Fall 2010

Detection and estimation of image blur

Harish Narayanan Ramakrishnan

DETECTION AND ESTIMATION OF IMAGE BLUR

by

HARISH NARAYANAN RAMAKRISHNAN

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2010

Approved by

Sanjeev Agarwal, Advisor
Y. Rosa Zheng
Steven L. Grant

ABSTRACT

The airborne imagery consisting of infrared (IR) and multispectral (MSI) images collected in 2009 under airborne mine and minefield detection program by Night Vision and Electronic Sensors Directorate (NVESD) was found to be severely blurred due to relative motion between the camera and the object and some of them with defocus blurs due to various reasons. Automated detection of blur due to motion and defocus blurs and the estimation of blur like point spread function for severely degraded images is an important task for processing and detection in such airborne imagery. Although several full reference and reduced reference methods are available in the literature, using no reference methods are desirable because there was no information of the degradation function and the original image data. In this thesis, three no reference algorithms viz. Haar wavelet (HAAR), modified Haar using singular value decomposition (SVD), and intentional blurring pixel difference (IBD) for blur detection are compared and their performance is qualified based on missed detections and false alarms. Three human subjects were chosen to perform subjective testing on randomly selected data sets and the “truth” for each frame was obtained from majority voting. The modified Haar algorithm (SVD) resulted in the least number of missed detections and least number of false alarms. This thesis also evaluates several methods for estimating the point spread function (PSF) of these degraded images. The Auto-correlation function (ACF), Hough transform (Hough) and steer Gaussian filter (SGF) based methods were tested on several synthetically motion blurred images and further validated on naturally blurred images. Statistics of pixel error estimate using these methods were computed based on 8640 artificially blurred image frames.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Dr. Sanjeev Agarwal for his patience, encouragement, motivation and support throughout my MS program. I specifically would like to thank the time he spent in reviewing my thesis multiple times although he was away from the university campus. I am ever grateful to him.

I would like to thank Dr. Y. Rosa Zheng and Dr. Steven L. Grant for serving as my committee members. They have been most generous in accommodating my tight deadlines in their busy schedules. They suggested several constructive changes and rewrites in spite of their busy schedules because of which this thesis obtained the form it is today.

I would like to thank the department secretary Mrs. Regina Kohout for helping with the graduate program policies and departmental deadlines throughout my MS program.

I would like to thank Aditya Khanolkar and Raghavendra Ravikumar for accepting to perform subjective tests and helping me in every way possible when my access to my lab was limited. I would like to thank Yatin Bodas for graciously accepting to perform subjective tests. I would like to thank Soumya De for taking the time to review my thesis for structure and formatting in spite of his busy schedule with his research work.

I would like to thank the Countermine Division of Night Vision and Electronic Sensors Directorate (NVESD) for providing the data and the opportunity to work on this project.

I would also like to thank my parents and my sister for their constant support and encouragement.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

Images in the real world are subject to various forms of degradation during image capture, acquisition, storage, transmission and reproduction [1]. There are several attributes in images including blur, noise, contrast and saturation, which are directly related to above degradations. The task of an automated image quality inspection system is to make a reliable decision on image quality in near real time with minimum human involvement.

In general, image quality measures can be classified into subjective or objective techniques. Subjective evaluation of image quality is costly and time consuming and the outcome of such an experiment would also depend on viewing conditions [3]. Unlike subjective metrics, objective image quality metrics are faster and can produce immediate results [3] while requiring minimum human involvement. Objective techniques can be further classified into full-reference (FR), reduced-reference (RR) and no-reference (NR). In many situations, it is difficult to have a reference image to evaluate the image quality and NR methods for image quality evaluation are highly desired.

NR methods have been studied broadly in two major classes, namely the spatial domain and the frequency domain techniques. Spatial domain techniques concentrate on studying the change in behavior of high contrast sharp features of the image such as edges, corners, and textures that undergo significant changes during image degradation. Frequency domain techniques concentrate on evaluating the statistical features of the power spectrum of the image.

This thesis conducted a careful review of several image quality metrics including the No Reference assessment of blur and noise impacts on Image Quality (NRIQ) [3], Image and Video Quality Assessment (IVQA) [1], Blind Image Quality Assessment [33], Image Quality Assessment (IQA) based on human visual system (HVS) [31], and Image Quality Metrics (IQM) [32].

Selected methods were applied to the blur detection of mid wave infrared (MWIR) and multispectral imagery (MSI) images collected by the Night Vision and Electronics Sensors Directorate (NVESD) in 2009. The modified Haar algorithm incorporating singular value decomposition (SVD) resulted in the least number of missed detections and minimal number of false alarms. The Haar algorithm (HAAR) based detection results were compared with the modified Haar algorithm (SVD) and the intentional blurring pixel difference algorithm (IBD).

1.1 IMAGE BLUR DETECTION

Before attempting to process the image for any useful information, it is critical to ascertain that the image being processed is of good quality. Among several quality attributes including blur, noise, contrast and saturation, image blur deteriorates high frequency contents in the image thereby making the image unworthy for any useful information retrieval. Incorporating image blur detection in a camera will help discard bad image data at the source itself. Image blur detection for general purpose images is a challenging problem. Two primary sources for blur are relative motion between the camera and the object and poor focus. The purpose of a blur detection system is to eliminate missed detections while achieving a low false alarm rate.

For this study, various blur detection algorithms were analyzed [7, 34, 35, 36, 6, 26, 37]. In particular a perceptual no-reference blur metric was studied [4] in which the authors have arrived at a global blur metric by studying the edge blurring in the spatial domain. In this algorithm, all the vertical edges are found in the image and then horizontal start and end indices of each vertical edge are located. The difference between these horizontal locations gives the edge width which is defined as a local blur measure for that particular vertical edge. A global measure is then obtained by averaging all the local blur measures. The success of this algorithm would largely depend on the quality of detected edges and the number of edges detected. The drawback with this method is the presence of noise in the image might be detected as false edges thereby leading to false estimation. Also, this method is likely more suitable for scenes with manmade objects and less effective for natural ones.

Another blur detection technique is an extrema analysis of the image in the spatial domain [6] which is an NR approach. The method performs a raster scan on the image and counts the number of minima. If the number of minima’s are large, then the image is likely to be blurred since the loss of information in edges during an image blur often results in more minima values. The decision is arrived at by studying the statistical features of the histograms of different extremas. However, a training database is required in order to arrive at a reliable decision since the distributions of the extremas are highly dependent on the type of the scene.

A global blur metric based no-reference approach was studied that exploited the information content in the image using the Discrete Cosine Transform (DCT) [5]. Histograms of selected frequency contents in all 8X8 DCT blocks of the image gives an overall measure of the edge sharpness for the image. The blurred images have all of their high frequency DCT coefficients set close to zero irrespective of the content of the image. The number of zeros or near zero values in the histogram gives an estimate of blur in the image. As noted by the authors, this algorithm doesn't work well for over-illuminated and uniform background images due to little contribution of the blocks to the overall estimate of the image blur.

A rule based direct method of blur detection was proposed in [7] based on the Wavelet transform. This scheme uses a Haar Wavelet transform to detect the edge type and sharpness by exploiting the fact that local maxima of a wavelet transform detects the location of irregular features in the image. When the ratio of the sum of Dirac-Structure and Astep-Structure edge points to the total number of edge points in the image is greater than a threshold value, the image is considered to be un-blurred. In [7] the authors also present an algorithm to calculate the blur extent which can be used as a confidence measure to the blur decision. However, the drawback is a hard threshold value was imposed while deciding the edge map of the image. Any ambiguity in the threshold directly affects the decision.

1.2 BLUR PARAMETER ESTIMATION

After determining whether an image is blurred or not it is important to evaluate the image's Point Spread Function (PSF) to estimate the degree of degradation for further image restoration. Depending on the type of degradation, the blur may be parameterized differently. In the case of a motion blurred image this estimate gives the number of pixels in the blur and the blur direction. However, in the case of defocus blur the PSF is parameterized by the radius of the circle of confusion. Due to uncertain, short lived nature of the degradation process and due to the loss of information due to the blurring, the PSF recovery problem is an ill-posed problem [2].

The study of PSF estimation follows different approaches for motion blur and defocus blur images because motion blur and defocus blur have different frequency responses and hence different degradation functions. However, it is to be noted that both motion and out of focus blurs have the same effect of degrading the original images. The linear motion blur gives rise to parallel dominant lines in the frequency domain whereas out of focus blur gives rise to concentric circles in the frequency domain.

The motivation towards identification of the point spread function as mentioned in [8] was driven by inspecting the location of the frequency domain zeros. PSFs for motion and defocus blurs exhibit a unique behavior in that zeroes will become prominent in the frequency domain. Blur identification using spectral nulls that exploit the use of power spectrum and power cepstrum were studied [9]. The bispectrum is not affected by the presence of additive Gaussian noise. The estimation of motion blur is obtained by finding the first prominent local minimum for bispectrum whereas a global maximum estimate is used in case of bicepstrum. However, the drawback as mentioned by the authors is that this method fails to identify blur parameters at low Signal to Noise Ratio (SNR) or when the blur size is small.

Autocorrelation based PSF estimation for motion blurred images was proposed in [10]. This algorithm aims at extracting a blurred edge from the image by evaluating the phase gradient of the image. Then, an autocorrelation matrix is obtained on the extracted block and a contour of the autocorrelation matrix is drawn. The trace of the peak of the autocorrelation matrix is used to arrive at the direction of blur. A first order backward difference equation is evaluated on the trace of the peak of the autocorrelation points from which the number of pixels in blur due to motion can be obtained. A major advantage of this method as mentioned by the authors is that the autocorrelation not only preserves the motion characteristics but also is immune to noise.

Yet another autocorrelation based PSF estimation method was proposed by the authors in [11]. The PSF estimation consists of first determining the motion direction and then the pixel blur. First, an approximation of the image derivative in a particular direction is obtained and then the total intensity of an image in this particular direction is calculated by summing the absolute values of all pixels. The direction in which the total intensity of the absolute values of the image derivative is lowest gives the blur direction. A digital autocorrelation function (ACF) to the image derivative lines in the identified motion direction is now calculated and the average of the ACF is obtained. The distance between the zero shift point and the minimum of the averaged ACF gives the blur extent in terms of number of pixels. However, as mentioned in [11], this method works well only when the relative homogeneity of the PSF is greater than unity. Also, any error in estimating the motion direction will directly affect the blur extent calculation.

