Low Light Auto Focus Enhancement in Cell phone Cameras

A Project Report

submitted by

NATTHA RAJ KUMAR

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2015

THESIS CERTIFICATE

This is to certify that the thesis titled **Low Light Auto Focus Enhancement in Cell phone Cameras**, submitted by **Nattha Raj Kumar**, to the Indian Institute of Technology, Madras, for the award of the degree of **Dual Degree**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof.A.N. Rajagopalan
Research Guide
Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 6th MAY 2015

ACKNOWLEDGEMENTS

I take this opportunity to express my deepest gratitude to my project guide Dr. A.N. Rajagopalan for his valuable guidance and motivation throughout the project. I am very grateful to him for providing his valuable time to guide me during the project.

It is a privilege to be a student in IIT Madras. I express special thanks to all my teachers for all the academic insight obtained from them. I also acknowledge the excellent facilities provided by the institute to the students.

I am indebted to my parents for their unconditional love, support and guidance. I dedicate this thesis to them.

ABSTRACT

KEYWORDS: Auto focus, low-illumination, cell phone cameras.

Passive Auto Focus in Cell phone Cameras fails to produce a sharp image under low-illumination conditions due to the failure of focus measure in picking up the sharp image. In this, we propose a method to enhance the sharpness of image in low-illumination conditions.

The device we used is a Nokia Lumia 520 which is based on the Windows Phone 8 operating system. The device captures a stack of frames under different lens settings and send the frames to a computer through TCP to run offline for ease of research. We use retinex theory to separate the reflectance and shading part and apply various focus measures and then use tensor voting to pick one sharp frame. We show that the frame picked by this method has small blur compared to the frame picked by default camera application. We also tried to enhance the image by doing illumination compensation using the shading component. However, this method worked only under certain illumination conditions, as the retinex theory parameters needs to be varied for different illumination conditions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

SML	Sum Modified Laplacian
EOG	Energy of Gradient
EOL	Energy of Laplacian
CMAN	Contrast Measurement Adaptive to Noise
TV	Tensor Voting
DCT	Discrete Cosine Transform
FFT	Fast Fourier Transform
PSF	Point Spread Fuction
AF	Auto Focus
SDK	Software Development Kit
TCP	Transmission Control Protocol

CHAPTER 1

INTRODUCTION

Now a days, most people have access to cell phones and cell phone cameras are widely used in capturing photos in all occasions. Images can be captured at any time in the day, indoor or outdoor. The quality of the image taken will be good when taken in good illumination. Bad illumination include improper lighting conditions or images captured during night. In bad illumination conditions, the number of photons that fall on the image sensor are less and hence the image quality decreases.

One of the aspects of image quality is sharpness, which depends on the performance of Auto focus system present in the camera. It is important to increase the functionality of Auto focus to serve the consumer's desire of capturing good quality images in unfavourable lighting conditions.

1.1 Working of Auto-Focus

A robust Auto focus system is important in capturing a sharp image. Fundamental principle of Auto focus is to place the scene,that is to be captured at the focal position of the lens, by adjusting the its position. Basic optics says that, the part of the scene that is at the focal position of the lens will appear sharp and the remaining part appears blurry. Two images, taken at random lens positions are shown in Figure 1.1, we can clearly observe the sharpness difference between the two frames. So, it is very important to place the lens at the correct position automatically to get the sharp image. There are two types of Auto Focus systems: Active Auto focus and Passive Auto focus.

Active auto focus uses external sensors to measure the distance of the object from the lens and adjusts the lens position to get the sharp image. It fails when the sensor reading are hindered by some object, for example, when the images are captured through a

ily a great start every day with **Kellogg's Corn Flakes.**

es has toasted flakes of golden corn, sun-ripened in
s is fortified with 8 essential vitamins and iron. These
rform throughout the day and stay healthy.
bowl of Kellogg's Corn flakes with your choice of

pple Corn Flakes

s Corn Flakes, 1 medium size red apple (sliced), 2 cups (120 ml)
es (chopped) and 1 teaspoon honey (10 gms)

cups of cold/hot milk to a bowl with 1½ cups of Kellogg's Corn
d add it to the mix. Garnish with 2 chopped dates and top it up
th a spoonful of honey.

family a great start to the day!

Figure 1.1: Frames taken at different lens positions

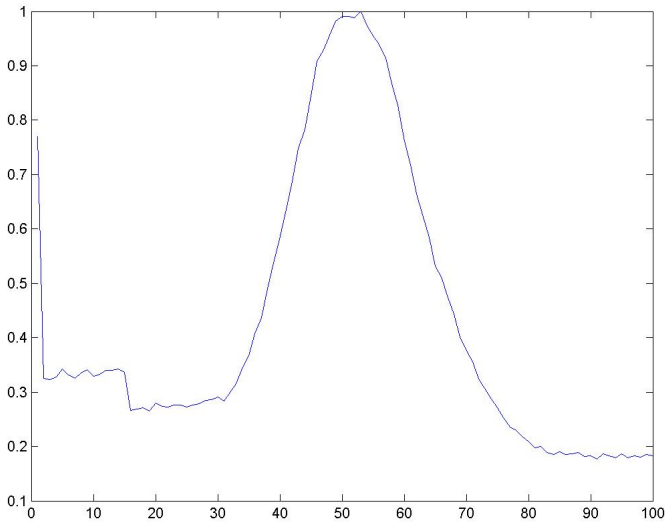


Figure 1.2: Normalized sharpness plot when focus measure is applied across a stack of 100 frames, captured in good illumination. Here, frame:53 is picked up as the sharp frame

window. It consumes lot of power, which makes it not suggestible to implement on cell phone cameras, where minimizing power consumption is a high priority issue.

Passive auto focus picks up a sharp frame from the high frequency information present with in the image. A focus measure is applied in a window across a stack of frames, that are captured with different lens positions and the peak in the focal measure plot gives the sharp image.For example, Sum Modified Laplacian(SML), a focus measure, is applied across a stack of 100 frames and the focus measure plot is shown in the Figure 1.2. In bad illumination, the focal measure fails to give a single peak, as shown in Figure 1.3, and it becomes difficult to find the sharp frame.

People have come up with alternate suggestions like using flash, increasing exposure time, increasing focus window size. But, using flash is not suggestible always as it might add artefacts to the scene, might create distraction from the scene that we want to capture and flash is not allowed everywhere like museum's. Increasing exposure time works but it might induce motion blur. Increasing focus window size might result in generating two peaks, corresponding to the objects present at different depths and it increases the complexity of computations as well.

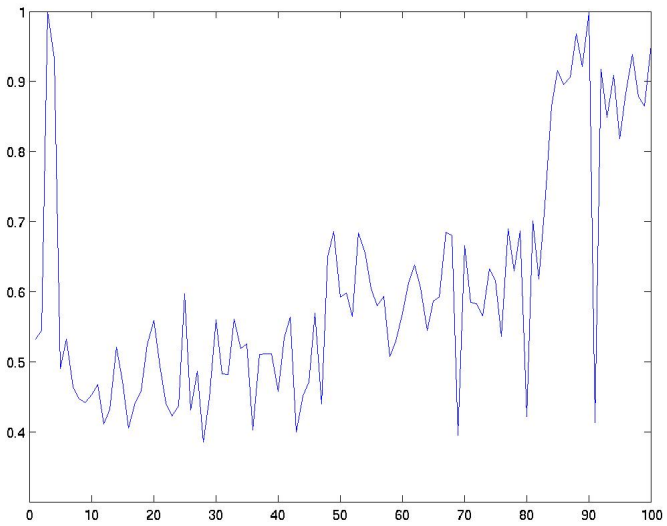


Figure 1.3: Normalized sharpness plot of stack of 100 frames, captured in bad illumination.

