Component Based Modeling and Analysis of Texture in Scene 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 " Component Based Modeling and Analysis of Texture in Scene 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, 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 wasnt acknowledged personally above.

Abstract

Separation of images into components has been a widely used technique in the field of image processing and computer vision. Many problems are solved by partitioning images into components. The simplest example is the breaking down of image into Red, Green and Blue channels/components. The aim of this thesis is to use the components of the image in better modeling of textures and text in scene images. We first introduce a framework where separation of images into direct and global components helps in modeling of 3D textures. 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 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 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 show segmenation of natural scene text using the RGB components of the images. This 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) model 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 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

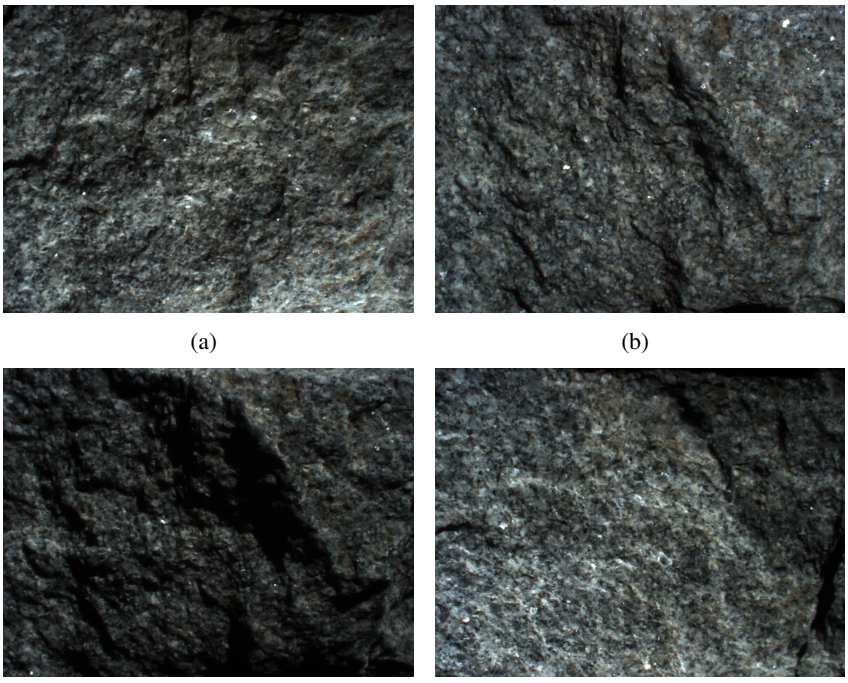
Texture refers to a surface characteristic and appearance of an object given by its geometry, density and surface reflectance, and the stochastic variation of these parameters. It is a detailed pattern that is mapped into a multidimensional space. It is an important cue in trying to achieve photorealistic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

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

Mapping 2D textures or images 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.1.1. 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. [11] 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 textures. These maps are generated using image re-lighting techniques [2, 13, 5] in which multiple images are captured under different lighting conditions.

Image based modeling techniques [3, 19, 23] have emerged as an effective approach for realistic rendering of 3D objects, where multi-view geometry is utilized in directly synthesizing an unseen view



(a) (b)

(c) (d)

Figure 1.1 Variation in appearance of the same surface patch, when illuminated from different lighting directions.

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 PTM capture the surface properties more faithfully, including the effect of small scale height variation on the surface.

Polynomial Texture Maps [13] 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). It uses a biquadratic polynomial function with 6 coefficients per pixel for modeling the reflectance. 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 [6]. It also offers advantages over traditional raking light photography for examining and documenting the surface texture and shape of paintings [17]. 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.

In this thesis, we propose an approach to image-based lighting interpolation that is based on estimates of geometry and shading from a set of input images.

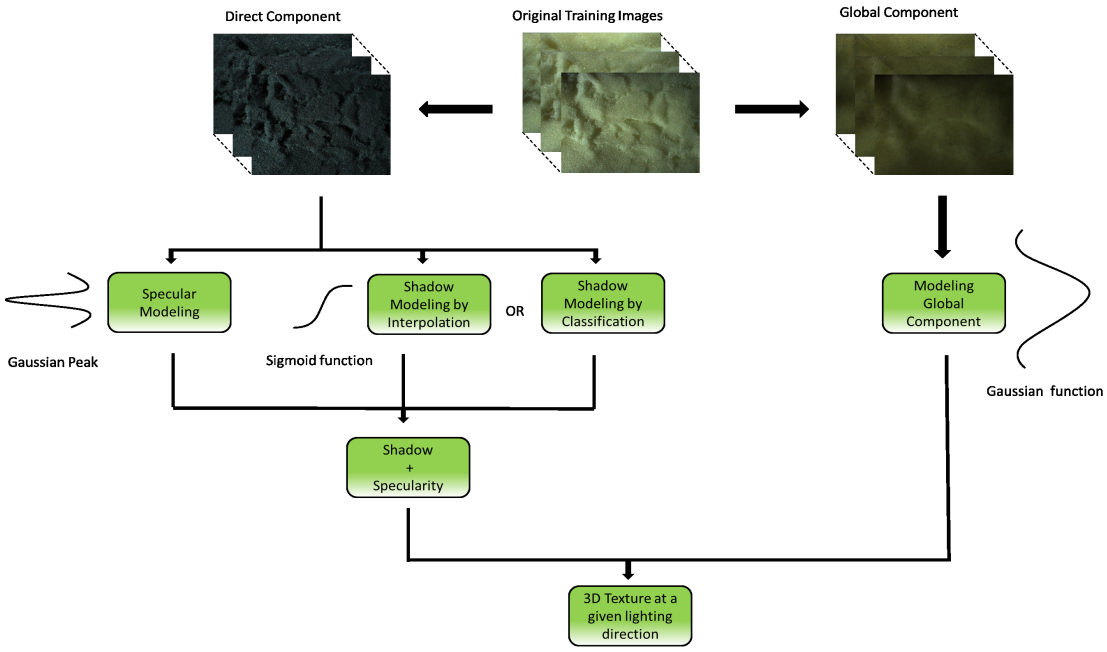


Figure 1.2 Component Based Modeling (CBM)

1.1 Problem

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. 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.

1.2 Motivation

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.

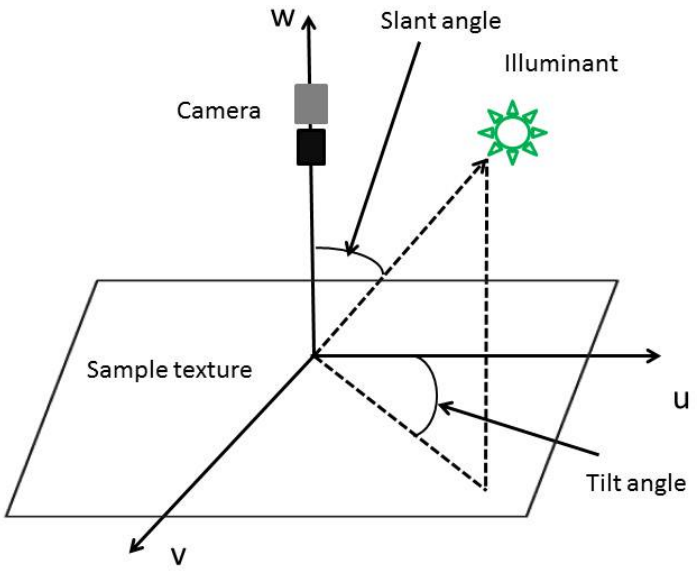


Figure 1.3 Experimental setup

1.3 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 projector is moved to different lighting positions for the purpose of obtaining images with different lighting directions. Experimental setup is shown in Figure 1.3(a)

When 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 global component contains the 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. The method is shown to indeed generate better results for non-observed lighting directions.