The Hough transform based PSF estimation for motion blurred images using the log spectrum of the blurred images was presented in [12]. The authors motivate the idea of identifying the motion direction by arguing that the spectrum of a non-blurred sharp image is isotropic whereas the same for a motion blurred image is anisotropic. Therefore, the spectrum of a motion blurred image is expected to be biased in a direction perpendicular to the direction of blur. The Hough transform is used to obtain the accumulator array from the binarized log spectrum image. The blur direction is obtained by locating the maximum value of this accumulator array. To calculate the blur length, the binarized log spectrum is rotated in the direction opposite to the identified motion blur. Averaging the rotated image along columns and taking the IFFT and locating the first negative value gives the blur length. However, an obvious drawback of this technique is that any error in the estimation of the blur angle would result in erroneous detection of the blur length. Another chance of erroneous result will be due to binarization of the log spectrum of the blurred image based on a threshold value. Additionally, this threshold will be different for different images and therefore a global measure will be needed to arrive at the correct estimate of the blur extent.

A frequency response based motion blur parameter estimation was proposed in [13] which uses a Gabor filter to estimate the angle and a neural network based training method to obtain the blur length. A two dimensional Gabor filter at different orientations is convolved with the log spectrum of the blurred image to obtain the response corresponding to different frequencies and orientations. For each orientation, the L_2 norm of the convolved image is calculated. The angle corresponding to the highest L_2 norm gives the blur direction. Radial basis function based neural network approach was proposed to obtain the blur length. However, training the neural network is a necessary part of this work.

Out of focus blur PSF estimation works in a manner similar to the motion blur case. However, it is to be noted that the frequency domain behavior of a defocus blur is very different from a motion blur. A Circular Hough Transform based defocus blur PSF estimation technique was proposed by [14]. This is a rule based algorithm that takes the log spectrum of the blurred image as input and computes the Circular Hough Transform (CHT). The blur radius R is then calculated by using a Lagrange polynomial mentioned in [14]. Although this algorithm is relatively straightforward and computationally simple, there is a price to be paid. As mentioned in [14], this method works only for noisy images with SNR greater than or equal to 55 dB.

A noise independent out of focus blur PSF estimation was addressed in [15]. The out of focus blur is estimated using an iterative approach that requires two rough estimates of the blur factor. This algorithm approximates the frequency response of the out of focus blur as Bessel function and uses a Wiener filter to restore the degraded image using two different radii. However, to use this method we need to know the type of the degradation function to get a close estimate of the out of focus blur.

A bispectrum based defocus PSF estimation was proposed in [16]. Their work is motivated by locating the zero crossings of the blurred image. However, if the blurred image is also corrupted by noise, then identifying those zero crossing points would be very difficult in the Fourier spectrum. In order to overcome that, this work evaluates the blurred image in the bispectrum transform domain instead of the conventional Fourier transform domain. The advantage of using a bispectrum transformation is that it suppresses signals that do not have more than two moments. Since Gaussian noise doesn’t have more than two moments, this algorithm is immune to images corrupted with Gaussian noise. The bispectrum transform of each 1-D sub-segment is a 2D matrix, so the computational cost is relatively high.

A gradient vector based PSF estimation for out of focus images is presented in [17]. This work is based on calculating the gradient vector’s logarithm amplitude spectrum in the polar domain and then suppressing the gradient vector fluctuation due to noise by using a Mean Filter Variable Window (MFVW). Then the COC radius is estimated based on applying a weighted mean comb filter on the obtained peaks from the gradient vector. The major advantage of this work mentioned by the authors is that the estimation of the out of focus blur radius is very effective even in a high noise environment. However, the drawback also mentioned by the authors is that when the blurring radius is very small in low frequency images, then the noise dominates the minima's thereby resulting in poor detections.

A Wavelet based PSF estimation for out of focus blurred images is presented in [18]. This algorithm studies the sharpness of the sharpest edges in the image by working out a smoothness measure using the Lipschitz exponent. The authors have showed a relationship between the Gaussian noise variance and the magnitude of the Lipschitz exponent and argue that such a relationship depends only on the image blur. So, when the magnitude of Lipschitz exponent is high, Gaussian noise variance will be high and so is the image blur. The estimation proceeds by calculating the center of gravity (CG) of the significant edges of the image. The PSF estimation of this method is applicable for vertical edges only.

1.3 ORGANIZATION OF THESIS

Section 1 presents a detailed literature review of the generalized problem of image quality detection, blur detection techniques and PSF estimation algorithms. Section 2 presents the details of implementation of three different blur detections algorithms, viz. Haar wavelet based blur detection (HAAR), modified blur detection using singular value decomposition (SVD), and intentional blurring pixel difference based blur detection (IBD). Results obtained on a comprehensive test on IR and MSI images are also tabulated in Section 2. In Section 3, point spread function estimation of motion blurred images using the auto-correlation method and Hough transform based methods were studied. The details of the respective algorithms are presented and tests were conducted by artificially blurring images by a given length in a given direction. Section 4 presents the conclusions and suggestions for future work.

2. BLUR DETECTION

2.1 IMAGE DEGRADATION PROCESS

The image degradation process can be expressed using a linear relationship as described in [19] and the following relationship holds good

$$g(x,y) = f(x,y) * h(x,y) + \eta(x,y) \qquad (2.1)$$

where, $g(x,y)$ is the degraded image in spatial domain, $f(x,y)$ is the uncorrupted original image in the spatial domain, $h(x,y)$ is the point spread function that caused the degradation and $\eta(x,y)$ is the additive noise.

In order to develop reliable blur detection it is important to understand the image degradation process. Degradation functions may be due to camera motion or defocus blur. Noise in the image is unavoidable as the sources of noise mostly stem from the imaging process itself and may be due to the imaging process (quantization effects), dark current, and others. For all practical purposes image noise is usually approximated to be white Gaussian.

The noise and degradation function have contradicting effects on the image spectrum. Noise often introduces additive broad band signals in the image data. Most degradation functions have the effect of averaging out the image data and act as a low pass filter.

Blurring happens when each pixel in the image gets spread over the surrounding pixels. This spreading process is more often referred to as a smearing out around the neighboring pixels. Thus, the blurred image now has pixels that are affected due to this smearing process. An image blur is represented as a mathematical convolution between the source image and the point spread function which is known as the blurring kernel.

Motion blur is caused by the relative motion between the object and the camera. The image so obtained contains an integration of pixel intensities that moved during the period of exposure governed by the camera’s shutter speed. A motion blurred image is easily identifiable to human eyes because such an image will be clearly blurred or smeared in the direction of motion. The PSF estimation of the motion blurred image will help identify the number of pixels by which motion happened and the direction of the blur.

Images that are captured due to a poor convergence of light from an object on the image sensor plane results in what is known as out-of-focus or defocus blur. In optics, focus is defined as the point where the light rays that might be coming from a point source will converge. The effect of a defocus blur on an image will cause the pixel intensities to be scattered around its neighbors in a circle. The center of such a circle is called the Center of Confusion (COC). PSF recovery of an image blurred with out-of-focus blur will work on estimating the radius of this COC.

Some examples of motion blurred and focal blurred image is shown in Figure 2.1 and Figure 2.2, respectively.

Figure 2.1 A motion blurred image

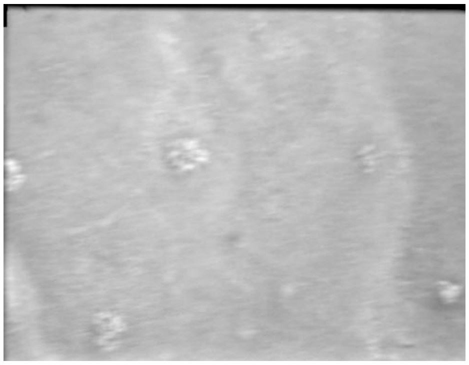


Figure 2.2 A defocus blurred image

2.2 WAVELET APPROACH TO BLUR DETECTION

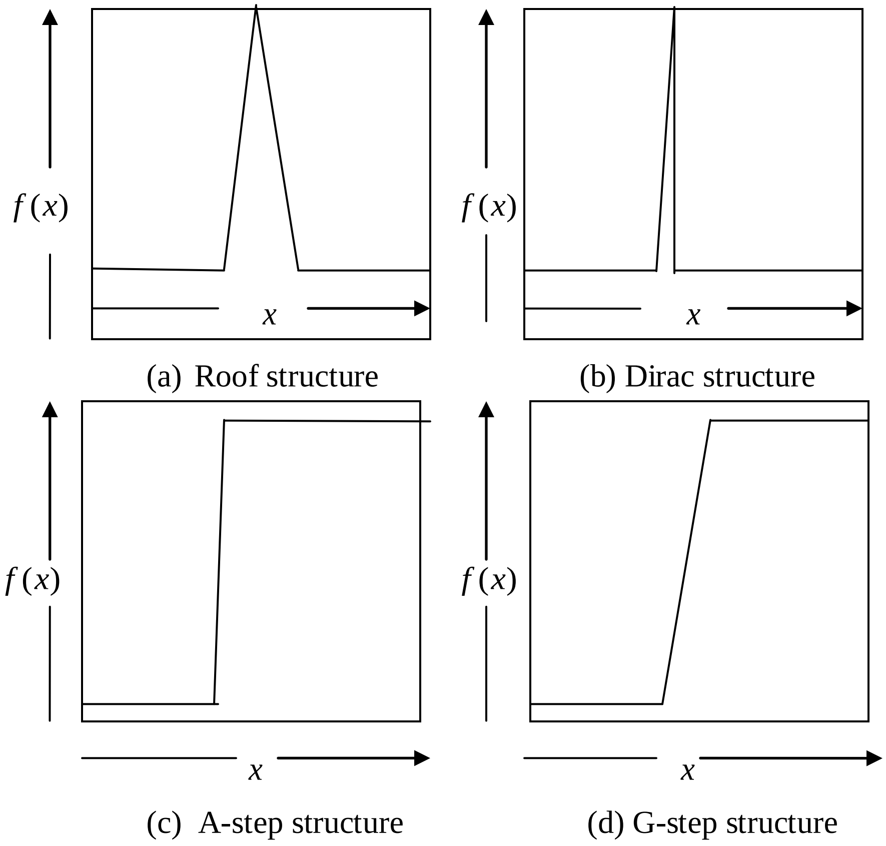
The idea of using a wavelet transform approach for blur detection was published by Tong et al. in [7]. The fundamental principle behind this approach is examining the edge sharpness level to arrive at a decision. Wavelet theory has proved to be an excellent mathematical tool for the analysis of singularities like edges in images and subsequently for edge detection. The ability of the wavelet transform to characterize the local regularity functions is a very important property [20]. In [20], a relationship is obtained between the geometric properties of the edges and Lipschitz exponent in the form of equation (2.2).

$$|f(x_1,y_1)-f(x_2,y_2)| \le K(|x_1-x_2|^2+|y_1-y_2|^2)^{\alpha/2}, |x_1-x_2| \to 0, |y_1-y_2| \to 0 \qquad (2.2)$$

where, $f(x,y)$ is the original image data and (x_1,y_1) and (x_2,y_2) correspond to two different points in the image, α is the edge sharpness factor and K is a numerical constant.