1.2 Previous Work

People have studied on which focal measures might work better in low-illumination conditions specific to Cell phone Camera Auto focus application in Xu *et al.* (2014), which will be discussed in detail in chapter 4. They 've shown that Tenegrand focus measure works better compared to other focus measures, but concluded that even it fails in some conditions and suggested for a more elaborate method to pick up the sharp frame.

(Gamadia *et al.*, 2007), have worked in addressing this issue by changing the pipeline of the camera.They tried image enhancement preprocessing approach to reduce the effect of the additive white Gaussian noise which corrupts the raw image sensor data. They addressed the issue where sharpness curve is almost flat. They studied the effect of SNR on the shape of sharpness function. Their auto focus system model for noise analysis and image enhancement preprocessing is shown in Figure 1.4.

They try to increase the SNR by reducing the effect of noise in the observation d. They estimate the noise free estimate $\hat{y}$ of the noise free image from the noisy image d

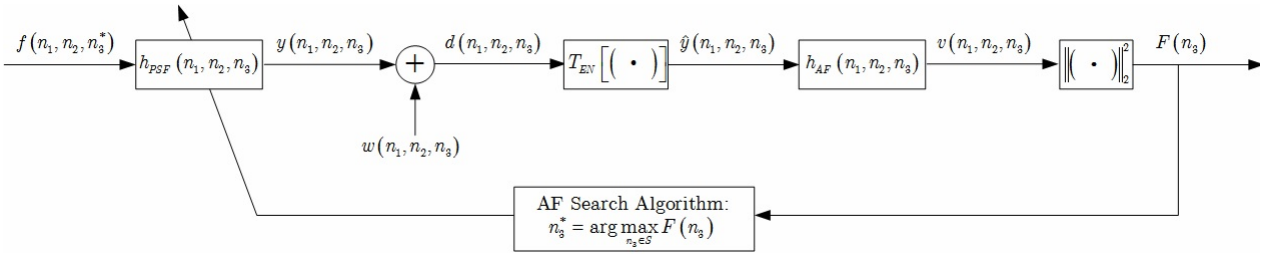


Figure 1.4: AF system model for noise analysis and image enhancement preprocessing.

using the below equation.

$$\hat{y} = T_{EN}[d] = T_{LPF2}[T_{CE}[T_{LPF1}[T_{FR}[d]]]]$$

The steps involved in enhancement are: Fault pixel removal step, low pass filter to smooth, contrast enhancement step and low pass filter to smooth. This can be used when the focus measure is flat, but it need not be always and hence we proposed the below model.

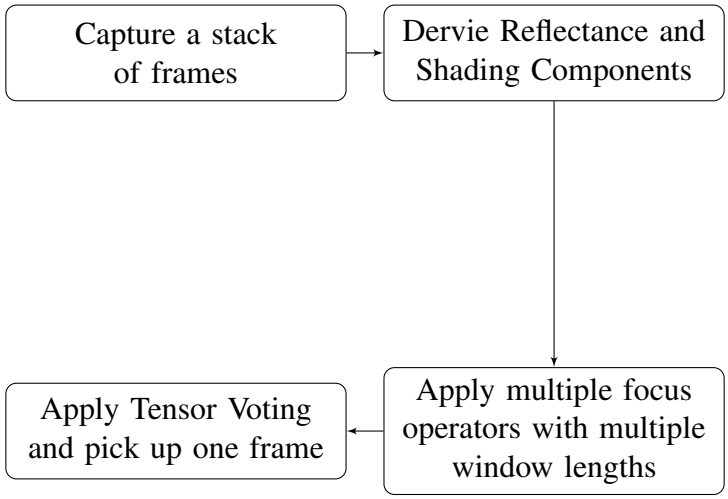


Figure 1.5: Flowchart of the steps involved.

1.3 Proposed Algorithm

In bad illumination conditions, image tends to have more dark regions which makes it difficult for the Auto Focus measure to pick up the edge details. So, we initially do split the images into reflectance and shading components and apply the focus measures

on the reflectance component, which is uniformly illuminated and contains better edge details than the original image. We divide the entire area of interest into N regions and apply focus operator across these N regions separately.

Still sometimes focus operators fail to pick up single frame. So, we apply multiple focus operators with multiple window sizes and get a set of frame numbers picked by each of the operators with each window size. We then use Tensor voting algorithm to pick up one single frame and we show experimentally that the frame picked by this method has smaller blur compared to the frame picked by the conventional method.

1.4 Organisation of Thesis

The Chapter on *Device* gives an in-depth account of the mobile platform, the application written for the mobile platform and the host side software.

The Chapter on *Retinex Theory* discusses implemented algorithm to derive reflectance and shading components from a single image. We show that the peak in sharpness curve when applied across the reflectance stack is same as the peak in sharpness curve when applied across the image stack.

The Chapter on *Focus Measures* presents different types of focus measures and explains few focus measures in detail, which we've used in our algorithm.

The Chapter on *Tensor Voting* explains the need for going for multiple window sizes and the Tensor Voting framework that picks up one single frame from multiple frames which are picked by different focus operators,applied at different pixel locations, with different window sizes.

The Chapter on *Experimental Results* presents the experimental results. We show that the frame picked up by our method has the small blur kernel compared to the blur kernel of the frame picked up by conventional method and we showed an enhanced image after illumination compensation.

CHAPTER 2

THE DEVICE

Mainly, we would like the device to capture a stack of frames with different lens positions and send it to the local host, where all post processing happens. The following things are considered in choosing the mobile platform:

- The device should be of low cost. We are looking at a mobile which would not cost above Rs. 15,000. This gave us the option to choose Android mobiles or the low end Windows Phone Mobiles.
- The device should have low end camera specifications.
- The device should have a good Software Development Kit. Android and Windows Phone both have a very good SDK.
- The device should have manual focus ability and easy image handling capabilities. With Nokia Imaging SDK, we can the control over lens position in Windows Phone.
- The device should have Transmission Control Protocol(TCP) capability. Most smart phones available today have this feature.

Considering the above points, specially the ease of control over camera hardware, we chose Nokia Lumia 520.

Camera Specifications:

- Main camera: 5.0 MP
- Main camera f-number/aperture: f/2.4
- Camera focal length: 28 mm
- Camera minimum focus range: 10 cm

From the point of developing applications, the Windows Phone 8 Software Development Kit has a very simple interface with coding in C# and GUI design using XAML script. Visual Studio makes it very easy to create applications fast. A number of code examples for developing WP8 apps are available at Network (2013–)

2.1 Device side application

We developed a camera application, where the lens is manually moved from one end to the other end in a fixed small increment, and the images were captured at each interval. For each of the stack of images captured, its sharpness value is calculated using a focus measure, and the peak in the sharpness curve is recorded as the optimum in-focus position.

The primary use of this application is to send the captured frames to a host computer, so that all image processing can be done offline. This gives us the power of computer and flexibility to try out different techniques. Primarily, there are two buttons and two text boxes. Their functionalities are:

Focus sweep button : This enables the application to send data to the connected host. We need to enable this before capturing a scene, to save frames in the computer

Click button: Capturing of the scene starts after we click this button.

IP address and port number text box: We need to provide the ip address and the port number of the host computer to which we want to transfer the frames in this text box.

Creating and using a TCP socket client

We send the data from the device to the host through TCP communication. An user interface for TCP socket client is created in our application. Then an end point is defined, which contains details like host name or ip address and remote port that is required by the application to connect to a server. The data to be transmitted is placed in a buffer as an array of bytes. A callback method is defined for the completed event. We create a context object which contains data buffer, callback method, and various other context-specific data and then passed to the asynchronous call. Once the call is over, we check the status of this object for operation result. We'll wait until the completed event is over or if call times out.Finally, we use a close method to close the socket that was created.