1.4 Contribution

The main contributions are (1) Direct and Global modeling characterized by shadows,specularity and luminance, (2) separate modeling and hence better capturing of shadows and specularities and (3) per pixel function model to achieve real-time rendering of enhanced 3D textures on GPU.

1.5 Outline

The rest of this thesis is organized as follows: We will survey techniques related to texture mapping in Chapter 2. In Chapter 3 component based modeling will be discussed where we will show how each component is modeled differently. we will explain shadow and specularity modeling In Chapter 4, we apply our model for extracting text inscribed in walls. Finally we conclude the thesis and discuss future work in Chapter 5.

Chapter 2

Background and Related Work

a)texture in scene images texture 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.

There are several texture mapping methods that avoids modeling of the complex surface details. 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 capture a high level detail of object properties. Generally an image is used as a texture map on a planar surface. However this method fails if the lighting conditions of the synthetic environment are different from the lighting conditions of the texture image. To solve this problem, people have used image based modeling and rendering (IBMR) techniques. In these methods several images of the same object are taken under varying lighting directions with a fixed viewpoint. These methods construct a surface reflectance model which characterizes surface appearance under different lighting directions. fgf

3D textures are a way to model the 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 textures. Theseml maps are generated using image re-lighting techniques in which multiple images are captured under different lighting/viewing conditions.

2.1 Image Based Rendering

To be written.

Image-based rendering techniques are widely used in computer graphics. Most frequently image-based rendering appears in the form of texture mapping when an image is used to represent a complex object's appearance. Texture mapping has several significant limitations, the primary one being that only a single lighting condition is captured. If dynamic lighting is needed, texture mapping alone is insufficient.

2.2 Bump Mapping

To be written

2.3 3D texture VS 2D texture

In 2D texture modelling,the reflectance and the strutural properties of natural surfaces are not captured.They fail to capture the variations in surfaces for different lighting and viewing directions.In ed texture mapping 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 mapping is required for realisting modelling of real objects.

2.4 BRDF

To be written

Reflectance texture maps are compact model for representing 3d textures.They model the spatial variation in surface luminance as a function of viewing and illumination direction. The bidirectional reflectance function [12] characterizes the color of a surface as a function of incident light and view directions.

2.5 Polynomial Texture Maping

To be written.

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 [**?**]. 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).

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 Figure 3.2.

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 global component contains the the lighting of a surface point from other parts in a scene, and hence the captures the overall

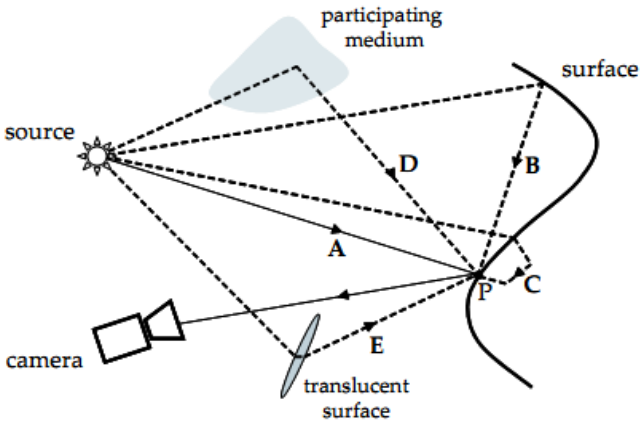


Figure 3.1 The luminance of scene point is due to direct illumination of the point by the source (A) and global illumination due to other points in the scene which is mainly due to inter-reflections (B), subsurface scattering (C), volumetric scattering (D) and translucency (E)

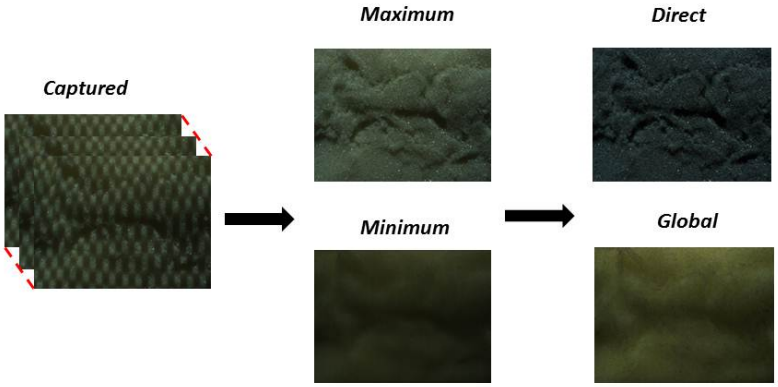
illumination as well as color variations of a surface with lighting direction. As the lighting direction changes, the luminance value of the global part varies significantly.

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 Modelling 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 the 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 Figure 2b). 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



(a) $l_u = .61, l_v = .35$

Figure 3.2 The steps involved in the computation of direct and global images using a set of shifted checkerboard illumination patterns

with low values indicate shadowed regions. A simple thresholding can be done to separate out the shadow part. A suitable threshold is determined using Otsu's method [**?**].

A variety of shadow detection techniques have been proposed in the past to capture this phenomenon in a realistic fashion [**?**, **?**, **?**, **?**]. 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 [**?**]. 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

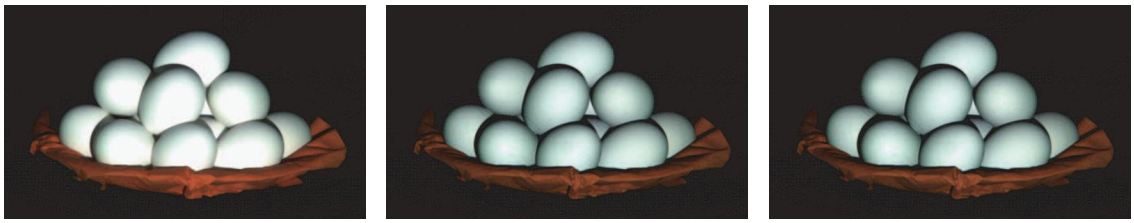
Consider a pair of images of a surface captured from the same view point, but by moving the light source through a short distance. We note that each pixel (surface point) belong to one of the three categories:

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.



(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 3.3 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.

