $\{s_i\}$ include mutual independence, stationary and zero mean. A set of observed signals $X = \{x_1, x_2, ..., x_n\}$, are regarded as the mixture of the source components. The most frequently considered mixing model is the linear instantaneous noise free model, which is described as:

$$x_i = \sum_{j=1}^{n} a_{ij} s_j \tag{4.1}$$

or in the matrix notation

$$X = A.S \tag{4.2}$$

where A is an unknown full rank mixing matrix, which is also called mixture matrix. Eqn 4.1 assumes that there exists a linear relationship between the sources S and the observations X. The ICA model describes how the observed data is generated by a process of mixing the components. In our case, 'n' is equal to 3.

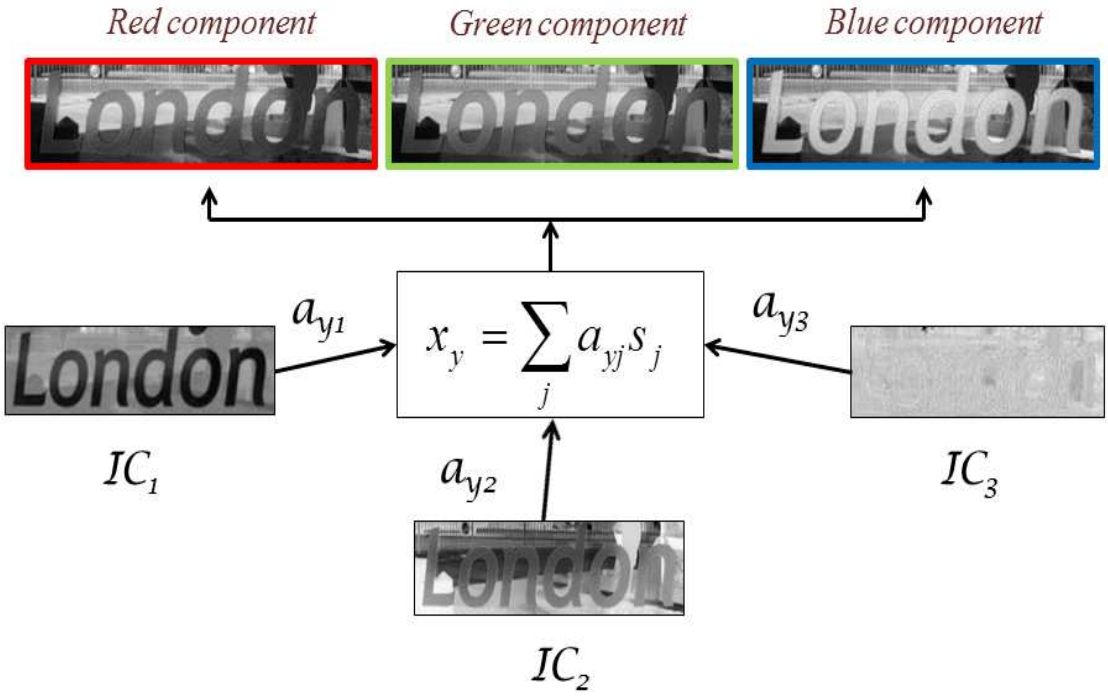


Figure 4.1 ICA Model - IC1: Independent component containing the foreground text, IC2: Independent component containing the Background, IC3: Independent component containing the mixture of foreground and background. y $\in$ {R,G,B}

4.2 Natural Scene Text Binarization

Binarization is a process to convert a grayscale or color images into black and white images. Binarization problem classifies individual pixels as foreground text or background. A pixel is grouped into two colors i.e black and white. The black pixels represent the foreground text while the white pixels represent the background and thus a binary image is created. It is a necessary step before recognition of text by OCR (Optical Character Recognition). Accurate segmentation of natural scene texts can significantly increase the success of the subsequent text recognition step. But these scene texts contain numerous degradations such as noise, uneven illumination, blur, highlights, shadows, multiple colors and complex textured background. These issues severely degrade the segmentation accuracy. Hence we propose a technique which helps is segmentation of text from complex backgrounds. Fig 4.2 shows the sample images that we considered in this work.

4.2.1 Binarization process

A wide variety of ICA algorithms are available in the literature [20, 21]. These algorithms differ from each other on the basis of the choice of objective function and selected optimization scheme. Here

Figure 4.2 Some sample word images we considered in this work containing (a) reflective (b) shadowed and (c) specular background

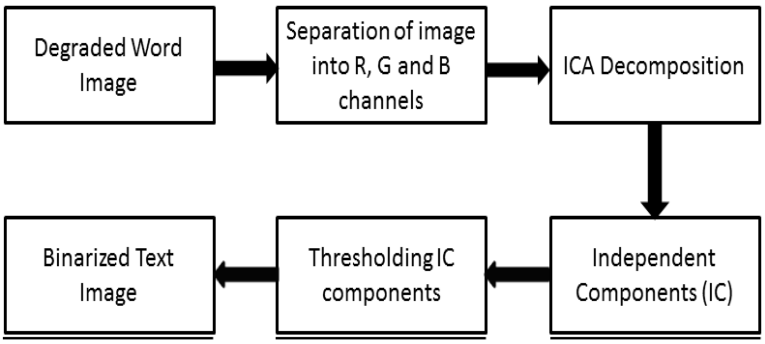


Figure 4.3 Framework for the proposed method

we use a fast fixed point ICA algorithm to separate out the text from complex background in images. A Blind Source Separation method based on SVD [61] can also be used. Fig. 4.3 shows the complete framework for the proposed method.