When the device gets connected to the host, it can be treated as continuous information channel with out any prior knowledge of the size of data being sent. To retrieve the data, we encapsulate the data in keywords, which are known to the computer.This results in just searching for starting and ending keywords of a particular key word. Example: We use 'STFR53' to indicate the data corresponding to the start of 53 frame and 'EDFR53' to indicate the end of 53 frame data.

```
// byte_preview has the RGB frame data in bytes
// app_comsocket.Send is used to send data over the connection.
app_comsocket.Send("STFR" + focus_counter.ToString() + "\n");
app_comsocket.Send(byte_preview);
app_comsocket.Send("EDFR" + focus_counter.ToString() + "\n");
```

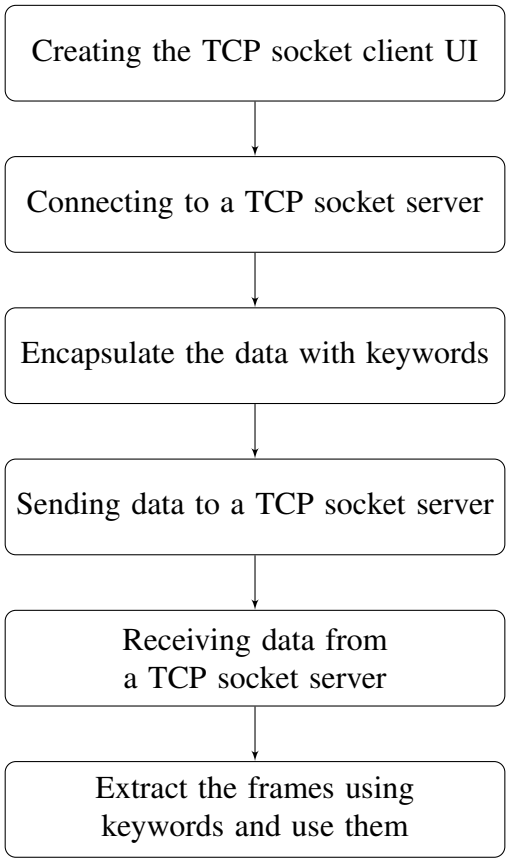


Figure 2.1: Flowchart of the steps involved.

2.2 Host side application

This includes methods like data reception, extraction and other image processing methods. Data reception and extraction are written in python and all image processing in done in matlab.

The communication from mobile to computer is done using the inbuilt socket module in python. As mentioned above, the data from the mobile is encapsulated and sent in keywords. These keywords are used to parse the incoming data and split it into frames accordingly. We continuously receive data and save the data between the start token and end token and save in a '.dat' file and then extract using the keywords.

Receiving the data from the device:

```
#start of the data
strt_token = 'S\x00T\x00I\x00G\x00'
#end of the data
end_token = 'E\x00D\x00I\x00G\x00'
frame_token = 'S\x00T\x00I\x00M\x00'
continuous_recv(strt_token, end_token, frame_token,
   'tmp/burst/tokens.dat')
```

Extracting the data from '.dat' file, using the keywords

```
#start of the frame
fstart = 'STFR'
#end of the frame
fend = 'EDFR'
extract_images('tmp/burst/tokens.dat', 100, fstart, fend,
   'tmp/burst/src')
```

Once we've the stack of frames, we use them to try various methods.

CHAPTER 3

RETINEX THEORY

Land *et al.* (1971) proposed in 1971 that every scene can be divided into two components, reflectance and shading components, where reflectance remains constant under different illumination conditions and shading represents the illumination of the scene. Deriving these components from a single image is an ill-posed problem and many have come up with suitable cues, specific to their application and separated the two components.

We've implemented the paper by Fan et al, which uses multiple cues to eliminate highlights and shadows effectively. Also, the shading component can be used to estimate the illumination of the scene and can be used to compensate for illumination changes and give an enhanced image to the user. One of the main reasons for choosing this paper is because, it is able to obtain the shading part very effectively and was able to keep complete color and edge information in the reflectance part, which is what we need for this application.

On the gradient domain of logarithmic image, they have used weighted edge map of c_1c_2, o_1o_2 color space as well as the chromatic characteristics and brightness characteristics of the neighbourhood pixels and proposed the classifier. Detailed approach is given in 3.2

3.1 Separation of Reflectance and Shading components

Let $I(x, y)$ be the image, $R(x, y)$ be the reflectance component and $S(x, y)$ the shading component. They are related by the following expression:

$$I(x, y) = R(x, y)S(x, y)$$

To derive the reflectance and shading components, they used the following operations:

$$I_L(x,y) = \log(I(x,y)) = \log(R(x,y)) + \log(S(x,y)) = R_L(x,y) + S_L(x,y)$$

The $\log$ operator makes the multiplicative relation to additive relation between the reflectance and shading components. Next, they apply horizontal derivative filter $f_x([0,-1,1])$ and vertical derivative filter $f_y([0,-1,1]^T)$ separately and obtain I^k_{Lx}, I^k_{Ly}, where k stands for R,G,B channels. Then, they are separated into R^k_{Lx}, R^k_{Ly} and S^k_{Lx}, S^k_{Ly} using gradient classifier M^k, which is calculated using multiple cues, that are to be discussed below. Finally, a deconvolution is applied to get back the reflectance R(x,y) and shading S(x,y) components.

$$R^k_j = \begin{cases} I^k_j, & \text{if } M^k = 1 \\ 0, & \text{otherwise} \end{cases}$$

$$S^k_j = \begin{cases} 0, & \text{if } M^k = 1 \\ I^k_j, & \text{otherwise} \end{cases}$$

where $j = Lx, Ly$ and $k = (R,G,B)$ and $M^k = 1$ corresponds to the reflectance part and $M^k = 0$ corresponds to the shading part.

3.2 Classification using Color model, Chromatic information and brightness information

The sensitivity of the different color models with respect to the imaging conditions is shown in Table 3.1.

Under the assumption that the object is composed of matte and the surface is dull, the c_1c_2 color model only depends on the the surface albedo and sensors. So c_1c_2 can denote the reflectance component of image. But, the assumption is not realistic always. Then, c_1, c_2 varies with material and highlights. o_1o_2 depends not only on the sensors and surface albedo,but also on object geometry and highlights.

Table 3.1: Overview of color models invariance to various color models.

$Colormodel$	$Shape$	$Shadow$	$Highlights$	$Material$
RGB	$+$	$+$	$+$	$+$
c_1c_2	$-$	$-$	$+$	$+$
o_1o_2	$+$	$+$	$-$	$+$

- denotes invariant of the color model to the imaging condition
+ denotes sensitivity of the color model to the imaging condition

The information containing material characteristics can be obtained by image gradient in c_1c_2, o_1o_2 color space. A reflectance related weight edge map W, is obtained from $\nabla c_1, \nabla c_2$ and $\nabla o_1, \nabla o_2$.

$$W = Mcc.Moo$$

where $Mcc = \max(\nabla c_1, \nabla c_2)$ and $Moo = \max(\nabla o_1, \nabla o_2)$. A threshold T_c is used to separate each gradient into shading and reflectance component.

$$M_c = \begin{cases} 1, & \text{if } W \geq T_c \\ 0, & \text{otherwise} \end{cases}$$

The shading component classified with color information contains the reflectance components on intensity change that cannot be classified correctly. So, a further classification is done on shading component using brightness information. Gradient amplitude of shading derivatives $|S_L^k|$is compared with a threshold and those below the threshold are directly classified as shading components and those above the threshold are further processed.

$$|S_L^k| = \sqrt{(S_{Lx}^k)^2 + (S_{Ly}^k)^2}$$

For the remaining pixels, a window around the pixel is taken and is divided into two regions $p1, p2$ by a line passing through the center pixel, whose sum of gradients is maximum. We compute the intensity averages $\bar{I}_{p1}^k, \bar{I}_{p2}^k$ in those two regions, and their absolute difference is compared to a threshold to further classify.