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 the sampling of images will always provide a better estimate. 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 luminances, L_1 and L_2 from the corresponding pixels of images taken from lighting directions p_1 and p_2, the value of the interpolated pixel, L, at lighting position p_0 is given by:

$$L = \frac{\omega_2 L_1 + \omega_1 L_2}{\omega_1 + \omega_2}, \tag{3.1}$$

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:

$$L = L_s + \frac{L_{ns} - L_s}{1 + \chi e^{-d}}, \tag{3.2}$$

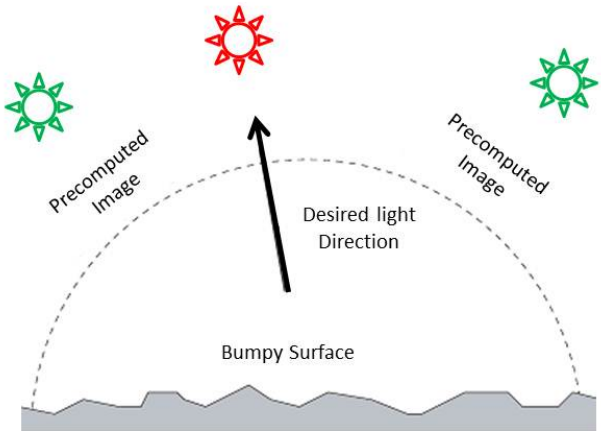
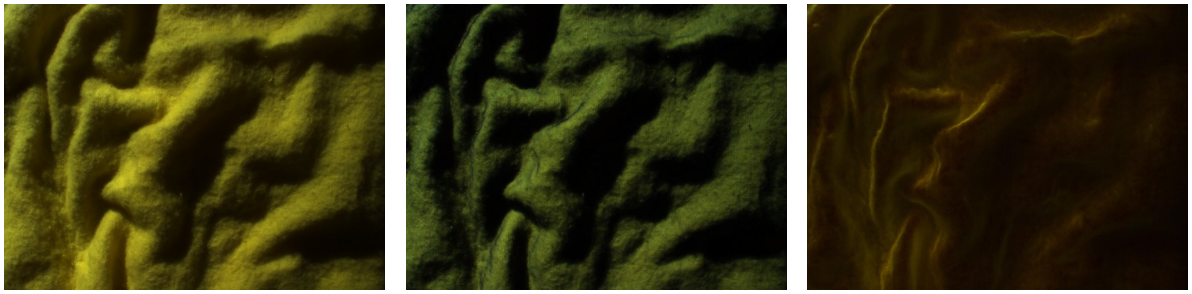


Figure 3.4 shadow modeling by interpolation



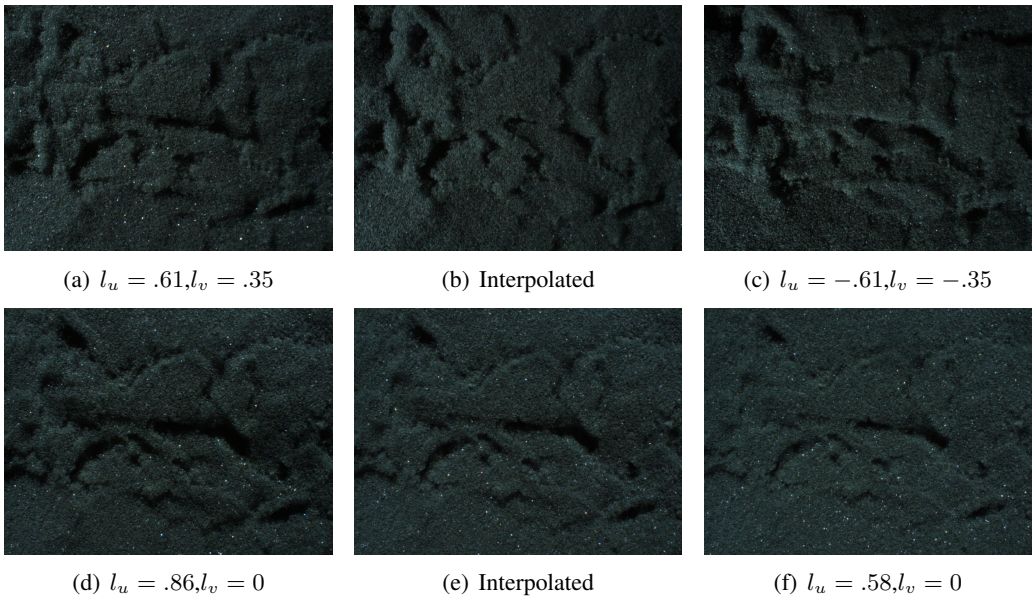
(a) (b) (c)

Figure 3.5 Components of a cloth image for a specific lighting direction: a) original image, b) direct component, c) global component.

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 C 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. One could estimate it in various ways. A quick approximation may be obtained by counting the number of pixels around the one under consideration that are in shadow and not.

The only unknown in carrying out the interpolation is the position p_x at which the transition occurs. 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 these fractions by taking masks of increasing sizes until



(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.

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:

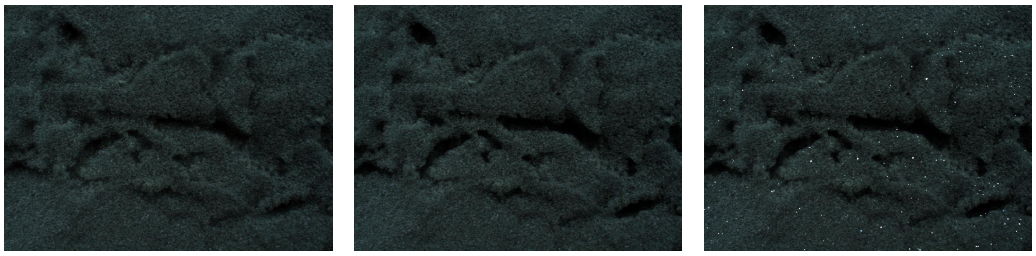
$$p_x = \frac{\chi_{ns} p_1 + \chi_s p_2}{\chi_{ns} + \chi_s} \tag{3.3}$$

The above sigmoidal interpolation can be extended to two dimensions, given an array of image samples with varying lighting positions, and thus one can compute the shadow image for any given lighting direction. We refer to this image as the shadow mask. Figure 3 shows the input images and the interpolated image at two different lighting positions.

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.1.3(a) 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



(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.

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.

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

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

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 shown below in Figure 4 a-c.

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 Figure 5).