4.2.1.1 The Separation Model

Consider the text image as a mixture of pixels from three different sources and assume it to be a noiseless instantaneous mixture. We use a single image i.e its R, G and B channels as three observed signals. Therefore, we can define that the color intensity at each pixel from these three observed signals mix linearly to give the resultant color intensity at that pixel. Denoting these mixture images in row vector form as x_r, x_g and x_b, the linear mixing of the sources at a particular pixel k can be expressed in matrix form as follows:

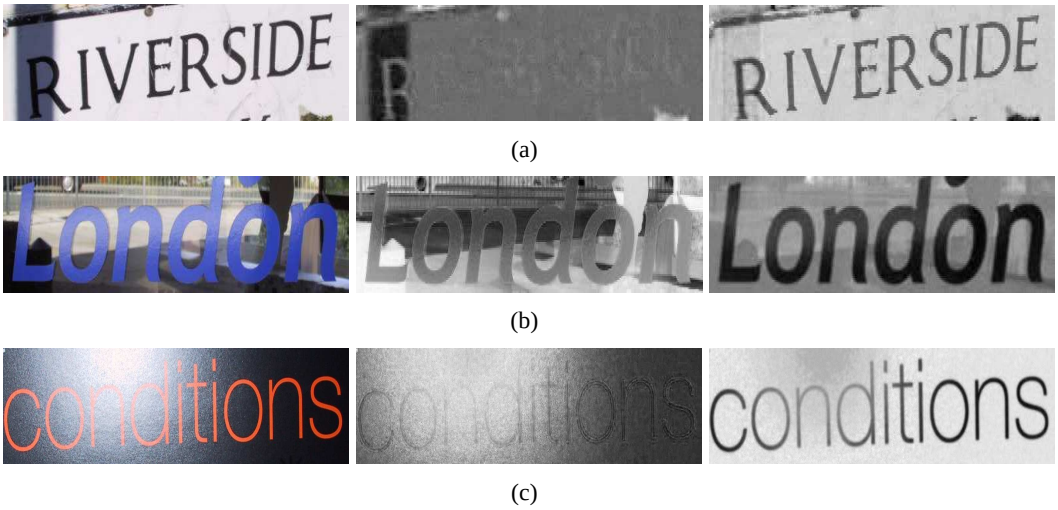


Figure 4.4 Foreground and Background Extracted: (a) Shadowed background and foreground text (b) Reflective background and foreground text (c) Specular background and foreground text

Algorithm 2 Fixed Point ICA

Require: X

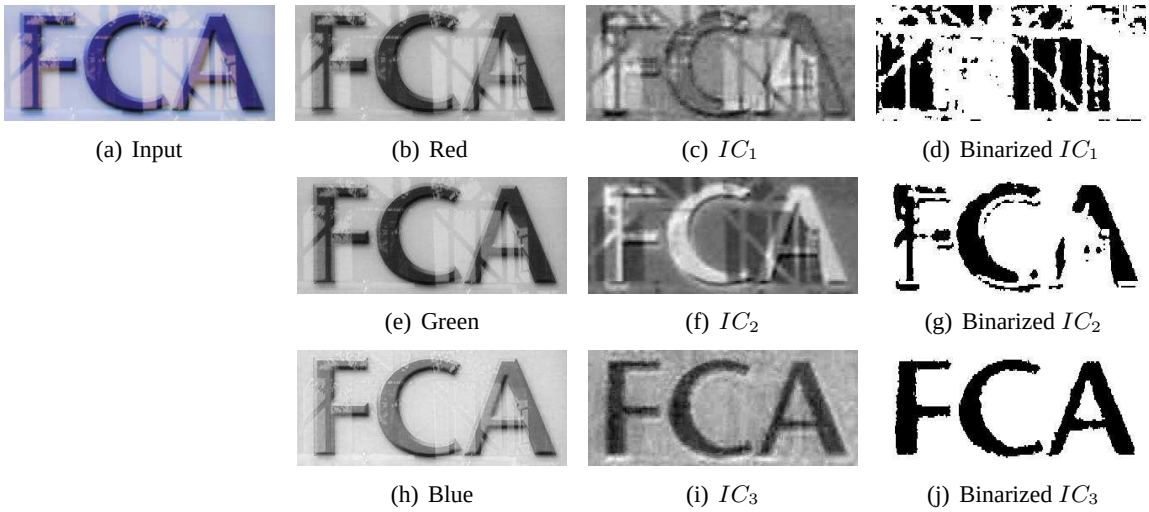
1: Random initialization of A
2: $S = A^T X$
3: $A^+ = Xg(S)^T$ where $g(x) = tanh(x)$
4: $A = A^+ / ||A^+||$
5: If not converged, go back to 2.

Ensure: A, S

$$\underbrace{\begin{bmatrix} x_r(k) \\ x_g(k) \\ x_b(k) \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} s_1(k) \\ s_2(k) \\ s_3(k) \end{bmatrix}}_{S} \tag{4.3}$$

where X is an instantaneous linear mixture of source images at pixel k, A is the instantaneous 3x3 square mixing matrix and S is the source images which add up to form the color intensity at pixel k. The mixed images in X contain a linear combination of the source images in S. We find the mixing matrix A and sources S using fixed point ICA algorithm. Derivation of the algorithm is beyond the scope of this paper. The reader is encouraged to refer [20] for this. We summarize the fixed point ICA method in Algorithm 2.