$$M_I = \begin{cases} 1, & \text{if } |\bar{I}_p1^k - \bar{I}_p2^k| \leq T_i \\ 0, & \text{otherwise} \end{cases}$$

Next, one more classification is done using chromatic information and the shading component is further divided.We first transfer the RGB image into chromaticity color rg. For every pixel, that is classified into shading part, we take a window around that pixel and divide it into two regions, p_1, p_2 as mentioned above. The chroma average $\bar{r_1}, \bar{r_2}, \bar{g_1}, \bar{g_2}$ of the pixel in the two regions are calculated. A measure, $M_{rg} = \sqrt{(\bar{r_1} - \bar{r_2})^2 + (\bar{g_1} - \bar{g_2})^2}$ is compared with a threshold T_{rg} to classify.

$$M_{ci} = \begin{cases} 1, & \text{if } M_{rg} \geq T_{rg} \\ 0, & \text{otherwise} \end{cases}$$

Final gradient classifier M^k can be written as:

$$M_k = \begin{cases} 1, \text{if } M_c = 1 or M_i = 1 or M_{ci} = 1 \\ 0, \text{otherwise} \end{cases}$$

When implementing,we computed the brightness threshold T_i and the chromatic threshold T_{ci} using OTSU algorithm(with reference) and the window size is chosen to be $(9, 9)$. An example of input image and its reflectance component is shown in Figure . We can observe that the edge details can be better extracted by the auto focus method from the reflectance component than from the input image.

From the Figure 3.2 we can see how effectively this is able to separate out reflectance and shading parts. We can use this shading component to compensate for the scene illumination and get better quality images.

We've studied how blur kernel splits among reflectance and shading components. We did synthetic experiments and the results are shown in the Figure 3.3. We observe that, the blur kernel shrinks in reflectance component compared to the original image.

When we apply focus measure on a stack of frames, we are trying to pick up the frame with the smallest blur kernel. From above observation, we can say that the blur kernel of reflectance component of the sharp frame will be the smallest blur kernel compared to the reflectance component of the other frames.Hence, we can say that the sharpness

(a) Input image

(b) Reflectance image

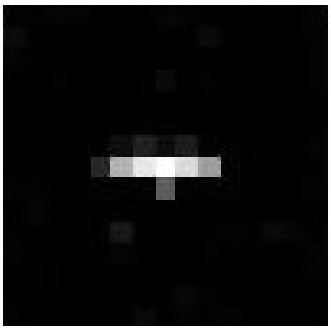
Figure 3.1: Example 1: Input input and reflectance component calculated using the method suggested by Fan *et al.* (2013)

(a) Input image

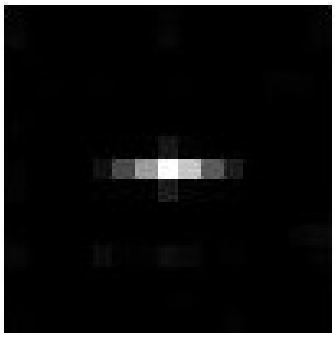
(b) Reflectance Component

(c) Shading component Component

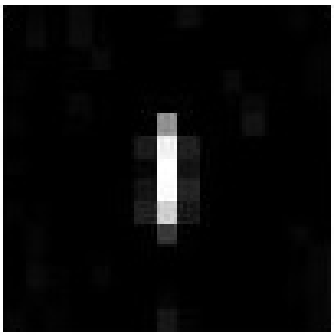
Figure 3.2: Example 2: Input input, reflectance component and shading component calculated using the method suggested by Fan et al



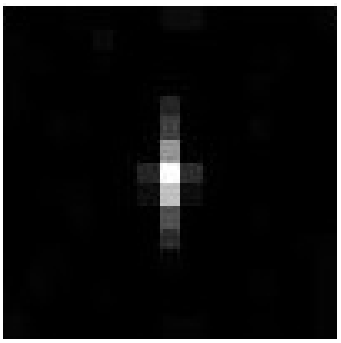
(a) Horizontal blur kernel



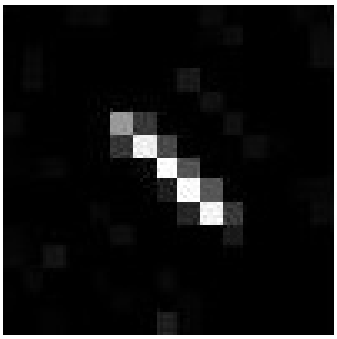
(b) Horizontal blur kernel of reflectance component



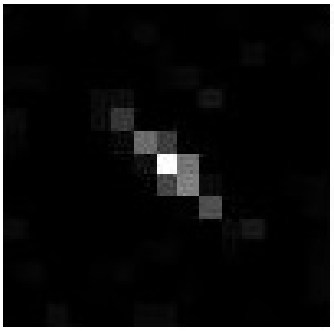
(c) Vertical blur kernel



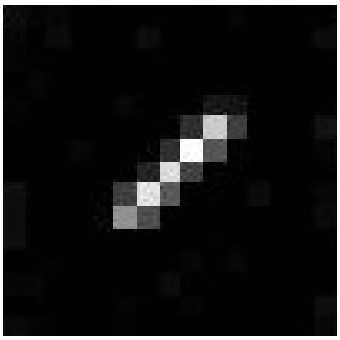
(d) Verticcal blur kernel of reflectance component



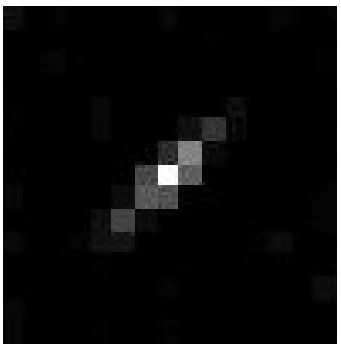
(e) Diagonal blur kernel



(f) Diagonal blur kernel of reflectance component



(g) Diagonal blur kernel



(h) Diagonal blur kernel of reflectance component

Figure 3.3: Study of how blur kernel splits to reflectance component

peak obtained by applying focus operator on reflectance frames will be the same as the sharpness peak in obtained by applying focus operator on original frames.

CHAPTER 4

FOCUS MEASURES

Focus measure is a numerical measure of the sharpness of an image in mathematics, and indicates the degree of focus. A number of focus measures have been studied and compared in the past. Each of them gives an ideal curve with a well defined peak corresponding to the best focused frame. However, frames obtained in low light conditions possesses a small contrast value,varied illumination across the frame, which may be easily influenced by noise and focus measures may result in unpredictable sharpness curves with many local peaks.

Bilen *et al.* (2012) broadly divided these measures into three main categories: histogram, differentiation,and statistics-based focus measures. Some of the histogram based measures are Entropy which is spatial domain measure, fast Fourier transform (FFT), discrete cosine transform (DCT) which are frequency domain measures. Differentiation based methods are Normalized variance, Energy of Laplacian (EOL), Sum Modified Laplacian (SML) and Tenengrad/Sobel.

There are a lot of focus measures used in practise. Most of the statistics based focus measures are specific to the application. For example Tian *et al.* (2011) proposed a focus measure depending on image gradient statistics to enhance the perception of scene. These focus measures have high computational complexity. Some of the most prominently used focus measures in this area area discussed below.