It is shown that the value of exponent α in equation (2.2) plays a critical role in determining the singularity of a signal, and categorizing the edges into three basic geometric structures namely (1) step structure edges with α=0 , (2) roof structure edges with α=1, and (3) dirac structure edges with α= -1. These three basic edges structures have been dealt with in detail with regard to Lipschitz exponent α and wavelet transform [7].

The Step-structure edges are further classified into Astep-structure and Gstep-structure edges based on whether change of pixel intensity is gradual or abrupt. A graphical description of different types of edges as mentioned in [7] is presented in Figure 2.3.



(a) Roof structure (b) Dirac structure

(c) A-step structure (d) G-step structure

Figure 2.3 Different types of edge profiles in typical images

The x-axis in the edge types represents the co-ordinate direction x and the y-axis represents the corresponding edge profile $f(x)$. Gstep and Roof type edge structure have a parameter α (Lipschitz exponent) that is directly proportional to the sharpness level of the edge. In the event of an image blur, irrespective of the fact that the blur was caused due to motion or out of focus, all the Dirac and Astep edges disappear and both the Gstep and Roof type edge structure lose their sharpness.

A simplified overview of the blur detection scheme as presented in [7] is given in the form of a flow chart in Figure 2.4.

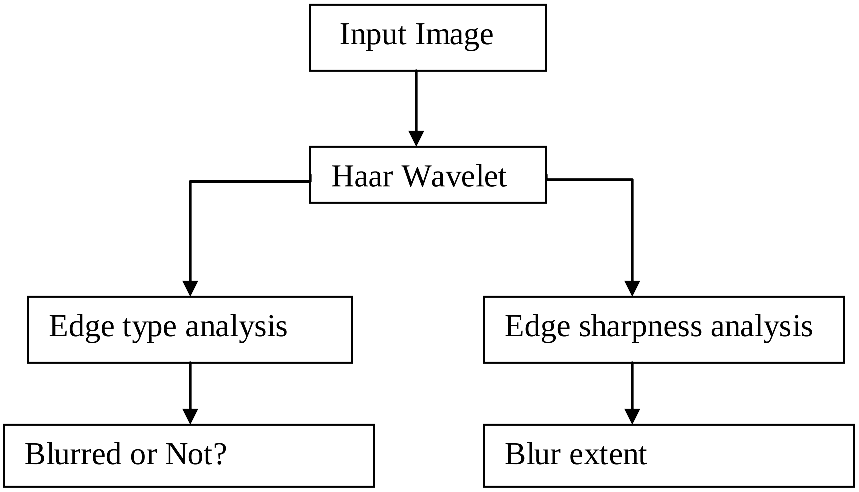


Figure 2.4 Overall view of blur detection using wavelet transform

2.3 BLUR EFFECT ON DIFFERENT TYPES OF EDGES

In equation (2.1), Haar algorithm has assumed the noise term $\eta(x, y)$ can be neglected. Then (2.1) becomes

$$g(x, y) = f(x, y) * h(x, y) \tag{2.3}$$

The effect of blur on different edge structures is given in Table 2.1. The Haar algorithm in [7] has taken advantage of the ability of Haar wavelet transform to distinguish different edge types and also presented a method to recover the sharpness measure of a blurred image.

Table 2.1 Overview of the effect of blur on different edge types

Original	After Blur	Change in α
Dirac Structure	Roof type Structure	N/A
Astep Structure	Gstep Structure	N/A
Gstep Structure	Gstep Structure	Decrease
Roof type Structure	Roof type Structure	Decrease

Without losing focus on image blur detection the next step is to study the characteristics of Haar wavelet which makes it such a powerful tool for image analysis.

2.4 HAAR WAVELET TRANSFORM

Let us consider a 1D image with just 4 pixels [9 7 3 5]. Now, the first low resolution image can be obtained by pair wise averaging giving [8 4]. There is a reduction of 50% image data as evident from the fact that there are only two pixels instead of the original four pixels. Certainly there is loss of information. These two pixels so obtained are called the approximation coefficients of the original image. In order to recover the original four pixels, the detail coefficient needs to be stored in some form. The first detail coefficient is one so that seven is one more than the first approximation coefficient and one less than nine. Similarly, a second detail coefficient value of negative one leads to recovering the third and fourth pixel of the original image. Table 2.2 gives the detail and approximation coefficients for this example.

Table 2.2 Decomposition and approximation/detail coefficients

Resolution scale	Average	Detail coefficients
4	[9 7 3 5]	
2	[8 4]	[1 -1]
1	[6]	[2]

This process of recursively computing the approximation and detail coefficients by the method of averaging and differencing is called the filter bank. The ability to view the image at any given scale is the basis of Haar Wavelet Transform (HWT).

The interest to use wavelet transform for edge detection stems from its ability to perform multi-resolution processing. Multi-resolution in image processing refers to the ability to apply the image into different scales depending upon the level of detail required. The use of the wavelet transform refers to both detail and approximation coefficients. The approximation subsignal gives the general trend of the pixels [23] like mean intensity values whereas the three detail coefficients represent the horizontal, vertical and diagonal details in the subimage.

The motivation towards using wavelet transforms for image processing applications is its ability to preserve the temporal information while performing frequency domain analysis which is however not possible by using the Fourier transform. Haar transform on the other hand is simple and easy to use owing to its processing speed and is also the simplest known orthonormal wavelet [22]. The Haar transform in its basic form can be expressed as

$$\mathbf{T} = \mathbf{WFW} \tag{2.4}$$

where, **F** is the original $N \times N$ image matrix, $\mathbf{W}$ is the $N \times N$ transformation matrix and **T** is the resulting $N \times N$ transform. The Haar function is discontinuous and therefore not differentiable. The Haar function is both separable and symmetric [22]. The Haar basis functions in the most general form is given as [22]

$$w_0(z) = w_{00}(z) = \frac{1}{\sqrt{N}}, z \in [0,1] \tag{2.5}$$

The 2-D Haar basis functions as mentioned in [24] is given by

$$w_{uv}(x, y) = w_u(x) w_v(y) \tag{2.6}$$

$$w_0(i) = w_{0,0}(i) = \frac{1}{\sqrt{N}};\ \text{for } 1 \le i \le N \tag{2.7}$$

$$w_k(i) = w_{p,q}(i) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2}, \frac{q-1}{2^p} N \le i \le \frac{q-1/2}{2^p} N \\ -2^{p/2}, \frac{q-1/2}{2^p} N \le i < \frac{q}{2^p} N \\ 0, \text{ otherwise for } 1 \le i \le N \end{cases} \tag{2.8}$$

$$p, q \text{ such that } k = 2^p + q, \text{ for } 1 \le k \le N \tag{2.9}$$

In the above equation p and q correspond to scale (dilation) and translation in wavelet terms [24]. A simple 2×2 Haar transformation matrix $\mathbf{W}_2$ is given by

$$\mathbf{W}_2 = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \tag{2.10}$$

In general 2D Haar matrix is given by,

$$\mathbf{W} = \begin{pmatrix} w_{0,0} & \cdots & w_{0,N-1} \\ \vdots & \ddots & \vdots \\ w_{N-1,0} & \cdots & w_{N-1,N-1} \end{pmatrix} \tag{2.11}$$

The local statistics of a HWT function are relatively constant and can be easily modeled [22]. Having many values close to zero makes it an excellent candidate for image compression [22]. Both coarse and fine resolution approximation images can be extracted from the DWT image. The HWT has a poor energy compaction but its high spatial localization is an important feature that makes it so widely useful [23]. The next section deals with edge detection using HWT.

2.5 EDGE DETECTION USING HWT

The basis of blur detection proposed by [7] stems from the edge content analysis of the input frame. The HWT is computed on the input image and the decomposition level is set to 3. The outcome of such a transform will lead to a pyramid like structure as shown in Figure 2.5. Computing the HWT with the decomposition level set to three results in three images at three different scales. The next step is to construct the edge map at each scale as given by equation (2.12).

$$Emap_i(k,l) = \sqrt{(LH_i^2(k,l) + HL_i^2(k,l) + HH_i^2(k,l))} \tag{2.12}$$

where $LH_i(k,l)$ refers to horizontal low pass and vertical high pass (vertical information), $HL_i(k,l)$ refers to horizontal high pass and vertical low pass (horizontal information) and $HH_i(k,l)$ refers to horizontal high pass and vertical high pass (diagonal information). Edge maps at the same three scales were constructed based on the above formulation.