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

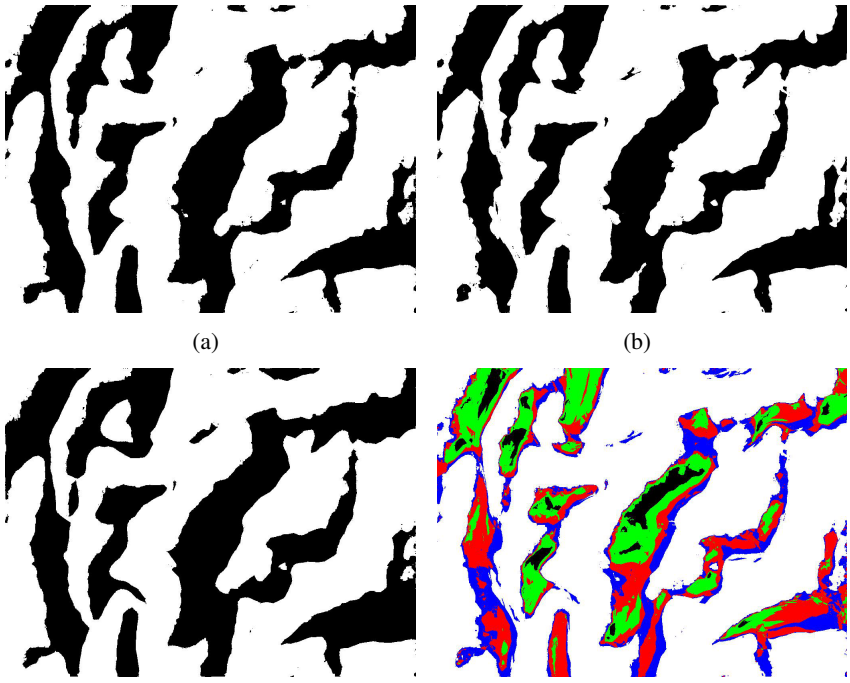


Figure 3.8 a) Binarized image of cloth shadow, b) binary image as rendered by classification technique, c) binary image obtained using interpolation, d) distance image of pixels from classifier boundary. Blue pixels are closest to the hyperplane and include pixels at the edge of a shadow or pixels present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

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

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

an interpolated image is generated (Figure 4(a)). This image is then multiplied with shadow mask. Biquadratic function (equation 8) 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 (Figure 4b). 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.

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

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.

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:

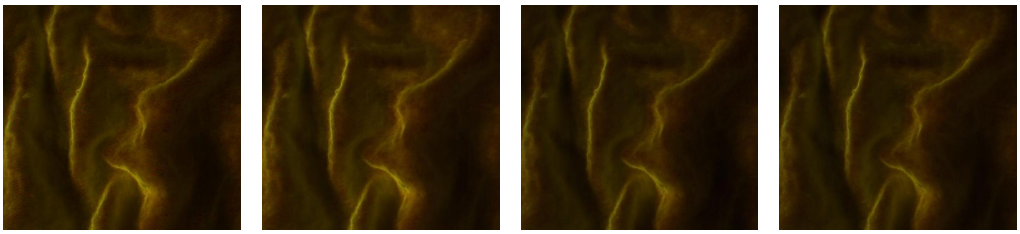
$$S = \eta exp - \left[\frac{(l_u - \mu_x)^2 + (l_v - \mu_y)^2}{\delta} \right], \tag{3.7}$$

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 (see Figures 9d,e). Figure 4(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



(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.

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.

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:

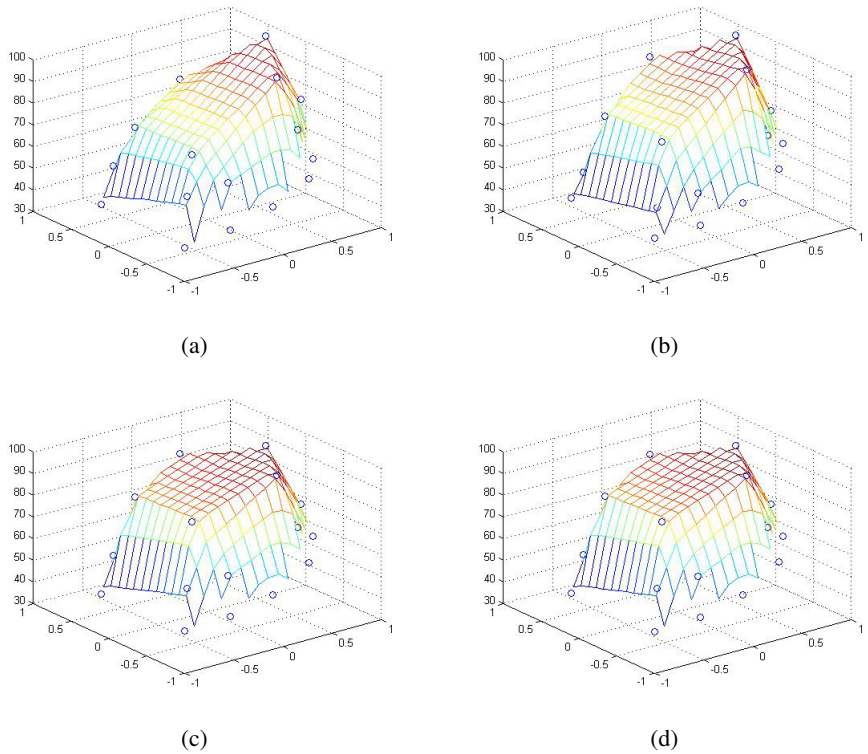
$$\begin{bmatrix} l_{u1}^2 & l_{v1}^2 & l_{u1}l_{v1} & l_{u1} & l_{v1} & -1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ l_{un}^2 & l_{vn}^2 & l_{un}l_{vn}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

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 [**?**], 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

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



(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.

Figure[6 a-d] 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 Figure[7a] shows how the luminance of a given pixel varies with different lighting directions. Figure [7 (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 sampled set of points from different surfaces is shown for comparison in table 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. The camera is mounted vertically above a table that holds the surface (See Figure **??**). The scene is illuminated using a high frequency checkerboard pattern using the projector. The projector 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 images with lighting from a hemispherical set of world coordinates. We capture 30 images from different lighting directions. For each lighting direction, using component separation technique described in [**?**], 25 images are captured and then we separate the image into its global and direct components. The resultant dataset used for modelling contains 60 images per surface, with 2 from each lighting direction.

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 [15], 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.

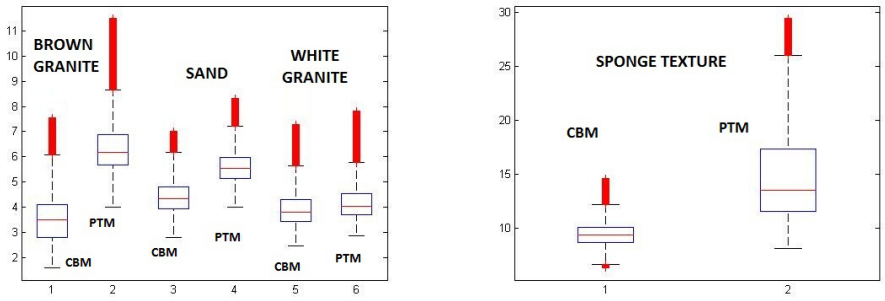
3.5 Experimental Results and Analysis

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 9(a)-(l) 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 9(a)-(c)) The sponge texture (Fig 9(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 photorealistic. In Figure 9(g)-(l), 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 8).



(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.

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 8a-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 8a-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 8(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 10, 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.