4.1 Normalized Variance

Variance is the simple and most effective focus measure compared to most other focus measures. For a patch of size $[M, N]$ of a gray scale image f, variance can be computed

as:

$$Variance = \frac{1}{MN} \sum_{x=1}^{x=M} \sum_{y=1}^{y=N} (f(x,y) - \bar{f})$$

where $\bar{f}$ is the average gray level over the $[M, N]$ patch, computed as

$$\bar{f} = \frac{1}{MN} \sum_{x=1}^{x=M} \sum_{y=1}^{y=N} (f(x,y))$$

4.2 Sum Modified Laplacian

One of the ways to get the high frequency details is to find out its second derivative. Sometimes, second derivatives can be negative and may cancel each other, so we sum absolute values of each second derivative and it is called Modified Laplacian(ML). Finally, Sum Modified Laplacian(SML) can be computed by summing up all the values of Modified Laplacian in a window.

$$SML = \sum_{x=1}^{x=M} \sum_{y=1}^{y=N} (|ML(f(x,y))|)^2$$

$$ML(f(x,y)) = \left|\frac{\delta^2 f(x,y)}{\delta x^2}\right| + \left|\frac{\delta^2 f(x,y)}{\delta y^2}\right| = |f_{xx}| + |f_{yy}|$$

$$\left|\frac{\delta^2 f(x,y)}{\delta x^2}\right| = |2f(x,y) - f(x-1,y) - f(x+1,y)|$$

$$\left|\frac{\delta^2 f(x,y)}{\delta y^2}\right| = |2f(x,y) - f(x,y-1) - f(x,y+1)|$$

4.3 Energy of Gradient

Energy of gradient(EOG) is computed by finding the energy of the gradient terms in the following manner.

$$EOG = \sum_{x=1}^{x=M} \sum_{y=1}^{y=N} (\frac{\delta f(x,y)}{\delta x})^2 + (\frac{\delta f(x,y)}{\delta y})^2 = \sum_{x=1}^{x=M} \sum_{y=1}^{y=N} (f_x(x,y)^2 + f_y(x,y)^2)$$

where $f_x(x, y)$ and $f_y(x, y)$ are horizontal and vertical gradients of the image f, computed as

$$f_x(x, y) = f(x + 1, y) - f(x, y)$$

$$f_y(x, y) = f(x, y + 1) - f(x, y)$$

4.4 Energy of Laplacian

Energy of Laplacian(EOL) is similar to SML operator, but the convolution operator is different.

$$EOL = \sum_{x=1}^{x=M} \sum_{y=1}^{y=N} \nabla^2 f(x, y)$$

where, $\nabla^2 f(x, y) = (f_{xx} + f_{yy})^2$,
fxx, fyy are double derivatives, same as defined in SML.

4.5 Tenengrad

$$Tenengrad = \sum_{x=1}^{x=M} \sum_{y=1}^{y=N} [\nabla S(x, y)]^2$$

where $\nabla S(x, y) = [\nabla S_x(x, y)^2 + \nabla S_y(x, y)^2]^{\frac{1}{2}}$ and ∇S_x and ∇S_y are gradients along x and y using Sobel mask. They can be computed by

$$\begin{aligned}\nabla S_x(x, y) =& (f(x + 1, y - 1) + 2f(x + 1, y) + f(x + 1, y + 1)) \\ & -(f(x - 1, y - 1) + 2f(x - 1, y) + f(x - 1, y + 1))\end{aligned}$$

$$\begin{aligned}\nabla S_y(x, y) =& (f(x - 1, y + 1) + 2f(x, y + 1) + f(x + 1, y + 1)) \\ & -(f(x - 1, y - 1) + 2f(x, y - 1) + f(x + 1, y - 1))\end{aligned}$$

However, most of these focal measures fail to pick up the sharp frame in low illumination conditions. In the past (Xu *et al.*, 2014) tried to compare various focal measures in low contrast images. Their experimental results claimed that Tenengrad yielded better performances than most others. However, they suggested that it is necessary to have

a more elaborate method because both these fail to generate a single peak in the sharpness curve in some cases.

Since, some focus measures work better under certain conditions and others pick up wrong frame, we tried to use multiple focus measures and use the information given by multiple focus measures to eliminate false alarms. We chose the above explained five focus measures. Each focus measure picks up one frame, we pick up one single frame, from the all the information we've using Tensor Voting which will be discussed in next chapter.

CHAPTER 5

TENSOR VOTING

Primarily, there are three parameters in auto focus model. Firstly, type of focus measure to use. We've decided to use 5 focus measures that are mentioned in Chapter 4. Next is Focus region selection. We either concentrate on the center of the image or anywhere else where user points to. Instead of running the focus measure around the centre of the region, we proposed to run the focus measure around multiple pixels, to remove false alarms. One key parameter is size of the window. If the size of window is too large, there might be illumination changes and we might end up being wrong. If the size of window is too small, it might be too noisy. We've studied on what window sizes can be taken and used multiple window sizes.

Now, for each focus measure, and for each window size, around a fixed point, one frame is picked using sharpness plot. We will have multiple frames picked by different focus operators, with different window sizes around different pixels. To pick one frame from all the information we have,we used Tensor Voting.

Tensor Voting sees how well the frames picked by neighbouring pixels can fit into a 3-D scene. Once we run multiple focus operators, with multiple window sizes around multiple pixels, we've generated frame cloud that contains stack of 3-D tokens of form $[x, y, frame\ index]$, where (x, y) is the pixel around which focus operator is run and $frame\ index$ is the frame picked at this pixel. Then we run the Tensor Voting code, suggested in Hariharan and Rajagopalan (2012) to pick up one frame. The pseudo code of the algorithm proposed in their paper is mentioned below.

From the observations, we've realized the following observations: If illumination is not varying, Tensor Voting picks up the same frame except where there are no details. As illumination varies, multiple focus operators are picking different frames for different window sizes, which implies the necessity of Tensor Voting.

Result: Tensor Voting
T = eye(3,3)
(Initialize T to identity matrices before voting)
for every token t= [x,y, f] from the frame cloud do
(Get the set of neighbouring tokens N(t) on the neighbourhood defined by σ)
for every token t_1 in $N(t)$ do
$v = t_1 - t$
$vote(t1, t) = e^{(-\frac{v^2}{\sigma^2})}(eye(3,3) - (\frac{vv^T}{|v^T v|}))$
(Get second-order vote)
$T(t) = T(t) + vote(t_1, t)$
Voting analysis and final depth estimates
for every pixel(x,y) do
for every token $t = [x, y, f]$ do
Eigen decompose $T(t)$, sort Eigen values λ
Compute surface saliency as λ_1 - λ_2
$\bar{f} = argmax_d(surfacesaliency(x, y, f))$
(Final frame estimate $\bar{f}(x, y)$ at pixel (x, y) is the estimate with maximum surface saliency).

Algorithm 1: Tensor Voting Pseudo Code by Hariharan and Rajagopalan (2012)

We initially chose the following window sizes:
$[(3,3),(5,5),(7,7),(11,11),(13,13),(15,15),(17,17),(21,21)]$. We've divided the entire region into N sub regions and ran the focus operator in each of these regions with the above mentioned window sizes. We've realized that focus operators with small window lengths $[(3,3),(5,5)]$ have failed miserably in picking up the sharp frame due to noise. And frames picked by $(15,15)$ window sizes are same as frame picked by window sizes greater than this. We finally chose the window sizes to be:
$[(5,5),(7,7),(11,11),(13,13),(15,15),(17,17)]$. We chose the value of N to be 16 or 36 or 64 and 100. With increase in value of N, we've seen more accurate results, but computational complexity and time increased. So, we had a trade off between the number of sub regions and time complexity.