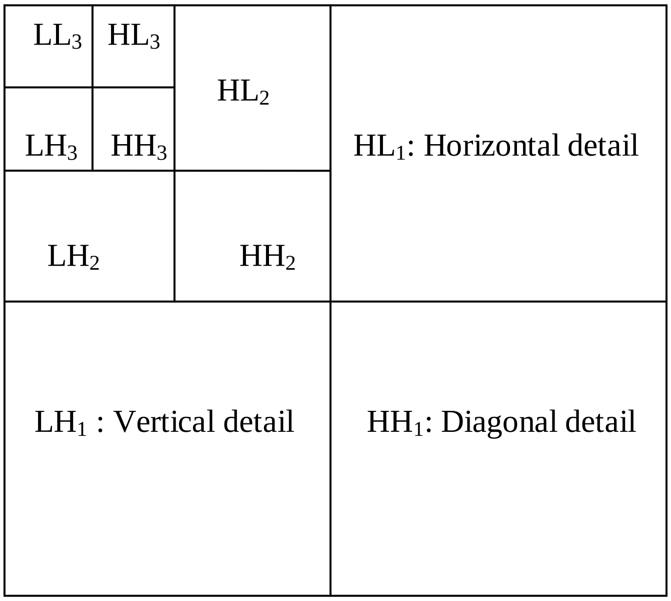


Figure 2.5 Pyramid structure of Haar decomposition

After constructing the edge map, the next step is to partition the edge map and locate the local maxima in each window. For this purpose, three mask sizes were used: 2×2, 4×4 and 8×8. The outcome of such a maxima finding is denoted as $Emax_i$ where "i" denotes the corresponding scale number 1, 2 and 3, respectively. The edge map is partitioned and windowed with the above mentioned mask to arrive at $Emax_i$ image. A non maximum suppression algorithm [30] has been used in this thesis. One can say that the larger the $Emax_i$, the larger the edge strength is. If $Emax_i(k,l) > threshold$, then the location (k,l) is categorized as an edge pixel.

In order to account for the varying image data characteristic, the *threshold* was obtained by convolving each frame of the image data with a Laplacian mask as given in Figure 2.6. Thanks to its second derivative characteristics, the Laplacian mask results in a response that is rotation invariant [22] and hence gives the same response irrespective of the direction of the edge. The threshold was then obtained on each frame by calculating the mean absolute intensity of the convolved image and multiplying by a factor of 5. The blur threshold for the entire data set was then obtained by calculating the median of all such threshold values.

0	-0.25	0
-0.25	1	-0.25
0	-0.25	0

Figure 2.6 Laplacian mask for threshold detection

2.6 HAAR WAVELET BLUR DETECTION ALGORITHM

The algorithm details mentioned in [7] are given below in the form of 5 rules. The 5 rules mentioned below were implemented with certain modifications in threshold calculations.

1. If $Emax_1(k,l) > threshold$ (OR) $Emax_2(k,l) > threshold$ (OR) $Emax_3(k,l) > threshold$, then (k,l) is an edge point.
2. For any edge point (k,l) if $Emax_1(k,l) > Emax_2(k,l) > Emax_3(k,l)$, then (k,l) is a Dirac-Structure or Astep-Structure.
3. For any edge point (k,l), if $Emax_1(k,l) < Emax_2(k,l) < Emax_3(k,l)$ then (k,l), is a Roof-Structure or Gstep-Structure.
4. For any edge point (k,l), if $Emax_2(k,l) > Emax_1(k,l)$ AND $Emax_2(k,l) > Emax_3(k,l)$, (k,l) is a Roof-Structure.
5. For any Gstep-Structure or Roof-Structure edge point (k,l), if $Emax_1(k,l) < threshold$, (k,l) is more likely to be in a blurred image.

Based on the above 5 rules, Haar algorithm for blur detection as presented in [7] is given below.

1. Calculate the edge map and obtain the local maxima on the edge map image at each of the 3 decomposition levels viz. $i = 1,2,3$.

2. Use Rule1 to obtain all the edge points and count the sum total of all such edge points across the 3 decomposition levels. Denote the total number thus obtained as N_{edge}
3. Use Rule2 to find all Dirac-Structure and Astep-Structure edge points. Let N_{da} be the total number of them
4. Use Rule3 and Rule4 to find all Roof-Structure and Gstep-Structure edge points. Let N_{rg} be the total number of them
5. Use Rule5 to find all Roof-Structure and GStep-Structure edge points that have lost their sharpness. Denote the total number thus obtained as N_{brg}
6. Calculate the ratio of Dirac-Structure and Astep-Structure to all the edges

$$Per = N_{da} / N_{edge} \tag{2.13}$$

7. If $Per > MinZero$, judge that the image is un-blurred and vice versa, where $MinZero$ is a positive number close to zero. A $MinZero$ value of 0.05 was used in this work.
8. Calculate how many Roof-Structure and Gstep-Structure edges are blurred and hence obtain an estimate of blur extent given by

$$BlurExtent = N_{brg} / N_{rg} \tag{2.14}$$

A confidence estimate of the blur decision is given by the $BlurExtent$.

2.7 HAAR ALGORITHM ON SHARP AND BLURRED FRAMES

Sample blurred and sharp images used for testing Haar algorithm is shown in Figure 2.7 and Figure 2.8, respectively. The original image size used was 640×512 pixels.

Figure 2.7 Blurred frame

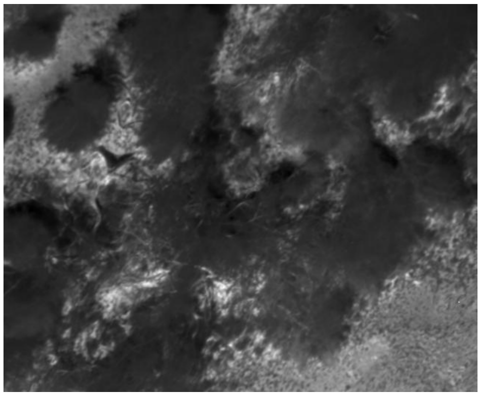


Figure 2.8 Sharp frame

A summary of edges identified by Haar algorithm on each of the images of Figure 2.7 and Figure 2.8 is tabulated in Table 2.3. It can be seen from Table 2.3 that both the total number of edges detected (N_{edge}) and the number of Dirac or Astep structure edges (N_{da}) for the blurred image is less than the sharp image. Also, the blur factor (Per) for the sharp image is much higher than the threshold value of 0.05 thereby classifying the frame in Figure 2.8 as a sharp image. The distribution of Haar blur factor Per for data collection 15_49 IR data is given in Figure 2.9.

The chart in Figure 2.9 conveys that a large number of frames resulted in a Per value between 0.04 – 0.07.Also very few frames resulted in a $Per > 0.07$ which leads to believe that the sharpest frames had a Per value close to 0.07. However, a good number of frames were detected with $Per < 0.01$ thereby leading to believe that the blurriest frames had Per close to 0.

Table 2.3 Haar algorithm detection statistics on sample blurred and sharp frames

Parameter	Blurred image	Sharp image
N_{edge}	11527	14465
N_{da}	528	1122
N_{rg}	11517	14372
N_{brg}	10773	12916
Per	0.0458	0.0776
Classification	Blurred	Not blurred

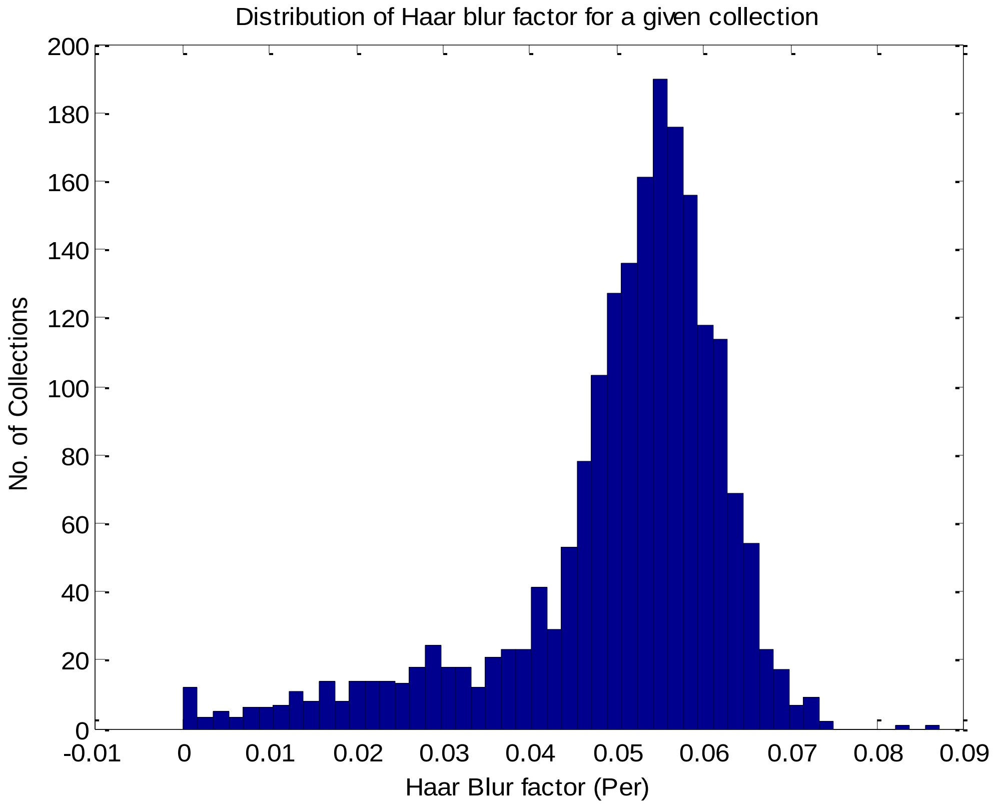


Figure 2.9 Distribution of Haar blur factor (Per)

Some of the sample missed detections and false alarms reported by Haar algorithm are shown in Figures 2.10 and Figure 2.11, respectively. In the case of example missed detections shown in Figure 2.10, it can be seen that there are a number of transitions from high gray level intensity to low gray level intensities thereby resulting in false edge detection and hence resulting in a high value of *Per* thereby reporting a missed detection.

On the other hand, the example images shown in Figure 2.11 have a rather smooth trend throughout the image data. As a result, the number of edges detected is small and thereby resulting in a lower *Per* value and hence reporting false alarms. Missed detections and false alarm statistics using Haar algorithm for Figure 2.10(a) and Figure 2.11(a) are given in Table 2.4.