(a) Original Sand Texture

(b) Modelled using CBM

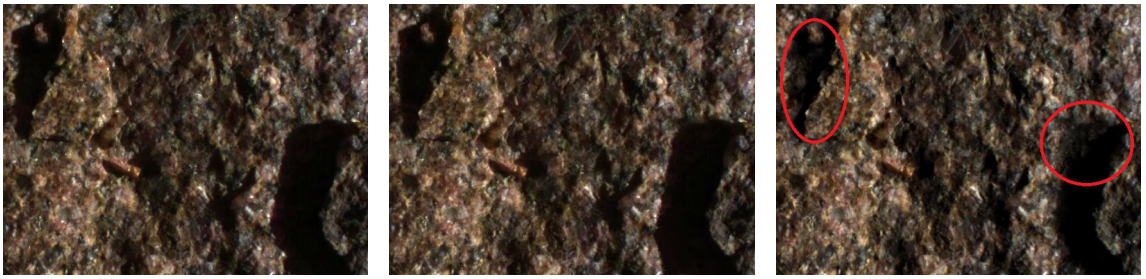
(c) Modelled using PTM

(d) Original Sponge Texture

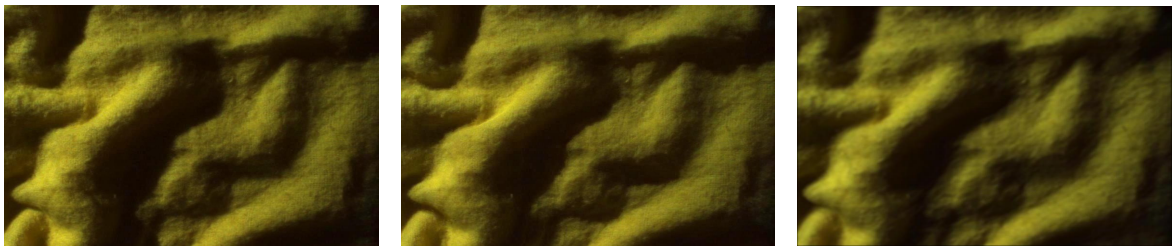
(e) Modelled using CBM

(f) Modelled using PTM

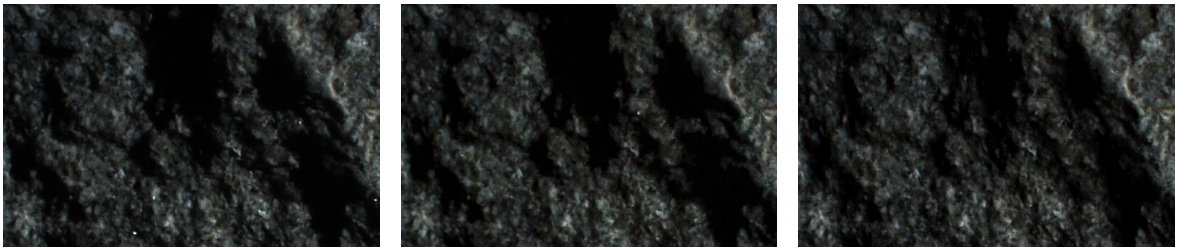
But this improvement is achieved at the cost of more coefficients per pixel as compared to 6 coefficients in PTM. CBM uses a total of 19 coefficients as compared to [24] which requires the estimation of parameters- α (a scalar), β (3-vector) and γ (n-vector). These parameters are estimated per pixel, separately for shadow and specularity modeling. Since 'n' is generally 40-50, therefore total coefficients are very high as compared to ours.



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

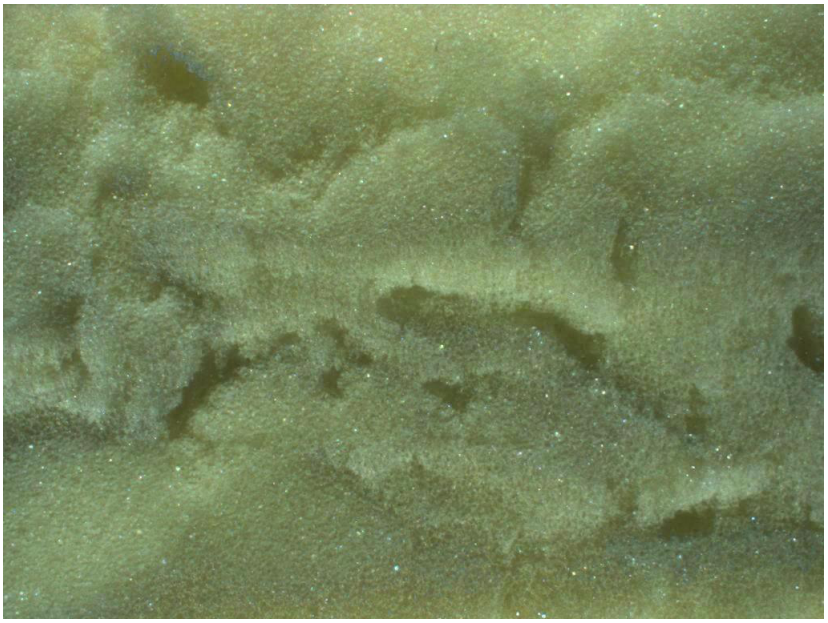


(j) Original Cloth Texture (k) Modeled using CBM (l) Modeled using PTM

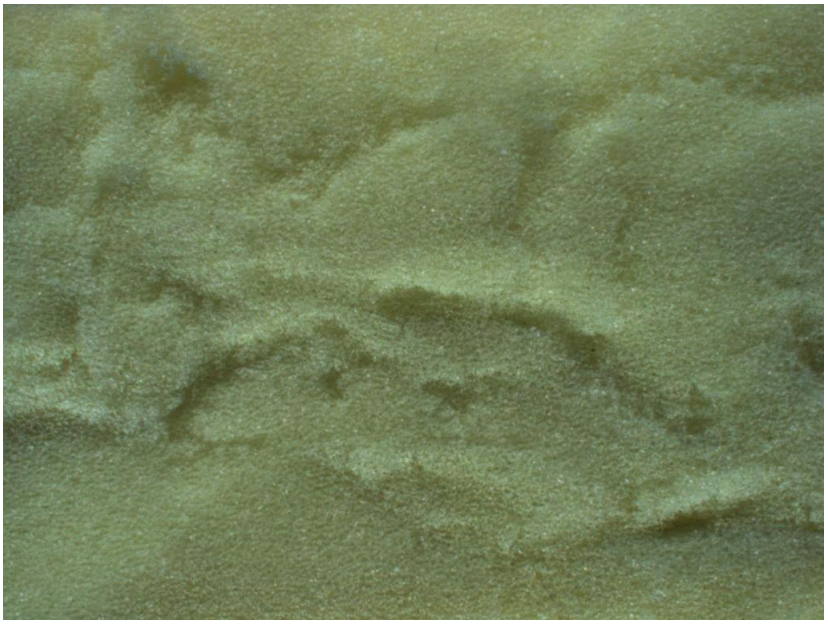


(m) Zoomed part of White Granite (n) Modelled using CBM (o) Modelled using PTM

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 image



(b) PTM based Image

Figure 3.13 Multiple simultaneous Light Sources effect. For (a) and (c) light sources are placed at top(10°) and bottom(180°) side of the texture. For (c) and (d) they are placed at top(10 °) and top left side(45°).