CHAPTER 6

EXPERIMENTAL RESULTS

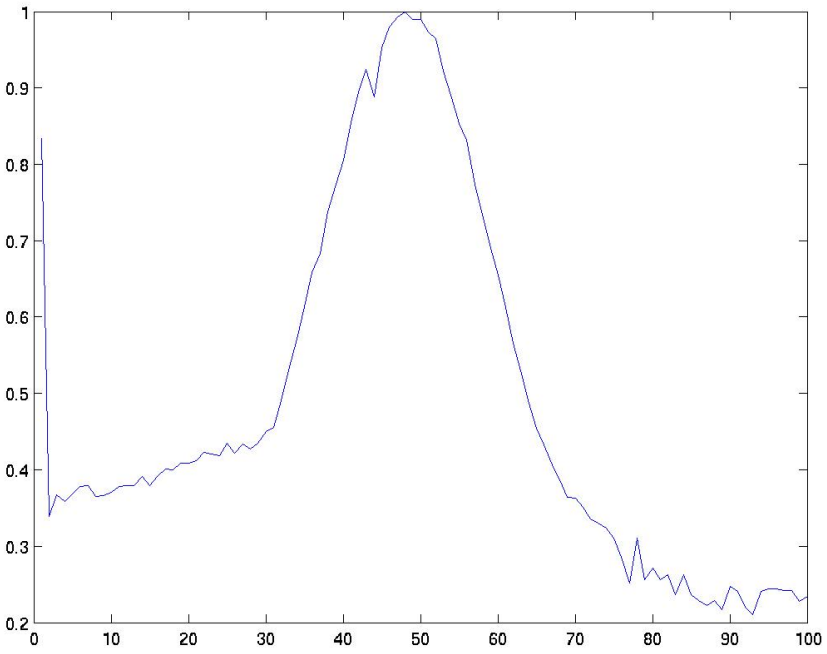
We chose the number of regions, N to be 16 and the window sizes are varied among: $[(5,5),(7,7),(11,11),(13,13),(15,15),(17,17)]$.

Example 1: The scene when captured in good illumination set the lens positions such that frame no 48 is the sharp frame.This serves as a ground truth. The sharpness plot is shown in Figure 6.1a and the frame is shown in Figure 6.1b. When the same scene is captured in bad illumination, from Figure 6.2a, we can see that frame 79 is picked as sharp frame, which is shown in Figure 6.2b.

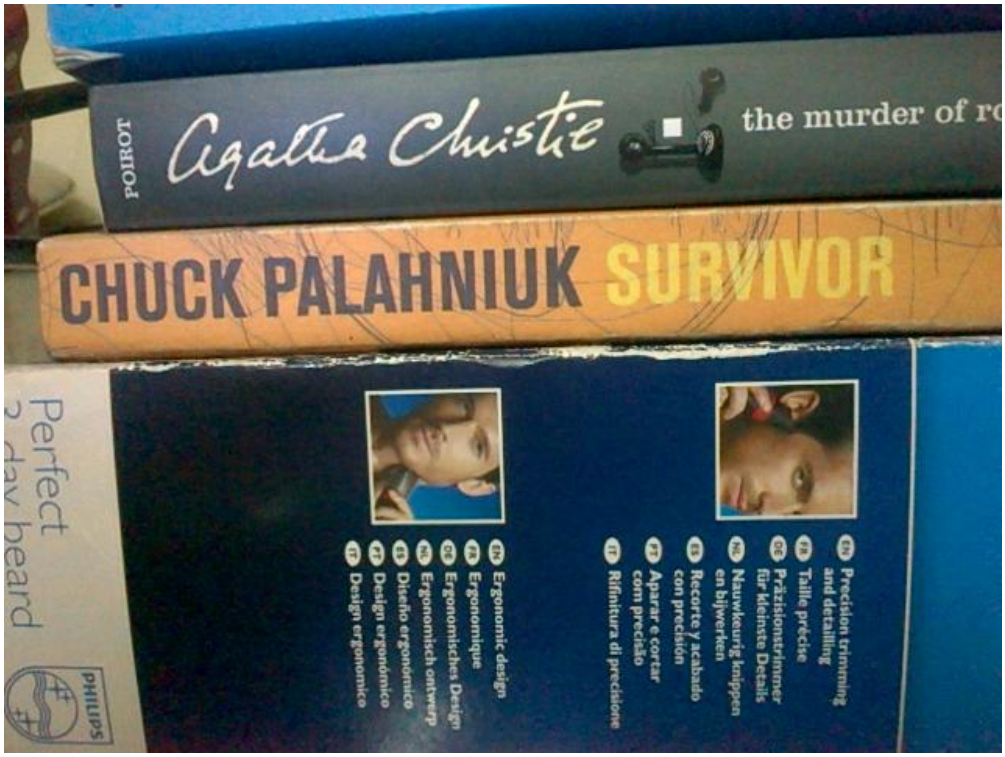
After applying Retinex theory and applying SML on reflectance stack, the sharpness plot is shown in Figure 6.3a, which picks up frame 52. Most focus measures picked up frame 52 and few picked other frames. Using Tensor Voting, we were able to eliminate the false alarms and chose the frame as 52.

To validate our results, we used deconvolution code Punnappurath *et al.* (2014) and showed the blur kernel of the frame picked by conventional method in Figure 6.4a and the our method in Figure 6.4b. We can see that the frame picked by our method has small blur compared to the frame picked by conventional method.

Example 2: In figure 6.5, we have shown the figure corresponding to the frame picked by the conventional method in 6.5a and by our method in the Figure 6.5b. Using the shading component, we did illumination compensation and the final result is shown in 6.5c.