Table 2.4 Haar algorithm missed detections/false alarms statistics

Parameter	Blurred image	Sharp image
N_{edge}	2293	6521
N_{da}	183	182
N_{rg}	2286	6520
N_{brg}	2062	6373
Per	0.0798	0.0279
Classification	Sharp	Blurred

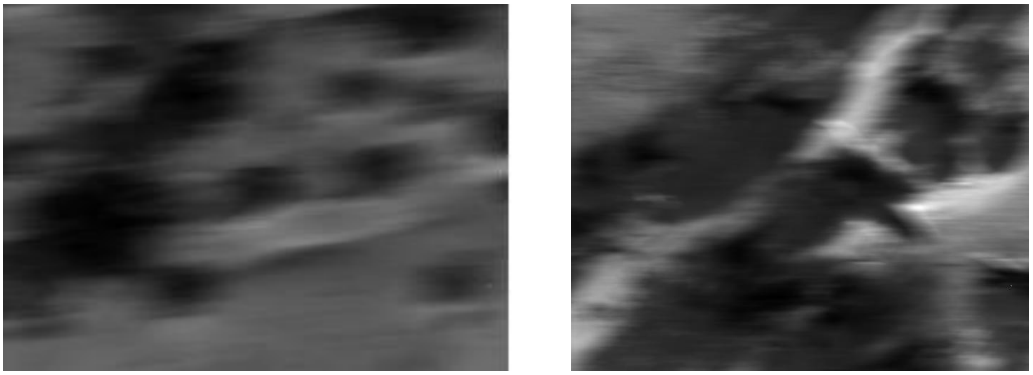


Figure 2.10 Sample missed detections by Haar algorithm

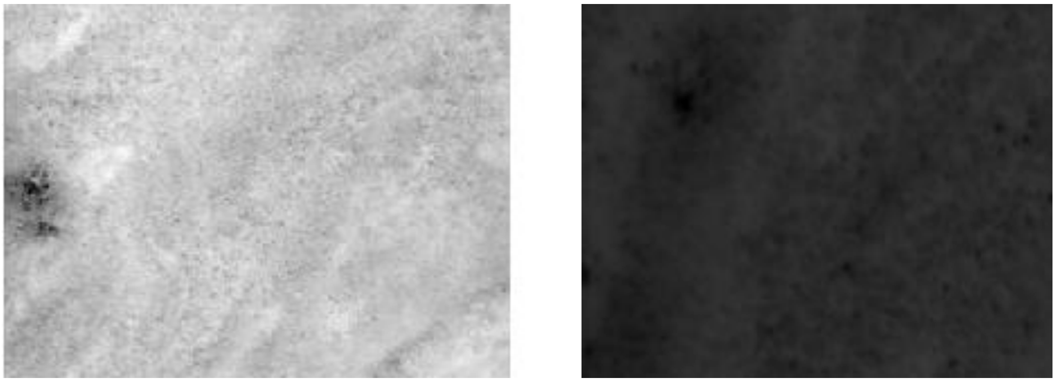


Figure 2.11 Sample false alarms reported by Haar algorithm

2.8 MODIFIED BLUR DETECTION ALGORITHM USING SVD

The Haar algorithm described in Section 2.6 resulted in a number of missed detections and false alarms because the blur decision is heavily dependent on the correct identification of edges. Image data with a smooth trend resulted in false alarms whereas images that were truly blurred but had sudden variation from high detail to low detail resulted in missed detection.

When the background of any given frame is relatively constant with few sharp features, the number of edges detected in such an image will be less. As a result, Haar algorithm reported false alarms in those types of images. Thus, there was a need for improvement in the Haar algorithm and a modified algorithm was developed that exploited the concept of the singular value decomposition (SVD).

Edge detection was performed on the edge map as explained in Section 2.5 to arrive at $Emax_i$ where $i = 1, 2, 3,$ respectively. A second order Laplacian filter mask was convolved with the input frame for the threshold calculation for the edge detection. The Laplacian mask used for the SVD algorithm was the same as the one used for Haar algorithm in Figure 2.6.

The blur factor Per was then calculated by considering only the horizontal and vertical components in the 3rd level Haar decomposition given by $\mathbf{I}_{3x}$ and $\mathbf{I}_{3y}$. All pixels in $Emap_3 > threshold$ were considered strong candidates for edge pixels. The identified edge pixels in $\mathbf{I}_{3x}$ and $\mathbf{I}_{3y}$ are represented as I_{3xedge} and I_{3yedge}, respectively. It is to be noted that I_{3xedge} and I_{3yedge} are column vectors. A 2D matrix was constructed by taking into account only the horizontal and vertical edge locations namely I_{3xedge} and I_{3yedge} and labeled **X**. Mathematically, the matrix **X** can be written as

$$\mathbf{X} = [I_{3xedge} \; I_{3yedge}] \tag{2.15}$$

$$\mathbf{Y} = \left(\sqrt{diag(\frac{1}{I^2_{3xedge} + I^2_{3yedge}})} \right) \times \mathbf{X} \tag{2.16}$$

Y can be decomposed using singular value decomposition to give the vector of singular values, S

$$\mathbf{Y} = \mathbf{U}\mathrm{S}\mathbf{V}^{\mathrm{T}} \tag{2.17}$$

Now, the image sharpness factor is given by

$$factor = \frac{\min(\mathrm{S})}{\max(\mathrm{S})} \tag{2.18}$$

The blur factor obtained is compared with a *blurthreshold* value to make the blur decision on the image. In this thesis, a *blurthreshold* value of 0.73 was used for detecting whether the image is a sharp image or not. The effect of blur decision was tested with different values and with different image types. A distribution of blur factor obtained for the same collection used for Haar algorithm collection 15_49 IR data using SVD algorithm is given in Figure 2.12.

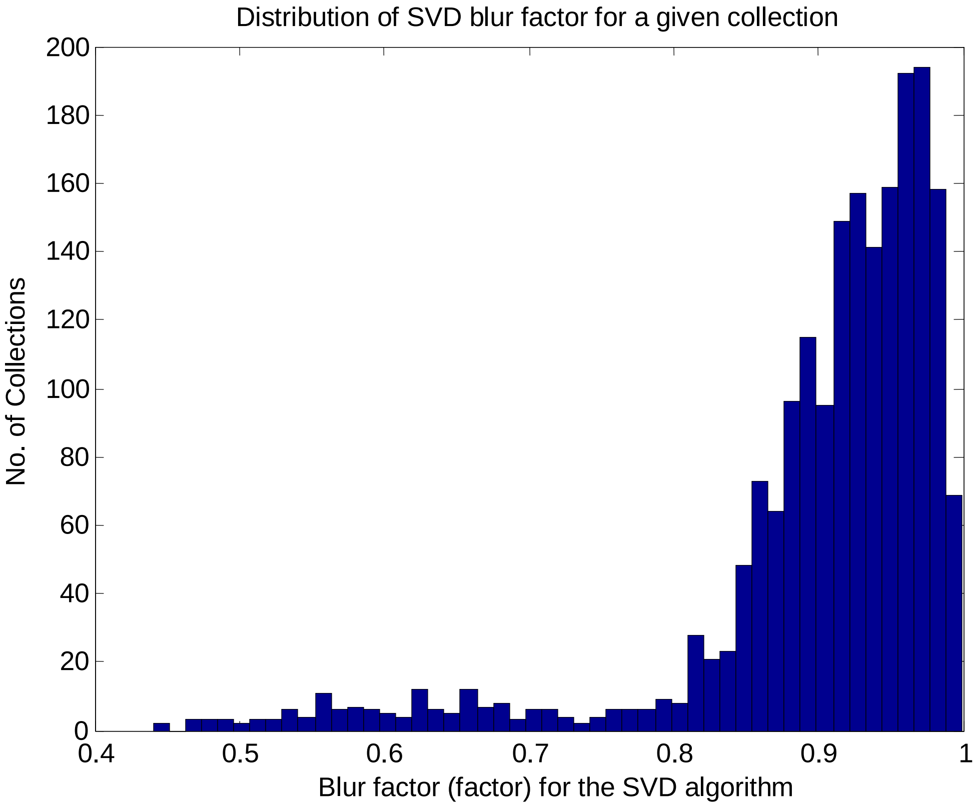


Figure 2.12 Histogram plot of SVD blur factor for a single collection

Test results of the SVD algorithm performed on sample images of Figure 2.7 and Figure 2.8 is given in Table 2.5.

Table 2.5 SVD algorithm detection statistics on test blur and test sharp image

Parameter	Blurred Image	Sharp Image
factor	0.36	0.785
Classification	Blurred	Not blurred

Some of the sample missed detection and false alarms reported by SVD are shown in Figure 2.13 and Figure 2.14, respectively. The SVD missed detection and false alarm statistics for Figure 2.13 and Figure 2.14 (b) are given in Table 2.6.