From this step, we get three independent sources or components. Fig. 4.4 shows the background and the foreground extracted. The resultant independent components for a particular word image can be seen in Fig. 4.5 which shows the independent component free from reflective background and containing maximum information of the foreground text.



(a) Input (b) Red (c) IC_1 (d) Binarized IC_1

(e) Green (f) IC_2 (g) Binarized IC_2

(h) Blue (i) IC_3 (j) Binarized IC_3

Figure 4.5 (a) Original word image (b),(e),(h) R, G and B channel respectively (c),(f),(i) Independent Components, (d),(g),(j) Binarized image

(a) (b)

(c) (d)

Figure 4.6 (a) Text containing specular highlight (b) IC (c) Otsu (d) Niblack

4.2.1.2 Thresholding

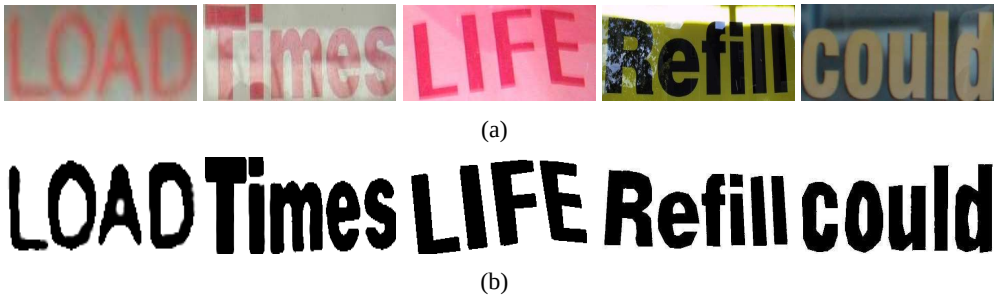
Otsu thresholding [45] is a well-known algorithm that determines a global threshold for an image by minimizing the within-class variance for the resulting classes (foreground pixels and background pixels). This is done by equivalently maximizing the between-class variance $\sigma_B^2(T)$ for a given threshold T:

$$\sigma_B^2 = \alpha_1(T)\alpha_2(T)[\mu_1(T) - \mu_2(T)]^2 \tag{4.4}$$

where α_i denotes the number of pixels in each class, μ_i denotes the mean of each class, and T is the value of the potential threshold. We apply this thresholding algorithm on all the three independent components to get the binarized image (Fig. 4.5). We can also apply Kittler [25] algorithm which is also a global thresholding method.

To find the IC that contains the foreground text, we examine the connected components (CC) in the binarization of each IC. For each binarized image, we extract the following features from the CCs:

Figure 4.7 Failure cases for thresholding based methods. From left to right (a) Text Image (b) kittler (c) Niblack (d) Otsu (e) Sauvola



(a)

(b)

Figure 4.8 (a) Scene Text Image (b) Ground Truth Binary Image

average aspect ratio, variance of CC size, and the deviation from linearity of their centroids. A simple linear classifier is designed to separate the text and non-text classes in the above feature space. After binarization, we identify the connected components and remove non-text portions based on size and aspect ratio.

In some cases where the text image is severely degraded and contain different colored text, adaptive thresholding methods work better and produce good results. As shown in Fig. 4.6, adaptive thresholding method may perform better than global one. However, in practice we note that a simpler global thresholding scheme works well in most cases.

4.2.2 Experimental Results and Analysis

We used the ICDAR 2003 Robust Word Recognition Dataset [1] for our experiments. The dataset contains a set of JPEG images of single words (Sample (171 words), TrialTrain (1157 words) and TrialTest (1111 words)). For qualitative evaluation, we selected the word images that had complex

reflective, shadowed and specular background. We separate these word images into Red, Green and Blue channels assuming that these are the mixture images of the independent source images that contains the foreground (text) and background. These three images are used for extracting the foreground as described before.

Table 4.1 Quantitative Results (Average)

Method	Precision	Recall	F-score
Otsu [45]	.68	.75	69.17
Sauvola [56]	.63	.81	66.94
Kittler [25]	.66	.76	64.33
Niblack [42]	.70	.76	71.32
MRF [35]	.79	.86	80.38
Proposed	.86	.83	83.60

Table 4.2 OCR Accuracy (%)

Method	Word Accuracy
MRF [35]	43.2
Proposed	61.6

We compare the performance of our method with four well known thresholding algorithms i.e Kittler [25], Otsu [45], Niblack [42] and Sauvola [56]. In the presence of complex background, the segmentation accuracy of the thresholding based methods decreases (Fig 4.7). We also compare with the recent method by Mishra *et al* [35]. It although performs well for many images but severely fails in cases of shadows, high illumination variations in the image. This poor show is likely due to fact that performance of the algorithm heavily depends on initial seeds. We show both qualitative and quantitative results of the proposed method. We took around 50 images from the dataset and generated its ground truth images for pixel level accuracy. Some of these images are shown in Fig 4.8. We use well known measures like precision, recall and F-score (Fig 4.9) to compare the proposed method with different binarization methods (Table I). We also use OCR accuracy to show the effectiveness of our method. Note that we are only using the subset of images that are most degraded by shadowing, illumination variations, noise and specular reflections. The results of thresholding schemes are too poor for the OCR algorithm to give any output. Therefore we only compare with the recent MRF [35] based model as shown in Table II. Some OCR results are shown in Fig 4.10.