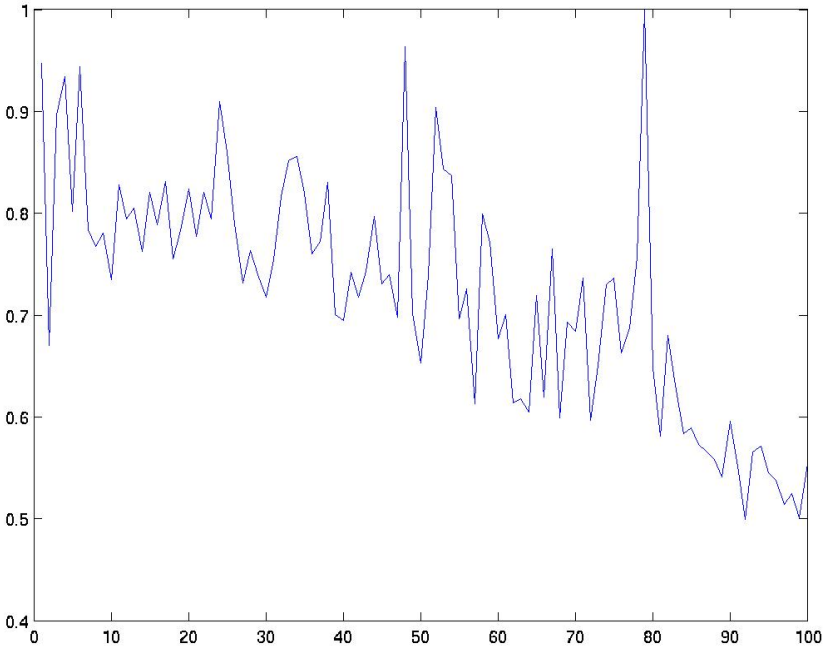


(a) Normalized Sharpness Curve in good illumination

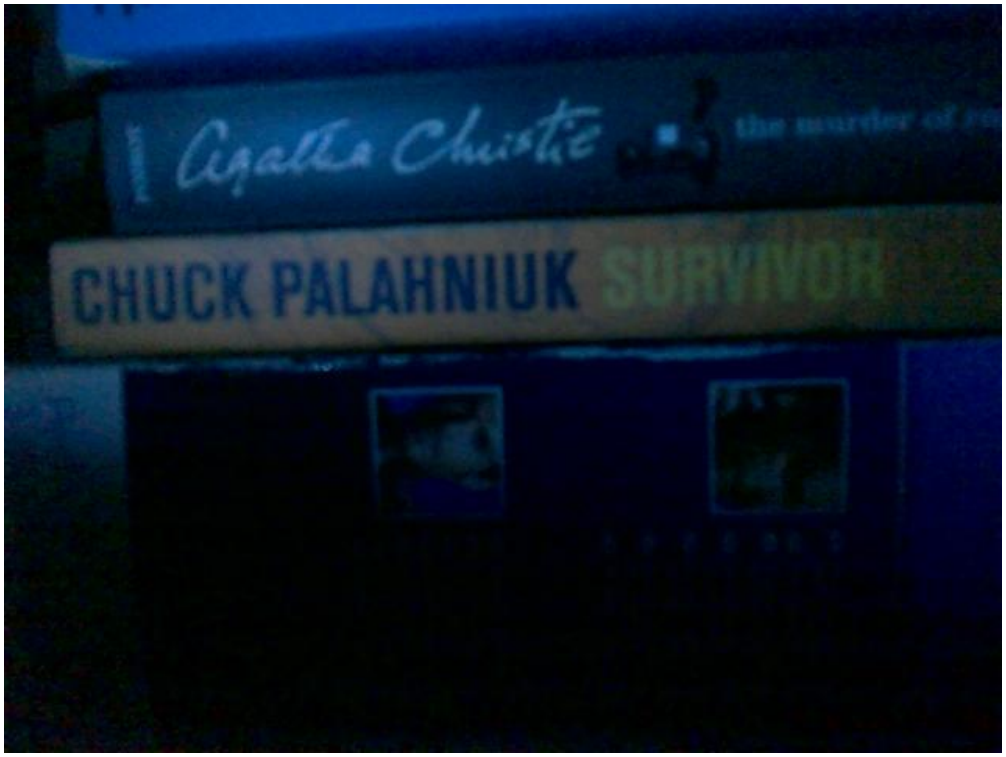


(b) Sharp Frame(48) that was picked in good illumination

Figure 6.1: Example 1:Sharpness plot and the corresponding sharp frame in good illimination are shown in 6.1a and 6.1b

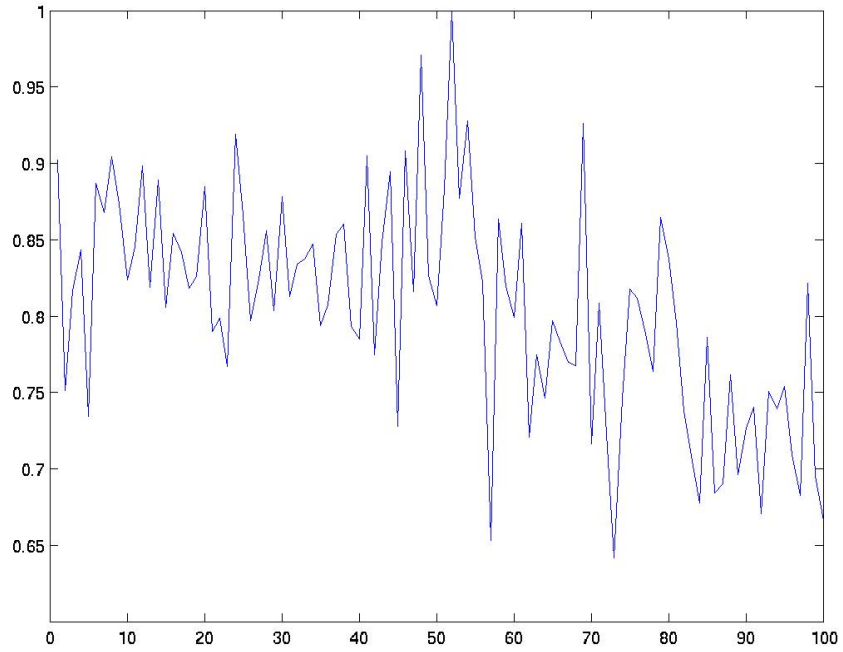


(a) Normalized sharpness plot in bad illumination

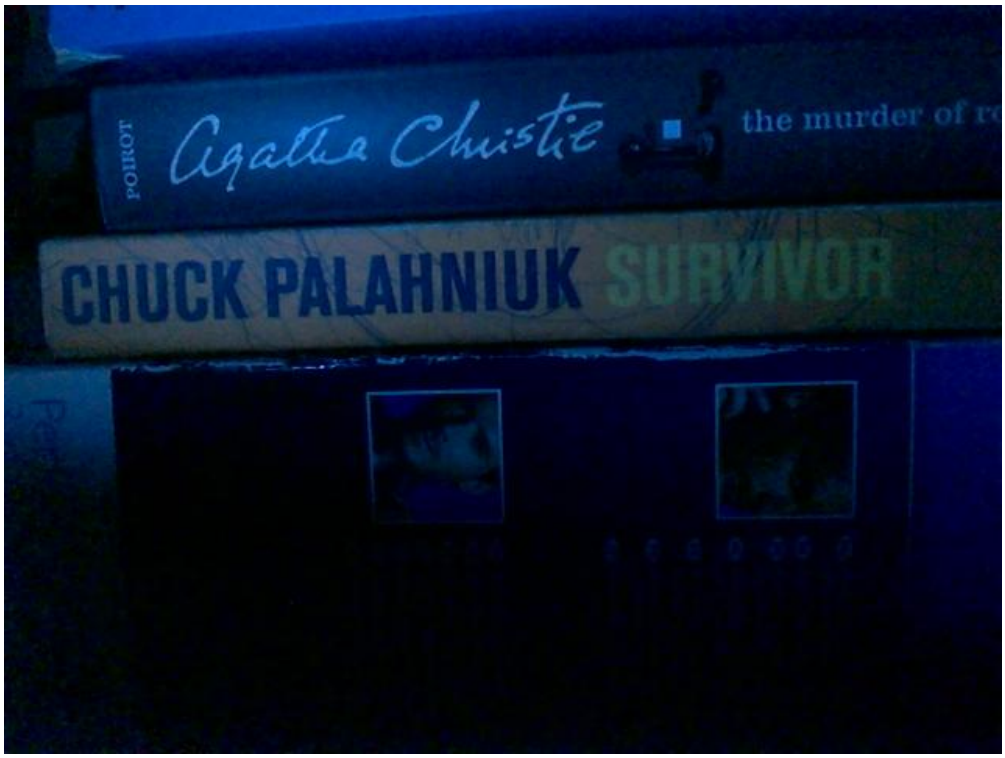


(b) Frame 79 is picked as sharp frame by SML in bad illumination

Figure 6.2: Example 1:Sharpness plot and the corresponding sharp frame picked by the conventional method in bad illumination is shown in 6.2a and 6.2b

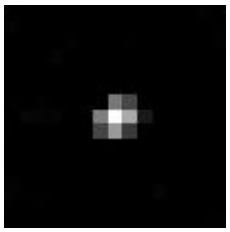


(a) Normalized sharpness plot of reflectance stack when SML is applied

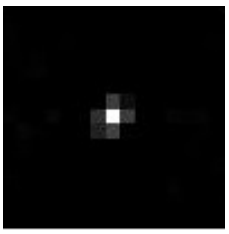


(b) Frame 52 is picked as sharp frame by SML on reflectance stack

Figure 6.3: Example 1:The sharpness plot of SML focus measure applied on reflectance stack is shown in 6.3a and the sharp frame picked by our method is shown in 6.3b