Chapter 4

Component Based Text Segmentation

4.1 Independent Component Analysis (ICA) Model

Independent Component Analysis (ICA) is a useful signal processing and data analysis method developed in the research of blind signals separation. Using ICA, even without any information of the source signals and the coefficients of transmission channels, people can recover or extract the source signals only from the observations according the stochastic property of the input signals. It has been one of the most important methods of blind source separation and received increasing attentions in pattern recognition, data compression, image analyzing and so on, because the ICA process derives features that best present the data via a set of components that are as statistically independent as possible and characterizes the data in a natural way [1-9]. In ICA model, more than one observation signals are needed to achieve the analysis, so when ICA is used to image/video processing, how to generate observations from one image must be firstly considered.

The statistical model in Eq. 4 is called independent component analysis, or ICA model. The ICA model is a generative model, which means that it describes how the observed data are generated by a process of mixing the components si. The independent components are latent variables, meaning that they cannot be directly observed. Also the mixing matrix is assumed to be unknown. All we observe is the random vector , and we must estimate both and using it. This must be done under as general assumptions as possible.

Independent Component Analysis (ICA) has been an active research topic because of its potential applications in signal processing. Only recently it has received attention in image processing tasks. 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

Figure 4.1 Text image where both background and foreground are of same color

(a) (b) (c)

Figure 4.2 (a) Sponge Texture (b),(c) Independent components

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.1 assumes that there exists a linear relationship between the sources S and the observations X. In our case, 'n' is equal to 3.

4.2 Natural Scene Text Binarization

In the recent years, content-based image analysis techniques have received more attention with the advent of various digital image capture devices. The images captured by these devices can vary dramatically depending on lighting conditions, reflections, shadows and specularities. These images contain numerous degradations such as uneven lighting, complex background, multiple colours, blur etc. We propose a method for removing reflections, shadows and specularities in natural scene text images and extracting out the text from a single image.

(a) (b) (c)

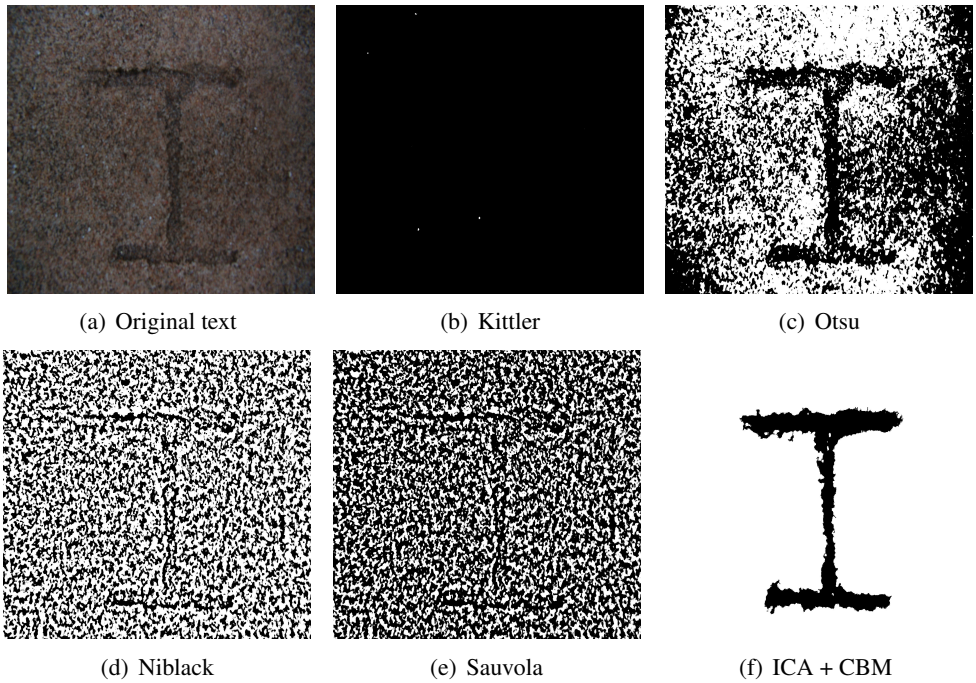
Figure 4.3 (a) Image containing text (b),(c) Independent components

Binarization method is one of the important pre-processing steps in document image analysis system. It directly affects the performance of the subsequent steps like character/word segmentation and 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 [7, 21, 18]. 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 channel as three different sources.

Traditional thresholding based binarization can be grouped into two categories: the one which uses global threshold for the given images like Otsu [16], Kittler *et al*. [10] and the one with local thresholds like Sauvola [22], Niblack [15]. In global thresholding methods [16, 20], 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 [4] 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 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. Mishra *et al* [14] has recently formulated the problem of binarization 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. However, their method is sensitive to the initial auto seeding process. On the



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

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

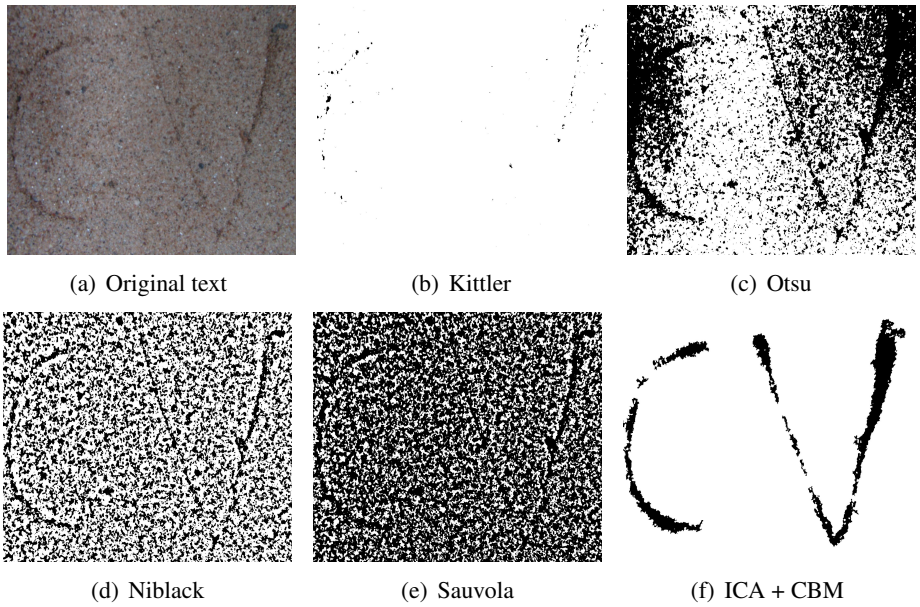
Figure 4.4 Binarized text

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.

The primary issue related to binarizing text from scene images is the presence of complex/textured background. When the background is uneven as a result of poor or non-uniform lighting conditions, the image will not be segmented correctly by a fixed gray-level threshold. These complex background vary dramatically depending on lighting, specularities, reflections and shadows. The above methods applied directly to such images give poor results and cannot be used in OCR systems. In this paper, we do an ICA based decomposition which enables us to separate text from complex backgrounds containing, reflections, shadows and specularities. 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 methods. Some of the word images that we used for experiments are shown in Fig **??**.

4.3 Binarization process

A wide variety of ICA algorithms are available in the literature [8, 9]. These algorithms differ from each other on the basis of the choice of objective function and selected optimization scheme. Here we use a fast fixed point ICA algorithm to separate out the text from complex background in images. A



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

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

Figure 4.5 Binarized text

Blind Source Separation method based on SVD [24] can also be used. Fig. 4.8 shows the complete framework for the proposed method.