The results show that the proposed method is an effective method and performs better than other methods in the case where images have complex background. The qualitative results are shown in Fig. 4.11. Fig 4.12 shows that our technique can also be applied to images containing text over another text. We analyze that the above methods do not work in the case where there is a complex and textured

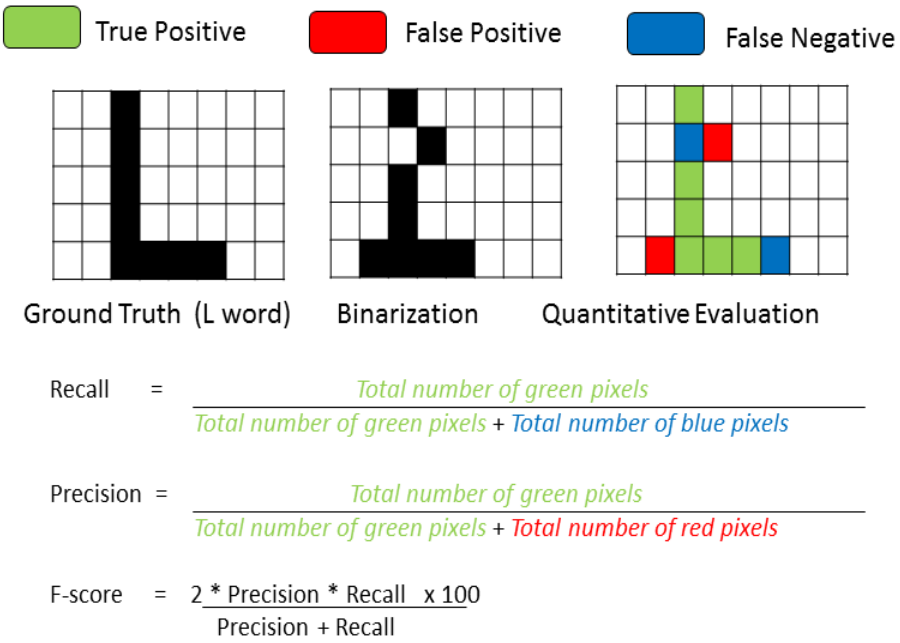


Figure 4.9 Pixel Grid showing precision and recall in a word image

Figure 4.10 OCR results on scene text images

Figure 4.11 Comparison of Binarization algorithms and the proposed method (From left to right Original, MRF, Proposed)

(a) (b)

(c) (d)

Figure 4.12 (a) Image containing Text over another Text (b) Foreground Text (c) Background Text (d) Text extracted

(a)

(b)

Figure 4.13 Failure case where (a) Both the foreground and background are of same color (b) Different Colored Text

background in the images. It is not that these methods do not work at all. No single algorithm works well for all types of images. Thus we can say that our method can extract out the text embedded in complex reflective, shadowed and specular background. The failure case of our method is shown in Fig. 4.13. Our method fails in cases where foreground text and the background are of the same color and cases where text is composed of different colored letters. Moreover, the approach works only with color images.

4.3 Applications

4.3.1 Inscribed Text Segmentation

Inscribed text is difficult to extract from one image as both the foreground and the background is of same color. The inscriptions are generally found engraved/carved into stones, marble, metal or wood. Extracted of the text is important as we want to preserve our historical writings which are currently being decayed. However, due to effects of uncontrolled lighting condition and degradation of the material of

Figure 4.14 Inscribed Text image where both background and foreground are of same color

(a) (b) (c)

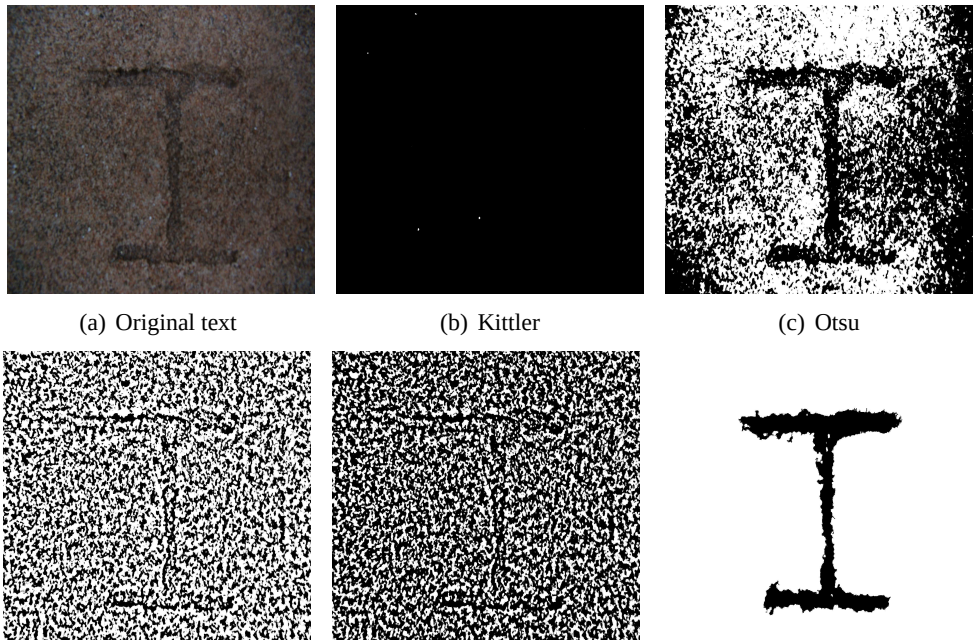
Figure 4.15 (a) Sponge Texture (b),(c) Independent components

the inscribed text, extracting text from these images has become challenging problem. Fig 4.14 shows the inscribed text images. Hence we propose an efficient technique which can solve this problem.