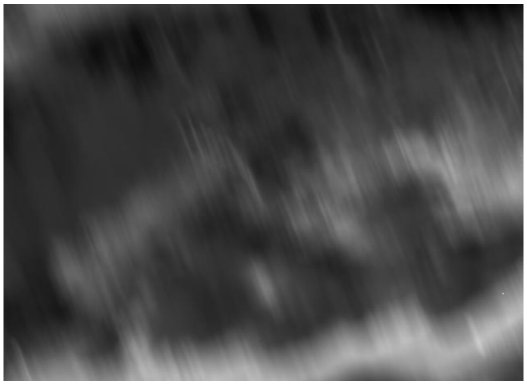


Figure 2.13 Sample missed detection by SVD algorithm

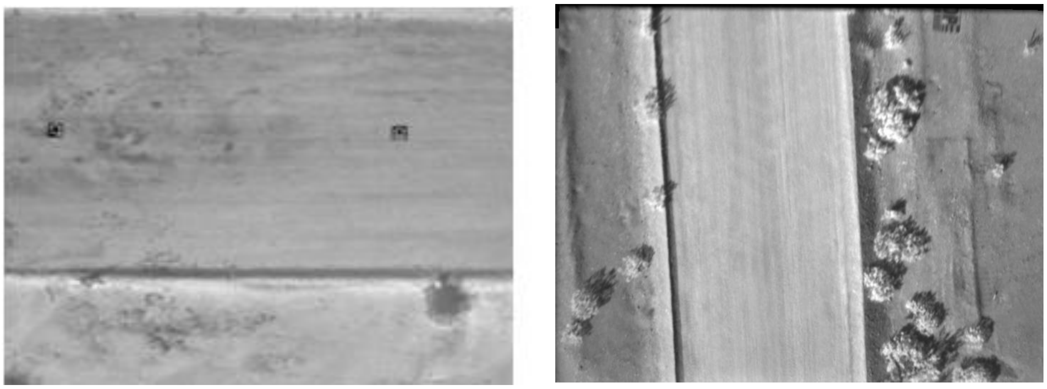


Figure 2.14 Sample false alarms reported by SVD algorithm

Table 2.6 SVD algorithm missed detection and false alarm statistics

Parameter	Blurred Image	Sharp Image
factor	0.9648	0.6982
Classification	Not blurred (Missed detection)	Blurred (False alarm)

Since SVD algorithm inspects the singularities in the image, which are affected on the event of an image blur, SVD resulted in the least number of false alarms and very minimal missed detections. On careful inspection of false alarms reported by the SVD algorithm, it was found that false alarms were due to very low contrast images or images with linear features like the one shown in Figure 2.14. Mathematically, the difference between the minimum of the SVD and the maximum of SVD was found to be very high, as a result of which the ratio of minimum of SVD to the maximum of SVD resulted in a very small value that determined the sharpness factor and thereby classifying the image as blurred.

2.9 NO-REFERENCE PERCEPTUAL BLUR METRIC

Both the Haar and SVD algorithms are dependent on correct identification of the edges. Edge detection involves using a threshold value to determine if a given pixel is blurred or not. Choosing incorrect threshold values will result in poor edge detection resulting in an incorrect blur decision. The motivation to use intentional blurring pixel difference (IBD) algorithm is due to the fact that it doesn’t require the use of edge detection. Another factor in considering IBD algorithm is its computational speed.

As the name states, this algorithm is based on intentional blurring of the given image. The principle stated in [26] is straightforward in that, the intentional blurring of a sharp image will result in huge gray scale variations. On the other hand, intentional blurring of an already blurred image would result only in small gray scale variations. A simplified blur estimation flow chart is given in Figure 2.15.

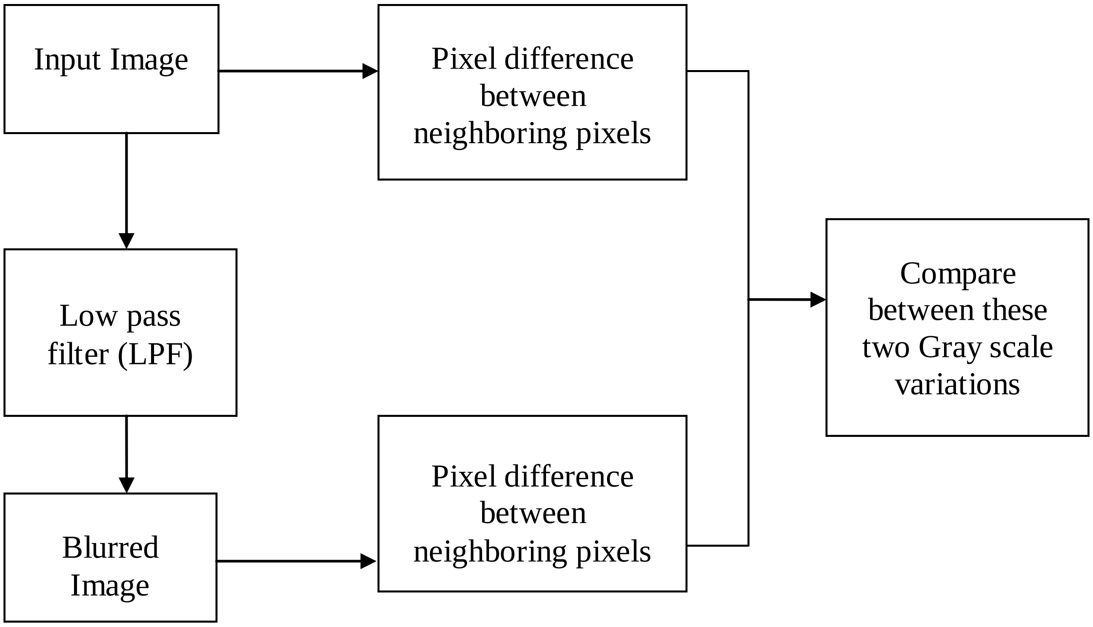

Input Image
Pixel difference between neighboring pixels
Low pass filter (LPF)
Compare between these two Gray scale variations
Blurred Image
Pixel difference between neighboring pixels

Figure 2.15 Block diagram of intentional blurring pixel difference algorithm (IBD)

The estimation proceeds by filtering the original image with horizontal low pass and vertical low pass filters separately. The filter model that was used is given in equations (2.19) and (2.20).

$$h_v = \frac{1}{9} \times [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1] \qquad (2.19)$$

$$h_h = h_v^T \qquad (2.20)$$

The reason for using a bigger filter size is to introduce significant blurring in the given image. In the case of color images, the luminance component is extracted first. This assumption in [26] is justified by the fact that the sharpness of any image is contained in its gray scale components. The filtered image is obtained using

$$B_{ver} = h_v * F \qquad (2.21)$$

$$B_{hor} = h_h * F \qquad (2.22)$$

F is the luminance image obtained from the input image I. The equation to extract the luminance component F for any given RGB image is given by

$$F = 0.299 * R + 0.587 * G + 0.114 * B \qquad (2.23)$$

where, R, G, B are the red, green and blue components of the input image. To study the gray scale variations absolute difference between the neighboring pixels are computed that are given below.

$$D\_F_{ver}(i,j) = abs(F(i,j) - F(i-1,j))\ for\ i = 1\ to\ m-1,\ j = 0\ to\ n-1 \qquad (2.24)$$

$$D\_F_{hor}(i,j) = abs(F(i,j) - F(i,j-1))\ for\ i = 0\ to\ m-1,\ j = 1\ to\ n-1 \qquad (2.25)$$

$$D\_B_{ver}(i,j) = abs(B(i,j) - B(i-1,j))\ for\ i = 1\ to\ m-1,\ j = 0\ to\ n-1 \qquad (2.26)$$

$$D\_B_{hor}\left(i,j\right) = abs(B(i,j) - B(i,j-1))\ for\ i = 0\ to\ m-1,\ j = 1\ to\ n-1 \qquad (2.27)$$

In equations (2.24) through (2.27), $D\_F_{ver}, D\_F_{hor}, D\_B_{ver}$, and D_B_{hor} signifies the absolute vertical difference of original image, horizontal difference of original image, vertical difference of intentionally blurred image & horizontal difference of intentionally blurred image , respectively. In order to analyze the variation of neighboring pixels after blurring and compare with the variation of neighboring pixels before blurring, variation absolute difference was computed given by

$$V_{ver} = max\left(0, D\_F_{ver}\left(i,j\right) - D\_B_{ver}\left(i,j\right)\right)\ for\ i = 1\ to\ m-1,\ j = 1\ to\ n-1 \quad (2.28)$$

$$V_{hor} = max\left(0, D\_F_{hor}\left(i,j\right) - D\_B_{hor}\left(i,j\right)\right)\ for\ i = 1\ to\ m-1,\ j = 1\ to\ n-1 \quad (2.29)$$

$$s\_F_{ver} = \sum_{i,j=1}^{m-1,n-1} D\_F_{ver}(i,j) \qquad (2.30)$$

$$s\_F_{hor} = \sum_{i,j=1}^{m-1,n-1} D\_F_{hor}(i,j) \qquad (2.31)$$

$$s\_V_{ver} = \sum_{i,j=1}^{m-1,n-1} D\_V_{ver}(i,j) \qquad (2.32)$$

$$s\_V_{hor} = \sum_{i,j=1}^{m-1,n-1} D\_V_{hor}(i,j) \qquad (2.33)$$

Equations (2.30) through (2.33) give the sum of coefficients of D_F_{ver}, D_F_{hor}, D_V_{ver}, D_V_{hor}, respectively. Normalizing the above sum coefficients was done using

$$b\_F_{ver} = \frac{(s\_F_{ver} - s\_V_{ver})}{s\_F_{ver}} \qquad (2.34)$$

$$b\_F_{hor} = \frac{(s\_F_{hor} - s\_V_{hor})}{s\_F_{hor}} \qquad (2.35)$$

The final blur estimate is given by

$$blur\_{}_{F} = \max(b\_F_{ver,} b\_F_{hor}) \qquad (2.36)$$

Intuitively one can say that the higher the value of the blur estimate in equation (2.36), it is more likely that the input image was blurred. Although [26] gives details on calculating the blur estimate, the blur decision algorithm is not mentioned. For the current work, a blur threshold of 0.40 was used.

IBD considers only horizontal and vertical blur. An image blurred in any other direction might possibly be classified as a sharp image. IBD looks for high gray scale variations between the given image and intentionally blurred image to classify the given frame as sharp. However, when the given image is a smooth image with no sharp features, IBD algorithm misclassified it as blurred image. IBD algorithm resulted in maximum number of missed detections and maximum number of false alarms.

A distribution of blur factor obtained using IBD algorithm for the same collection used for Haar algorithm collection 15_49 IR data is presented in Figure 2.16.