4.3.1 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. Therfore, 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:

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

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 [8] for this. We summarize the fixed point ICA method in Algorithm 1.

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

From this step, we get three independent sources or components. Fig. 4.7 shows the background and the foreground extracted. The resultant independent components for a particular word image can be seen in Fig. 4.9 which shows the independent component free from reflective background and containing maximum information of the foreground text.

4.3.2 Thresholding

Otsu thresholding [16] 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. 4.9). We can also apply Kittler [10] 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: 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. 4.10, adaptive thresholding method may perform better than global one. However, in practise we note that a simpler global thresholding scheme works well in most cases.

4.4 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 [16]	.68	.75	69.17
Sauvola [22]	.63	.81	66.94
Kittler [10]	.66	.76	64.33
Niblack [15]	.70	.76	71.32
MRF [14]	.79	.86	80.38
Proposed	.86	.83	83.60

Table 4.2 OCR Accuracy (%)

Method	Word Accuracy
MRF [14]	43.2
Proposed	61.6

We compare the performance of our method with four well known thresholding algorithms i.e Kittler [10], Otsu [16], Niblack [15] and Sauvola [22]. We also compare with the recent method by Mishra *et al* [14]. 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. The qualitative results are shown in Fig. 4.11.

We took around 50 images from the dataset and generated its ground truth images for pixel level accuracy. We use well known measures like precision, recall and F-score 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 [14] based model as shown in Table II.

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. Fig. 4.16 shows that our technique can also be applied to text image containing two different types of colored text. We analyze that the above methods do not work in the case where there is a complex and textured 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. 4.17. Our method fails in cases where foreground text and the background are of the same color. Moreover, the approach works only with color images.

4.5 Applications

4.5.1 Inscribed Text Segmentation

Inscribed text is difficult to extract from one image as both the foregound and the background is of same color. So we capture multiple images 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 4.2 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 out the shadows(Figure 4.3). Then we apply our component based model to efficiently binarize the text embedded (Figure 4.5).

4.5.2 Enhancing Edge Extraction

Boundary detection is a fundamental task in computer vision, with broad applicability in areas such as feature extraction, object recognition and image segmentation. The majority of papers on edge detection have focused on using only low-level cues, such as pixel intensity or color [15]. Recent work has started exploring the problem of boundary detection based on higher-level representations of the image, such as motion, surface and depth cues [68], segmentation [9], as well as category speci

c information [10, 11].

Edge detection refers to the process of identifying and locating sharp discontinuities in an image. The discontinuities are abrupt changes in pixel intensity which characterize boundaries of objects in a scene. Classical methods of edge detection involve convolving the image with an operator (a 2-D filter), which is constructed to be sensitive to large gradients in the image while returning values of zero in uniform regions. There are an extremely large number of edge detection operators available, each designed to be sensitive to certain types of edges. Variables involved in the selection of an edge detection operator include Edge orientation, Noise environment and Edge structure. The geometry of the operator determines a characteristic direction in which it is most sensitive to edges. Operators can be optimized to look for horizontal, vertical, or diagonal edges. Edge detection is difficult in noisy images,

since both the noise and the edges contain highfrequency content. Attempts to reduce the noise result in blurred and distorted edges. Operators used on noisy images are typically larger in scope, so they can average enough data to discount localized noisy pixels.

Boundary detection constitutes a crucial step in many computer vision tasks. A boundary map of an image can provide valuable information for further image analysis and interpretation tasks such as segmentation, object description etc. Fig. 1 shows an image and the associated boundary map as marked by human observers. It can be noted that the map essentially retains gross but important details in the image. It is hence sparse yet rich in information from the point of scene understanding. Extracting a similar boundary map is of interest in computer vision

4.5.3 Shadow Detection

Shadows, created wherever an object obscures the light source, are an ever-present aspect of our visual experience. Shadows can either aid or confound scene interpretation, depending on whether we model the shadows or ignore them. If we can detect shadows, we can better localize objects, infer object shape, and determine where objects contact the ground. Detected shadows also provide cues for lighting direction and scene geometry. On the other hand, if we ignore shadows, spurious edges on the boundaries of shadows and confusion between albedo and shading

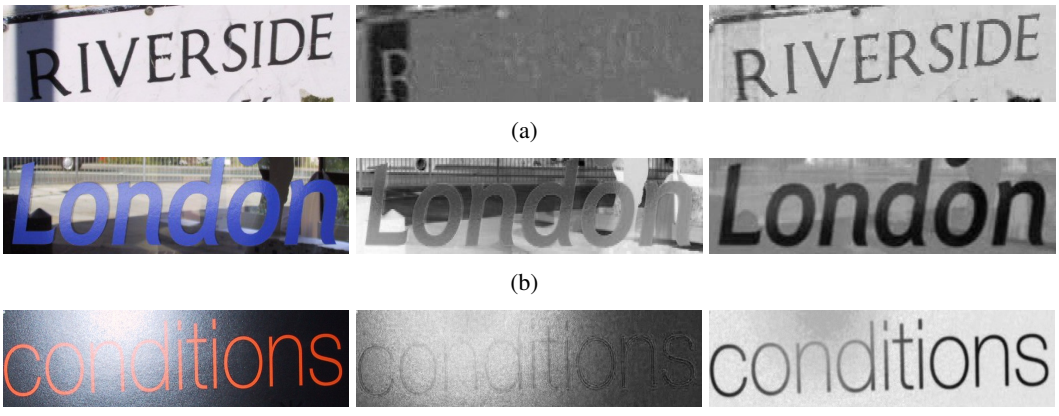
can lead to mistakes in visual processing. For these reasons, shadow detection has long been considered a crucial component of scene interpretation (e.g., [17, 2]). But despite it importance and long tradition, shadow detection remains an extremely challenging problem, particularly from a single image. The main difculty is due to the complex interactions of geometry, albedo, and illumination. Locally, we cannot tell if a surface is dark due to shading or albedo, as illustrated in Figure 1. To determine if a region is in shadow, we must compare the region to others that have the same material and orientation. For this reason, most research focuses on modeling the differences in color, intensity, and texture of neighboring pixels or regions. Many approaches are motivated by physical models of illumination and color [12, 15, 16, 7, 5].

The obstruction of light by objects creates shadows in a scene. An object may cast a shadow on itself, called self-shadow. The shadow areas are less illuminated than the surrounding areas. In some cases the shadows provide useful information, such as Shadow detection and removal is an important pre-processing task in many of the computer vision applications. The shadows may give rise to false segments in the image segmentation process. Also, shadows may be wrongly detected as objects in object detection algorithms. Various pixel-based and region-based methods were proposed to detect the shadows in an image. This section briefly reviews some of the important research works in shadow detection and removal. the relative position of an object from the source. But they cause problems in computer vision applications like segmentation, object detection and object counting. Thus shadow detection and removal is a pre-processing task in many computer vision applications. Based on the intensity, the shadows are of two types hard and soft shadows. The soft shadows retain the texture of the

background surface, whereas the hard shadows are too dark and have little texture. Thus the detection of hard shadows is complicated as they may be mistaken as dark objects rather than shadows.