(a) (b) (c)

Figure 4.16 (a) Image containing text (b),(c) Independent components

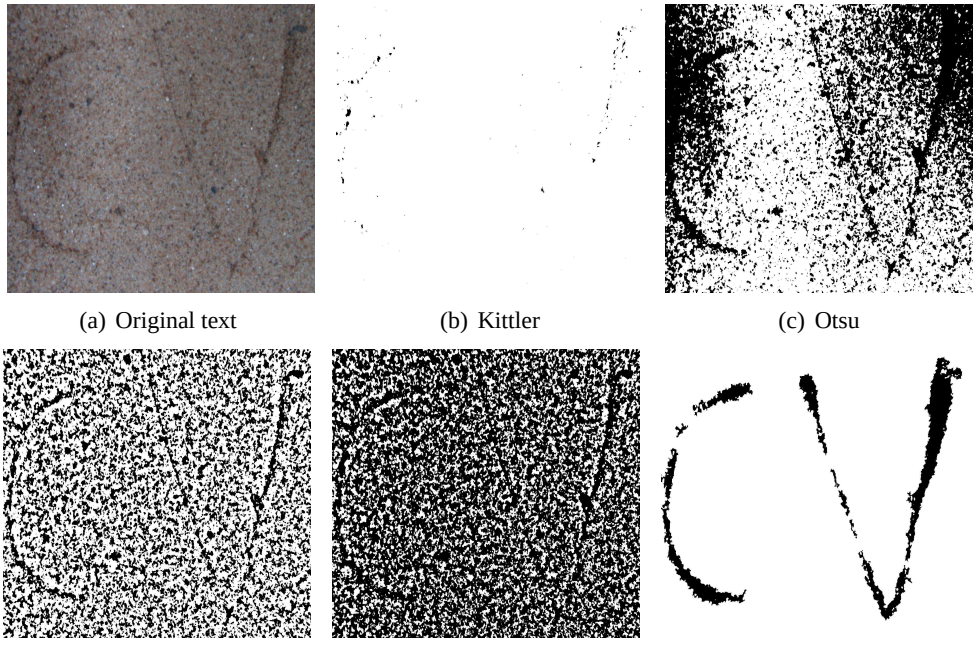
First we capture multiple images of the inscribed text and apply our model to extract the text with the help of shadows. For separating the global and the direct component, we used high frequency checkerboard pattern. This takes too much time as we have to capture many images.(Figure 4.15 shows the independent component of the sponge texture) But for this, we apply an Independent Component Analysis (ICA) based method to the images captured containing text. This method helps in extracting



(a) Original text (b) Kittler (c) Otsu

(d) Niblack (e) Sauvola (f) ICA + CBM

Figure 4.17 Binarized text



(a) Original text (b) Kittler (c) Otsu

(d) Niblack (e) Sauvola (f) ICA + CBM

Figure 4.18 Binarized text

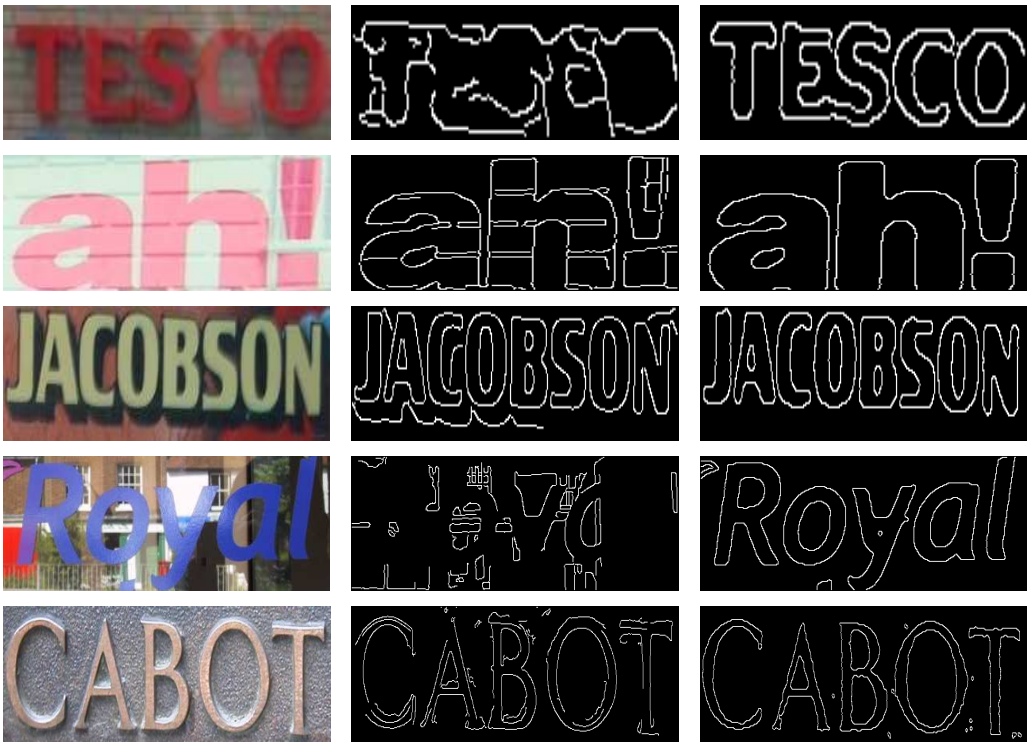


Figure 4.19 From left to right: Text image, Applying canny edge detector, Applying ICA model + canny edge detector