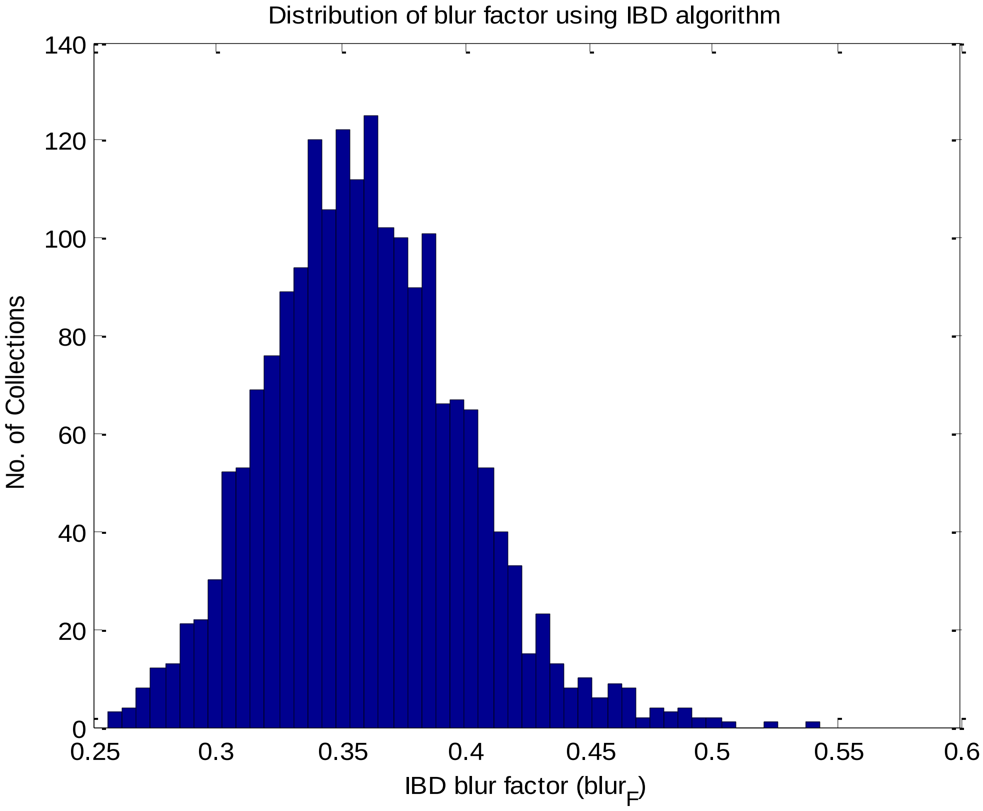
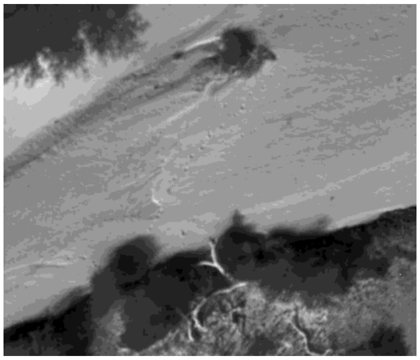


Figure 2.16 Histogram plot of the blur factor using IBD algorithm

A sharp image and intentionally blurred version of the sharp image are shown in Figure 2.17 and Figure 2.18. Figure 2.18 was obtained by applying a vertical low pass filter to the sharp image of Figure 2.17. A blurred frame and intentionally blurred version of blurred frame are shown in Figure 2.19 and Figure 2.20, respectively.



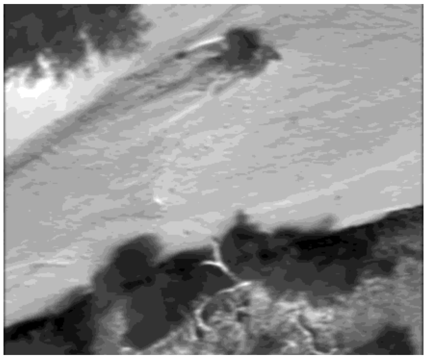
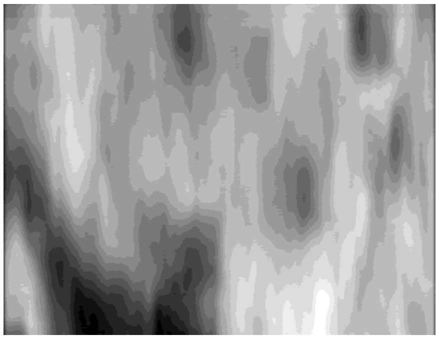


Figure 2.17 Sharp image

Figure 2.18 Intentional blur on a sharp image



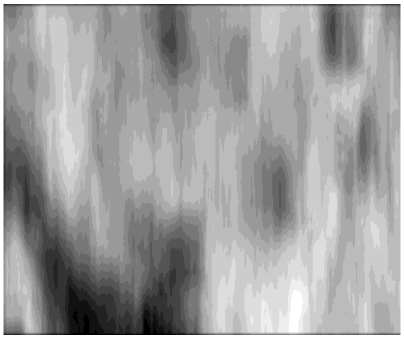


Figure 2.19 Blurred image

Figure 2.20 Intentional blur on a blurred image

Some of the sample missed detections and false alarms reported by IBD algorithm is given in Figure 2.21 and Figure 2.22, respectively.

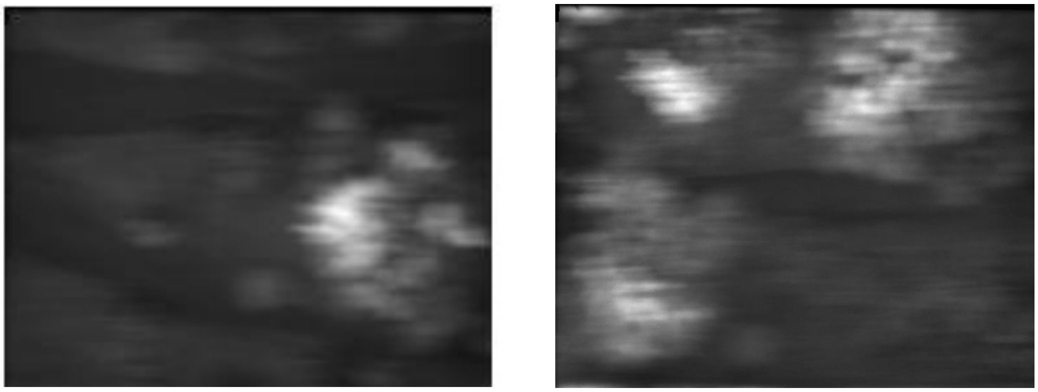


Figure 2.21 Sample missed detection using IBD algorithm

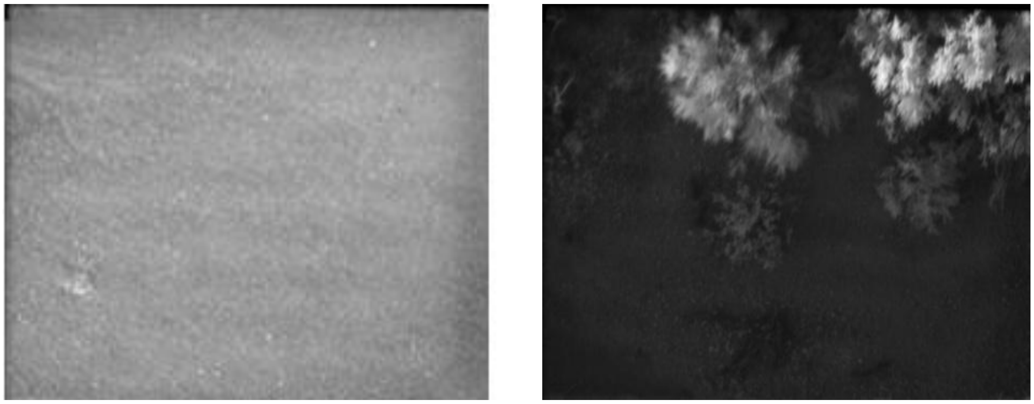


Figure 2.22 Sample false alarms reported by IBD algorithm

The IBD missed detection and false alarm statistics for the image frames in Figure 2.21 and Figure 2.22 is given in Table 2.7. Since IBD algorithm strictly looks at the difference in pixel intensities in the given image and intentionally blurred images, the image of Figure 2.22 resulted in false alarms.

Table 2.7 IBD algorithm missed detection and false alarm statistics

Parameter	Blurred Image	Sharp Image
factor	0.0607	0.4212
Classification	Not blurred (Missed detection)	Blurred (False alarm)

2.10 RESULTS & PERFORMANCE COMPARISON

The three blur detections described in this chapter were implemented in MATLAB® running on Intel Core 2 Duo machine with 2.40 GHz processor with 2 GB of RAM memory. Since there was an extensive amount of data that needed to be tested for image blur, the detection algorithms were run in parallel in several other machines including Intel Pentium 4 CPU, 3.20GHz processor with 2GB RAM and Intel Pentium D core CPU with 3.40 GHz processor speed and 2GB of RAM memory. The detection algorithms were tested on MWIR and MSI data provided by Night Vision and Electronic Sensors Directorate (NVESD) as part of their landmine detection program. The data that was used for this thesis was the flight data set of 2009.

MWIR images contain only the mid wave IR band where as MSI images are multi spectrum imagery that contain Red, Blue, and Green and near IR bands. Since MSI contains 4 bands, the detection algorithm can be tested in any of those 4 bands. In order to maintain uniformity band1 (near IR) was chosen across all MSI data.

The data capture effectively involved running a flight over the land mines and capturing the image frames with the help of built in camera sensor and GPS. The flight also collected metadata about the image frames. The metadata included location coordinates, altitude information, time of day of the flight (night/day time) and the type of data whether background or foreground data. Background data is of different kinds viz. vegetation, mix, dirt and rock. A typical flight data frame capture schematic is given in Figure 2.23.

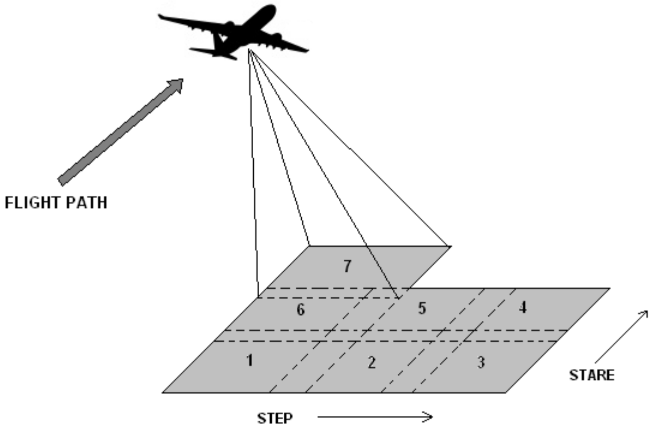


Figure 2.23 Typical flight frame capture

A program was written to process the massive data set of 2009. The automated program extracted the chosen data set from the database of collections and stored the frame wise detection results in the destination location as an Excel file. The frames that were detected as blurred were flagged as 1 and the frames that were detected as sharp were flagged as 0. The next section presents few examples of the detections and the summarization results follow in subsequent sections.