(a)

Figure 4.6 (a) Image containing Text over another Text (b) Foreground Text (c) Background Text (d) Text extracted



(a)

(b)

(c)

Figure 4.7 Foreground and Background Extracted: (a) Shadowed background and foreground text (b) Reflective background and foreground text (c) Specular background and foreground text

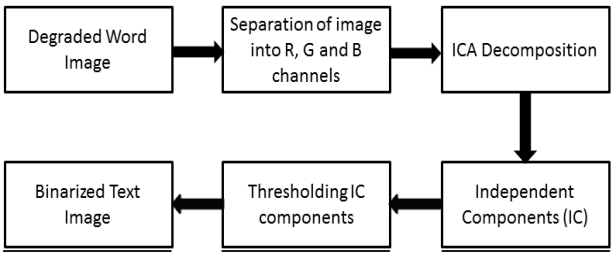
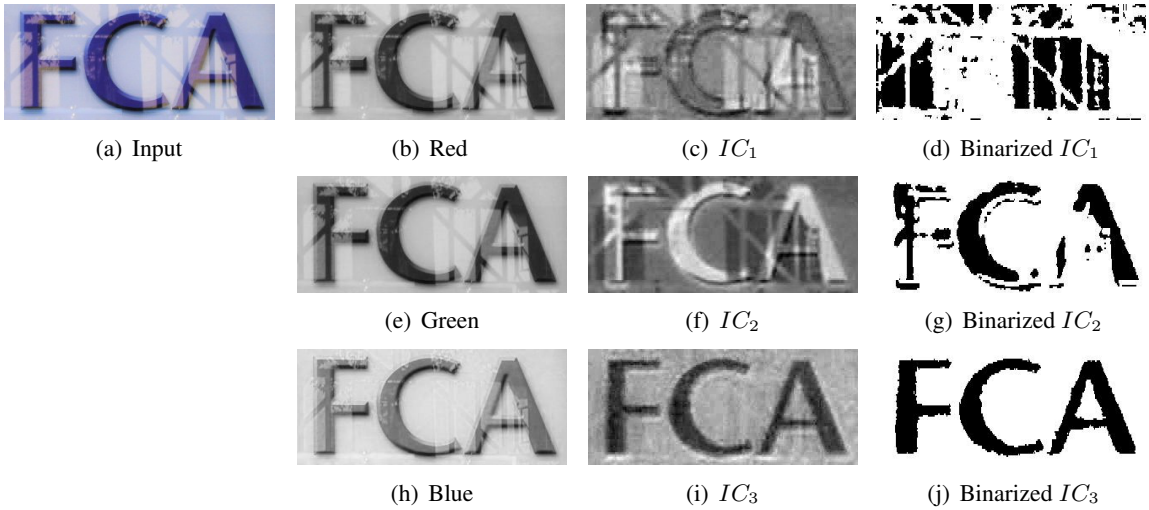


Figure 4.8 Framework for the proposed method

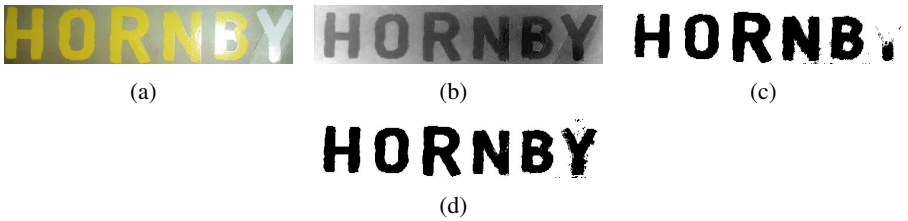


(a) Input (b) Red (c) IC_1 (d) Binarized IC_1

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

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

Figure 4.9 (a) Original word image (b),(e),(h) R, G and B channel respectively (c),(f),(i) Independent Components, (d),(g),(j) Binarized image

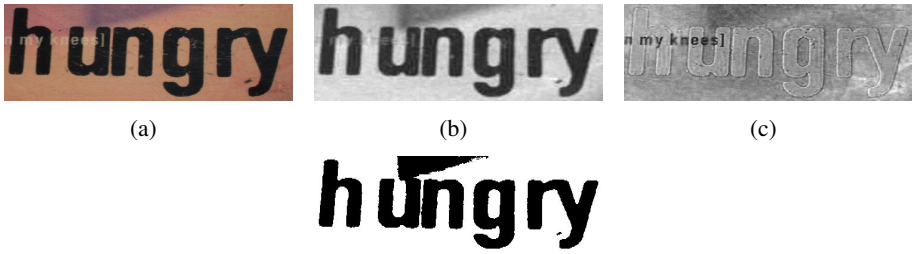


(a) (b) (c)

(d)

Figure 4.10 (a) Text containing specular highlight (b) IC (c) Otsu (d) Niblack

Figure 4.11 Comparison of Binarization algorithms and the proposed method (From left to right Original, MRF, Kittler, Otsu, Niblack, Sauvola, Proposed) Text containing specular background



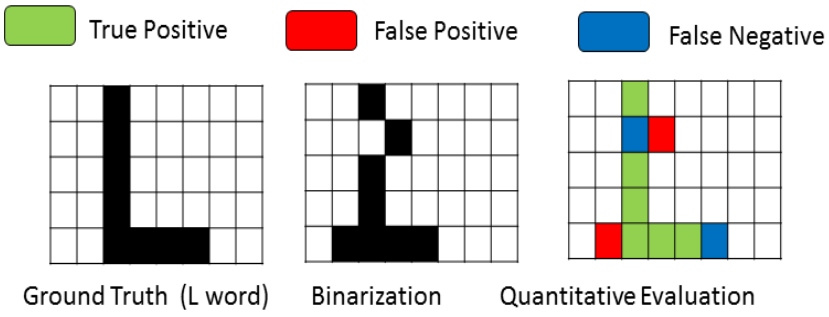
(a) (b) (c)

(d)

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

Figure 4.13 (a) Image containing Text over another Text (b) Foreground Text (c) Background Text (d) Text extracted

Figure 4.14 (a) Image containing Text over another Text (b) Foreground Text (c) Background Text (d) Text extracted



(a)

Figure 4.15 (a) Image containing Text over another Text (b) Foreground Text (c) Background Text (d) Text extracted

(a)

Figure 4.16 (a) Image containing Text over another Text (b) Foreground Text (c) Background Text (d) Text extracted

Figure 4.17 Failure case where both the foreground and background are of same color

Figure 4.18 (a) Image containing Text over another Text (b) Foreground Text (c) Background Text (d) Text extracted

Figure 4.19 (a) Image containing Text over another Text (b) Foreground Text (c) Background Text (d) Text extracted

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

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

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

[5] 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.

[6] 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.

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

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

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

[10] 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.

[11] 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.

[12] L. MacDonald and S. Robson. Polynomial texture mapping and 3d representations. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38, 2010.

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

[14] 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.

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

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

[17] 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.

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

[19] 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.

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

[21] 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.

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

[23] 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.

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