out the shadows (Figure 4.16). Then we apply our component based model to efficiently binarize the text embedded (Figure 4.17, 4.18).

4.3.2 Enhancing Edge Detection

Edge/Boundary detection is a fundamental task in computer vision with its applications in areas such as image segmentation, object recognition, object description, feature detection and extraction. It refers to the process of identifying points in an image where there are sharp discontinuities or the image brightness changes sharply. These points are ordered into a set of curved line segments known as edges. The abrupt changes in pixel intensity values characterizes boundaries of objects in a scene. The amount of data to be processed significantly reduces when we apply an edge detection algorithm. This helps in filtering out data which is less important and still preserving the structural properties of the image. The task of interpreting the important information content in the original image may be simplified if the edge detection step is successful. However, due to complexities in real scene images, it is not always possible to obtain clear edges which define the boundary of the object. Edge detection process is particularly sensitive to noise and uneven illumination conditions. There are many edge detection algorithms available. Each are sensitive to certain types of edges. Many models [59, 48] use edge detection as the first step to extract text from scene images. But the performance of this step

Figure 4.20 Scene images containing shadows and their corresponding shadow masks

decreases when complex background is present in the image. Here we will show how we use our ICA model to correctly identify the word boundaries. First we use canny edge detector [7] to detect edges from natural scene text images. When we directly apply canny edge detector on scene text images, we observe discontinuity in word boundaries. But when we use our ICA model followed the canny edge detector, the edge boundaries becomes much more cleaner. Fig 4.19 shows the effectiveness of our approach.

4.3.3 Shadow Detection

Shadows are generally created when objects obscure the light source. Detected shadows provide information about lighting direction and scene geometry. Due to the presence of shadows many computer vision tasks such as object detection, segmentation, scene analysis, tracking, etc are affected. For example, the shadows may assign false segments in the image segmentation process. They may be wrongly detected as objects in object detection algorithms. For these reasons, shadow detection is an important component in scene interpretation. Hence shadow detection and removal is an important preprocessing step for improving the performance of such vision tasks. But decomposition of the image into a shadow image and a non-shadow image is a difficult problem due to complex interactions of geometry and illumination. Various pixel-based and region-based methods are proposed to detect the shadows in an image [16, 15, 27, 37]. Here we show that our ICA model can be used to detect shadows and create shadow masks from the scene images. Fig 4.20 shows the shadow detected results. We first apply our ICA model followed by global binarization method. Our method does not give good results if some dark objects are present in the scene images which are mistaken as shadows.

4.4 Summary

In this chapter, we discussed the workflow of natural scene text segmentation. We saw that that text segmentation from scene images is a challenging task due to the variations in color, size, and font of the text. The results are often affected by complex backgrounds, different lighting conditions, shadows and reflections. We used Independent Component Analysis (ICA) model to map out the text region, which is inherently uniform in nature, while removing shadows, specularity and reflections, which were included in the background. We first decomposed the images into sub-components i.e Red, Green and Blue channels which were taken as the observations of ICA model. This enabled us to separate text from complex backgrounds. Then to get a clean binary image, we applied a global thresholding method on the independent components of the image and that with maximum textual properties was used for extracting the foreground text. Binarization results on ICDAR dataset showed significant improvement in the extraction of text over other previously reported methods. We also showed how our method can be used for different applications like inscribed text segmentation, enhancing edge detection and for shadow detection in scene images.

Chapter 5

Conclusion

In this thesis, we presented component based modeling in scene images. The decomposition of texture image into the direct and the global components preserves sharpness of shadows and also models specular reflection. This causes image to appear more photorealistic and single point source effect is more prominent. This technique results in enhanced photorealism which preserves sharp shadows and specular properties from smoothening out. The separate model for luminance estimation provides us with color values which are in close agreement with the color values of the original image. The advantage of the technique is that photorealism can be achieved without accurate modeling of complex real-world physical interactions. We capture lighting effects directly as they appear in reality. As they are captured in images, complex interactions like self-shadowing, inter-reflections and sub-surface scattering can be reproduced automatically. It also does not depend on the complexity of the scene and the surface properties of objects in the scene. It depends on the number of images captured or on the representation rather than the complexity of the scene. Results obtained on re-rendering the input images show a great improvement over original PTM technique. Applying this technique over the inscribed text also helps us to extract out text from degraded textured surfaces.