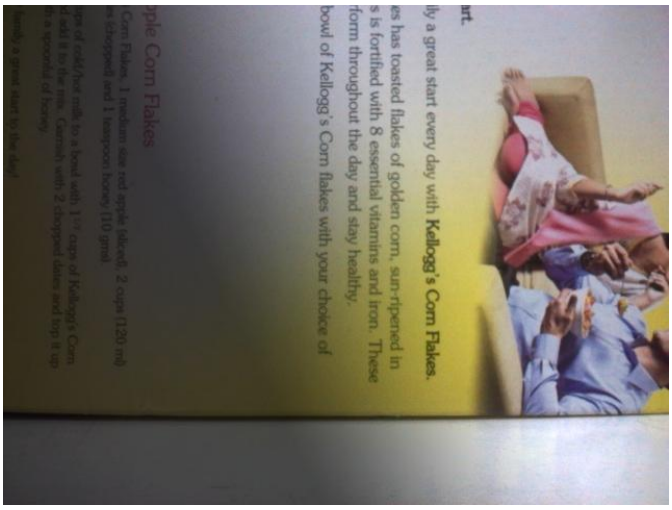


(a) Blur kernel of frame picked by conventional method

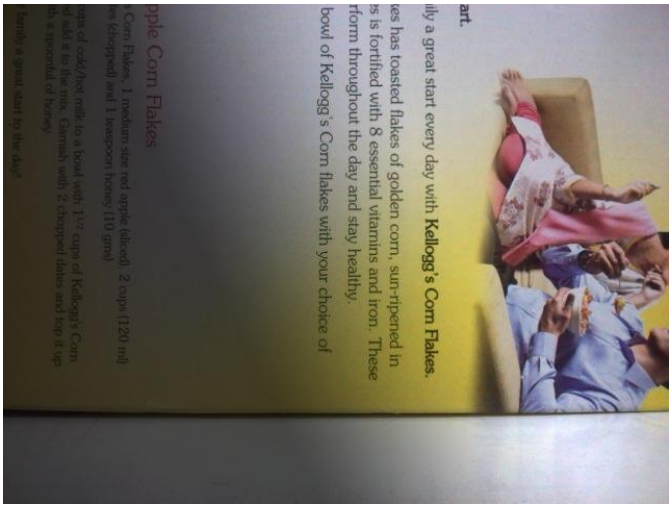


(b) Blur kernel of frame picked by our method

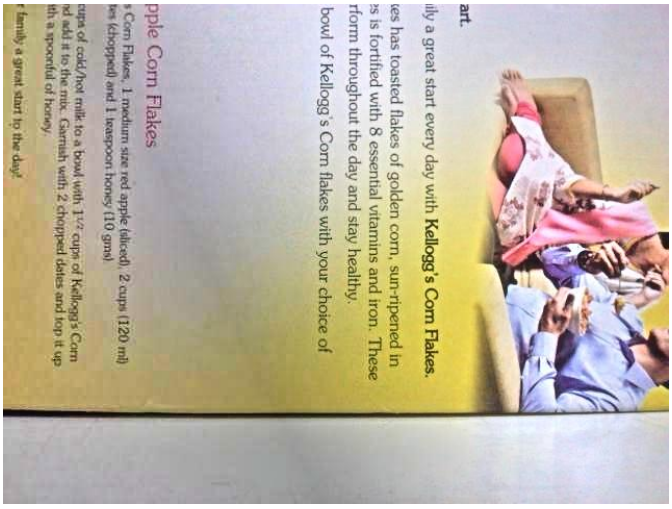
Figure 6.4: Example 1: PSF of frame picked by conventional method is shown in 6.4a and PSF of frame picked by our method is shown in 6.4b



(a) Frame picked by conventional method



(b) Frame picked by our method



(c) Illumination compensated image

Figure 6.5: Example 2: The frame picked by conventional method is in 6.5a and the frame picked by our method is shown in 6.5b and the illumination compensated image is shown in 6.5c

CHAPTER 7

CONCLUSIONS

In this work, we've proposed a new way of handling auto focus method, specially for low illumination conditions. We've shown that frame picked by this method has small blur compared to the frame picked by the conventional method. We can use the shading component and compensation for the scene illumination for the images that are captured in the same scene.

Many applications like mosaicing, viola face zones fails in bad illumination, we can use this method and find out the edge details and enhance their performance. This method works for certain illumination conditions and doesn't work for other illumination conditions. The frame picked up by this method depends on how effectively Retinex is able to get the edge details that are not present at all or that are present in dark regions. This depends on the thresholds used to obtain reflectance component. So, picking up the correct frame depends on the thresholds used. If we find the threshold values from the image itself, we might get this method work in other cases as well.

Also, because the lens is moving from one end to the other end, the focus region does not remain same across all the images in the stack. We can do a simple registration before applying the focus operators and find the sharpness in the same window across all the frames.

REFERENCES

1. **Bilen, H.**, **M. A. Hocaoglu**, **M. Unel**, and **A. Sabanovic** (2012). Developing robust vision modules for microsystems applications. *Mach. Vis. Appl.*, **23**(1), 25–42. URL http://dblp.uni-trier.de/db/journals/mva/mva23.html#BilenHUS12.

2. **Chern**, **Neow**, and **A. Marcelo**, Practical issues in pixel-based autofocusing for machine vision. *In ICRA*. IEEE, 2001. ISBN 0-7803-6578-X. URL http://dblp.uni-trier.de/db/conf/icra/icra2001.html#NgPA01.

3. **Fan**, **H. Zhu**, **G. Lin**, and **L. Cao** (2013). Deriving reflectance and shading components from a single image. *Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*.

4. **Gamadia**, **N. Kehtarnavaz**, and **K. Roberts-Hoffman** (2007). Low-light auto-focus enhancement for digital and cell-phone camera image pipelines. *IEEE Trans. on Consum. Electron.*, **53**(2), 249–257. ISSN 0098-3063. URL http://dx.doi.org/10.1109/TCE.2007.381682.

5. **Groen**, **Young**, and **Ligthart** (1985). A comparison of different autofocus algorithms.

6. **Hariharan, R.** and **A. N. Rajagopalan** (2012). Shape-from-focus by tensor voting. *IEEE Transactions on Image Processing*, **21**(7), 3323–3328. URL http://dblp.uni-trier.de/db/journals/tip/tip21.html#HariharanR12.

7. **Kehtarnavaz, N.** and **H. Joon** (2003). Development and real-time implementation of a rule-based auto-focus algorithm. *Real-Time Imaging*, **9**(3), 197–203. URL http://dblp.uni-trier.de/db/journals/rti/rti9.html#KehtarnavazO03.

8. **Land, E. H.**, **John**, and **J. Mccann** (1971). Lightness and retinex theory. *Journal of the Optical Society of America*, 1–11.

9. **Lee**, **Yogendera**, **Cho**, **S. W. Lee**, and **S. W. Kim** (2008). Enhanced autofocus algorithm using robust focus measure and fuzzy reasoning. *IEEE Trans. Circuits Syst. Video Techn.*, **18**(9), 1237–1246. URL http://dblp.uni-trier.de/db/journals/tcsv/tcsv18.html#LeeKCLK08.

10. **Nayar, S.** and **Y. Nakagawa** (Aug 1994). Shape from focus. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **16**(8), 824–831. ISSN 0162-8828.

11. **Network, M. D.** (2013–). Windows phone samples: learn through code. URL http://code.msdn.microsoft.com/wpapps/.

12. **Ni**, **Wei**, **Yuan**, and **Wu** (2009). Efficient auto-focus algorithm for optical measurement system.

13. **Otsu, N.** (1979). A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, **9**(1), 62–66. URL http://dx.doi.org/10.1109/TSMC.1979.4310076.

14. **Peddigari**, **Gamadia**, and **Kehtarnavaz** (2005). Real-time implementation issues in passive automatic focusing for digital still cameras.

15. **Punnappurath, A.**, **A. N. Rajagopalan**, and **G. Seetharaman**, Blind restoration of aerial imagery degraded by spatially varying motion blur. *In Society of Photographic Instrumentation Engineers, 2014. Proceedings SPIE'14*. 2014.

16. **Shih** (2007). Autofocus survey: a comparison of algorithms.

17. **Tian**, **Chen**, **Ma**, and **Yu** (2011). Multi-focus image fusion using a bilateral gradient-based sharpness criterion. *Opt Commun*.

18. **Vishwanath** (2014). Utilizing motion sensor data for some image processing applications. URL https://github.com/vishwa91/btp.

19. **Xu**, **Wang**, **Zhang**, **Li**, **Liu**, **Xiaofeng**, and **Tang** (2014). A comparison of contrast measurements in passive autofocus systems for low contrast images. *Multimedia Tools Appl.*, **69**(1), 139–156. URL http://dx.doi.org/10.1007/s11042-012-1194-x.