ICA decomposition when applied on natural scene images helps us to separate the foreground(text) from the complex/textured background. It is an effective method to binarize text from colored scene text images containing reflective, shadowed and specular background. By using a blind source separation technique followed by global thresholding, we are able to clearly separate the text portion of the image from the background. It enables us to separate reflections, shadows and specularities from natural scene texts so that the global thresholding methods can be applied afterwards to binarize the text image. Experimental results on ICDAR dataset demonstrate the superiority of our method over other existing methods. Possible directions for improvement of the approach includes a patch-based SVM classification for thresholding as well as integration of the results with a spatially aware optimization such as MRF. Working with text where the foreground and background have same color is also of great interest.

Related Publications

- Siddharth Kherada, Prateek Pandey and Anoop M. Namboodiri, “Improving Realism of 3D Texture using Component Based Modeling”, in Proceedings of IEEE Workshop on the Applications of Computer Vision (WACV) 2012.
- Siddharth Kherada and Anoop M. Namboodiri, “An ICA based Approach for Complex Color Scene Text Binarization”, in Asian Conference on Pattern Recognition (ACPR) 2013.

Bibliography

[1] Robust word recognition dataset. *http://algoval.essex.ac.uk/icdar/RobustWord.html*.

[2] J. Artur and A. Mrio. On the use of independent component analysis for image compression. *Signal Processing: Image Communication*, pages 378–389.

[3] M. Ashikhmin and P. Shirley. Steerable illumination textures. *ACM Transactions on Graphics*, 21:1–19, 2002.

[4] M. Bartlett, J. Movellan, and T. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, pages 1450–1464.

[5] A. Baumberg. Blending images for texturing 3d models. *In proc. Conf. on British Machine Vision Association*, pages 404–413, 2002.

[6] J. Bernsen. Dynamic thresholding of gray level images. *International Conference on Pattern Recognition (ICPR)*, pages 1251–1255, 1986.

[7] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 679–714, 1986.

[8] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 366–373, 2004.

[9] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. Wu, and A. Ng. Text detection and character recognition in scene images with unsupervised feature learning. *International Conference on Document Analysis and Recognition (ICDAR)*, pages 440–445, 2011.

[10] K. Dana, B. V. Ginneken, S. Nayar, and J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, 1999.

[11] M. Drew, N. Hajari, Y. Hel-Or, and T. Malzbender. Specularity and shadow interpolation via robust polynomial texture maps. *In Proceedings of the British Machine Vision Conference*, 2009.

[12] G. Earl, K. Martinez, and T. Malzbender. Archaeological applications of polynomial texture mapping: Analysis, conservation and representation. *Journal of Archaeological Science*, pages 2040–2050, 2010.

[13] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2963–2970, 2010.

[14] J. Feild and E. L. Miller. Improving open-vocabulary scene text recognition. *In Proceedings of the International Conference on Document Analysis and Recognition*, 2013.

[15] G. D. Finlayson, M. S. Drew, and C. Lu. Entropy minimization for shadow removal. *IJCV*, pages 35–57, 2009.

[16] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew. On the removal of shadows from images. *PAMI*, pages 59–68, 2006.

[17] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics and Image Processing*, 29:100–132, 1985.

[18] X. He, K. Torrance, F. Sillion, and D. Greenberg. Comprehensive physical model for light reflection. *Computer Graphics (SIGGRAPH 91 Proceedings)*, pages 175–186, 1991.

[19] J. Hurri, A. Hyvarinen, J. Karhunen, and E. Oja. Image feature extraction using independent component analysis. *IEEE Nordic Signal Processing Symposium*, 1996.

[20] A. Hyvarinen, J. Karhunen, and E. Oja. Independent component analysis. *John Wiley and Sons, New York,*, 2001.

[21] A. Hyvarinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13:411–430, 2001.

[22] C. Jiang and M. Ward. Shadow identification. *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 606–612, 1992.

[23] C. Jiang and M. Ward. Shadow segmentation and classification in a constrained environment. *In CVGIP: Image Understanding*, pages 213–225, 1994.

[24] K. Kim, H. Byun, Y. Song, Y. Choi, S. Chi, K. Kim, and Y. Chung. Scene text extraction in natural scene images using hierarchical feature combining and verification. *International Conference on Pattern Recognition (ICPR)*, pages 679–682, 2004.

[25] J. Kittler, J. Illingworth, and J. Foglein. Threshold selection based on a simple image statistic. *Computer Vision, Graphics, and Image Processing*, 30:125–147, 1985.

[26] E. Lafortune, S. Foo, K. Torrance, and D. Greenberg. Non-linear approximation of reflectance functions. *Computer Graphics (SIGGRAPH 97 Proceedings)*, pages 117–126, 1997.

[27] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Detecting ground shadows in outdoor consumer photographs. *European Conference on Computer Vision (ECCV)*, 2010.

[28] J. Lee, P. Lee, S. Lee, A. Yuille, and C. Koch. Adaboost for text detection in natural scene. *International Conference on Document Analysis and Recognition (ICDAR)*, pages 429–434, 2011.

[29] A. Leone, C. Distante, and F. Buccolieri. A texture based approach for shadow detection. *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 371–376, 2005.

[30] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal On Computer Vision*, 43(1):29–44, 2001.

[31] C. Li, X. Ding, and Y. Wu. Automatic text location in natural scene images. *International Conference on Document Analysis and Recognition*, pages 1069–1073, 2001.

[32] R. Lienhart and A. Wernicke. Localizing and segmenting text in images and videos. *Circuits and Systems for Video Technology*, pages 256–268, 2002.

[33] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. *In Computer Graphics, SIGGRAPH Proceedings*, pages 519–528, 2001.

[34] S. Milyaev, O. Barinova, T. Novikova, P. Kohli, and V. Lempitsky. Image binarization for end-to-end text understanding in natural images. *IEEE Proceedings of International Conference on Document Analysis and Recognition*, pages 128–132, 2013.

[35] A. Mishra, K. Alahari, and C. Jawahar. An mrf model for binarization of natural scene text. *Proceedings of International Conference on Document Analysis and Recognition*, 2011.

[36] A. Mishra, K. Alahari, and C. V. Jawahar. Top-down and bottom-up cues for scene text recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2687–2694, 2012.

[37] S. G. Narasimhan, V. Ramesh, and S. K. Nayar. A class of photometric invariants: Separating material from shape and illumination. *IEEE International Conference on Computer Vision (ICCV)*, 2003.

[38] S. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar. Fast separation of direct and global components of a scene using high frequency illumination. *ACM Trans. on Graphics (also Proc. of ACM SIGGRAPH)*, July 2006.

[39] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. *ACCV*, pages 2067–2078, 2010.

[40] L. Neumann and J. Matas. Real-time scene text localization and recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3538–3545, 2012.

[41] L. Neumann and J. Matas. Scene text localization and recognition with oriented stroke detection. *IEEE International Conference on Computer Vision (ICCV)*, pages 97–104, 2013.

[42] W. Niblack. An introduction to digital image processing. *New York: Prentice Hall*, 1986.

[43] F. Nicodemus, J. Richmond, and J. Hsai. Geometrical considerations and nomenclature for reflectance. *U.S. Dept. of Commerce, National Bureau of Standards*, 1977.

[44] J. Ohya, A. Shio, and S. Akamatsu. Recognizing characters in scene images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 214–220, 1994.

[45] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Systems, Man, and Cybernetics Society*, 9:62–66, 1979.

[46] J. Padfield, D. Saunders, and T. Malzbender. Polynomial texture mapping: A new tool for examining the surface of paintings. *ICOM Committee for Conservation*, 1:504–510, 2005.

[47] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26:1227–1249, 1993.

[48] W. Pan, T. Bui, and C. Suen. Text detection from scene images using sparse representation. *International Conference on Pattern Recognition (ICPR)*, 2008.

[49] Y. Pan, X. Hou, and C. Liu. A robust system to detect and localize texts in natural scene images. *Document Analysis Systems, DAS*, pages 35–42, 2008.

[50] Y. Pan, X. Hou, and C. Liu. Text localization in natural scene images based on conditional random field. *International Conference on Document Analysis and Recognition (ICDAR)*, pages 6–10, 2009.

[51] T. Phong. Illumination for computer generated images. *Communications of the ACM*, pages 311–317, 1975.

[52] C. Rocchini, P. Cignoni, C. Montani, and R. Scopigno. Multiple textures stitching and blending on 3d objects. *In Eurographics Rendering Workshop*, pages 119–130, 1999.

[53] P. Sahoo and G. Arora. A thresholding method based on two-dimensional renyis entropy. *Pattern Recognition*, 37:1149–1161, 2004.

[54] P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. C. Chen. A survey of thresholding techniques. *Computer Vision, Graphics and Image Processing*, 41:233–260, 1988.

[55] E. Salvador, A. Cavallaro, and T. Ebrahimi. Shadow identification and classification using invariant colour models. *In IEEE International Conference on Acoustic, Speech and Signal Processing*, pages 1545–1548, 2001.

[56] J. J. Sauvola and M. Pietikainen. Adaptive document image binarization. *Pattern Recognition*, 33:225–236, 2000.

[57] F. Sillion, J. Arvo, S. Westin, and D. Greenberg. A global illumination solution for general reflectance distributions. *Computer Graphics (SIGGRAPH 91 Proceedings)*, pages 187–196, 1991.

[58] M. Soucy, G. Godin, R. Baribeau, F. Blais, and M. Rioux. Sensons and algorithms for the construction of digital 3d colour models of real objects. *Image Processing Proceedings*, pages 409–412, 1996.

[59] A. Srivastav and J. Kumar. Text detection in scene images using stroke width and nearest-neighbor constraints. *IEEE Region 10 Conference on TENCON*, pages 1–5, 2008.

[60] P. Stathis, E. Kavallieratou, and N. Papamarkos. An evaluation technique for binarization algorithms. *J. UCS*, pages 3011–3030, 2008.

[61] R. Szupiluk and A. Cichocki. Blind signal separation using second order statistics. *Proc. of SPETO*, pages 485–488, 2001.

[62] C. Thillou and B. Gosselin. Color binarization for complex camera-based images. *Electronic Imaging Conference of the International Society for Optical Imaging*, 2005.

[63] T. Wakahara and K. Kita. Binarization of color characters in scene images using k-means clustering and support vector machines. *International Conference on Pattern Recognition (ICPR)*, pages 3183–3186, 2010.

[64] A. Woo, P. Poulin, and A. Fournier. A survey of shadow algorithms. *IEEE Computer Graphics and Applications*, November 1990.

[65] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1083–1090, 2012.

[66] Y. Zhou, J. Feild, E. Miller, and R. Wang. Scene text segmentation via inverse rendering. *In Proceedings of the International Conference on Document Analysis and Recognition*, 2013.