The detection algorithms described in this chapter were compared against the human subjective tests and the results of missed detections, false alarms and true detections are given in Table 2.8 for few selected collections. Three subjects were chosen to perform the test on randomly selected data sets. Each subject looked at each frame in the collection and classified them as blurred or not blurred. The "truth" for each frame was defined using majority voting and each algorithm and individual subject was evaluated against this "truth".

Table 2.8 Blur detection results: objective vs subjective tests

Test method	**052_02_08_41 Data: Night time IR**	**# of files processed**	**Missed detections**	**False alarms**	**Number of blurred detections**
Objective test	HAAR	1976	0	51	84
	SVD	1976	1	2	35
	IBD	1976	6	64	92
Subjective test	Human Subject 1	1976	1	1	34
	Human Subject 2	1976	0	0	34
	Human Subject 3	1976	4	0	30
Test method	**051_22_26_55 Data: Day time IR**	**# of files processed**	**Missed detections**	**False alarms**	**Number of blurred detections**
Objective test	HAAR	776	21	0	3
	SVD	776	0	2	26
	IBD	776	14	3	13
Subjective test	Human Subject 1	776	0	1	25
	Human Subject 2	776	0	0	24
	Human Subject 3	776	0	1	25

Table 2.8 Blur detection results : objective vs subjective tests (contd.)

Test method	**052_01_15_49 Data: Night time MSI**	**# of files processed**	**Missed detections**	**False alarms**	**Number of blurred detections**
Objective test	HAAR	1847	50	39	214
	SVD	1847	4	6	159
	IBD	1847	23	198	447
Subjective test	Human Subject 1	1847	2	1	156
	Human Subject 2	1847	4	0	153
	Human Subject 3	1847	1	0	156
Test method	**051_22_26_55 Data: Day time MSI**	**# of files processed**	**Missed detections**	**False alarms**	**Number of blurred detections**
Objective test	HAAR	775	17	1	3
	SVD	775	2	5	22
	IBD	775	2	10	27
Subjective test	Human Subject1	775	1	0	18
	Human Subject2	775	0	1	20
	Human Subject3	775	0	1	20
Test method	**051_22_26_55 Data: Day time MSI**	**# of files processed**	**Missed detections**	**False alarms**	**Number of blurred detections**
Objective test	HAAR	775	17	1	3
	SVD	775	2	5	22
	IBD	775	2	10	27
Subjective test	Human Subject1	775	1	0	18
	Human Subject2	775	0	1	20
	Human Subject3	775	0	1	20

From Table 2.8, it is inferred that Singular Value Decomposition (SVD) based modified detection algorithm produced the best results among the 3 algorithms under examination. A visual comparison of Haar, SVD and IBD algorithm against human subjective tests compared directly on the basis of number of blurred detections is given in Figure 2.24.

It can be inferred from Figure 2.24 that the number of detections identified by SVD closely agrees with the number detected by human subjects in this test.

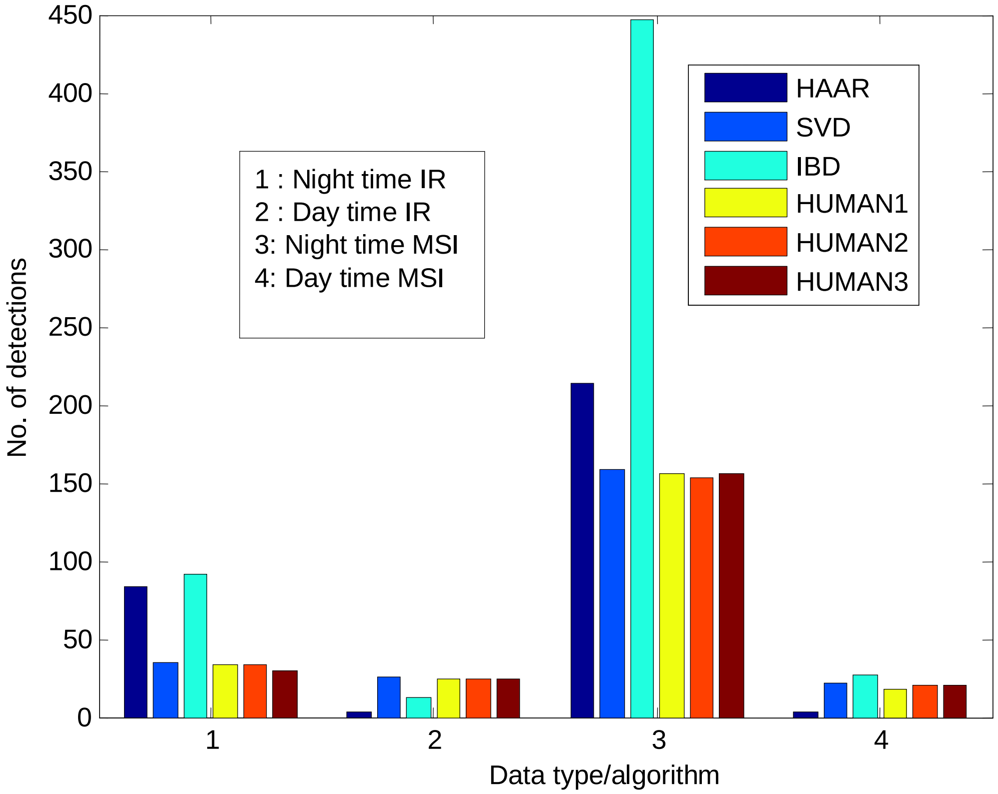


Figure 2.24 Comparison of objective algorithms vs subjective tests

A complete set of results of blur detection algorithms applied to a data set comprising of 88 collections of both IR and MSI data provided by NVESD (2009) is presented below. The result gives the number of detections for each algorithm. Since "truth" data was not available for all dataset, they could not be analyzed for statistics on false alarms and missed detections.

Collection flight day time IR data blur detection table using the 3 algorithms mentioned in this chapter are shown in Table 2.9. Collection flight night time IR data tested for blur using the detection algorithms described in this chapter are shown in Table 2.10. Collection flight night time II IR data blur detections are shown in Table 2.11. Collections of night time III IR data blur detection results are given in Table 2.12. Collection night time MSI data blur detection results are shown in Table 2.13. Collection day time MSI data blur detection results are given in Table 2.14.

It was observed that HAAR algorithm and SVD algorithm detected nearly same number of blurs in the night time foreground data. The number of detections reported by IBD was more than HAAR and SVD algorithms.

In case of day time data, the number of detections by SVD algorithm was much less than the number of detections by HAAR and number of detections by IBD.

Table 2.9 Number of detections on daytime IR data

Collection name	Number of frames	HAAR	SVD	IBD
051_22_26_55	776	243	28	83
051_22_38_24	1152	280	106	62
051_22_50_27	1352	374	122	188
051_21_33_05	1208	209	37	126
051_21_19_58	1376	196	47	99
051_21_39_29	1648	213	30	417
051_21_45_56	1680	259	41	352
051_21_59_53	1984	352	39	304
051_21_27_02	992	100	23	92
051_22_19_49	1184	377	12	35
051_22_25_35	264	56	3	12
051_21_53_30	728	99	25	162
051_21_55_50	800	109	20	165
051_21_16_35	592	65	43	46
051_22_06_40	1632	486	31	263
051_22_12_56	1456	467	12	154
051_22_31_39	1088	296	132	58
051_22_43_44	1112	263	99	49

Table 2.10 Number of detections on night time IR data

Collection name	Number of frames processed	HAAR	SVD	IBD
052_02_22_14	1496	1	4	32
052_02_08_41	1976	36	35	92
052_02_15_48	1624	5	5	48
052_02_29_13	1208	66	68	43
052_01_57_57	1064	12	14	26
052_02_03_56	912	6	5	22
052_01_42_25	1368	7	32	24
052_01_49_47	1672	114	119	102
052_01_28_39	1688	9	45	128
052_01_15_49	1960	145	196	300

Table 2.11 Number of detections on night time II IR data

Collection name	Number of frames processed	HAAR	SVD	IBD
062_18_32_40	32	29	28	24
062_19_25_30	64	62	59	57
062_19_50_03	2207	1182	1354	1336
062_19_56_43	2135	1072	1242	1354
062_19_37_05	656	467	453	390
062_19_32_17	688	474	450	376
062_19_27_04	1000	665	665	796
062_19_19_50	1440	1104	1030	1073
062_19_02_24	2151	1690	1568	1236
062_18_47_39	840	29	130	35
062_18_33_33	1032	427	607	971
062_18_24_35	48	45	44	6
062_18_25_45	1400	1123	1009	904
062_19_45_45	544	316	338	446
062_19_41_57	864	555	585	673
062_18_17_53	1952	1708	1530	915

Table 2.12 Number of detections on night time III IR data

Collection name	Number of frames processed	HAAR	SVD	IBD
051_02_34_01	792	215	50	135
051_03_39_25	1488	239	87	114
051_03_31_11	1560	43	2	288
051_03_46_59	1592	43	2	166
051_03_54_16	824	68	27	30
051_03_57_03	192	17	4	11
051_02_41_28	480	3	2	80
051_02_50_11	64	8	1	14
051_03_06_10	1376	7	2	153
051_03_13_31	1216	1	3	110
051_03_20_22	768	13	2	79
051_03_25_41	1160	33	25	148

Table 2.13 Number of detections on night time MSI data

Collection name	Number of frames processed	HAAR	SVD	IBD
052_02_22_14	1493	1452	281	1492
052_02_08_41	1943	1884	405	1942
052_02_15_48	1623	1575	316	1622
052_02_29_13	1180	1159	198	1180
052_01_57_57	1060	1059	69	1060
052_02_03_56	904	892	165	904
052_01_42_25	1365	1364	88	1364
052_01_49_47	1580	1579	291	1578
052_01_28_39	1682	825	5	1655
052_01_15_49	1847	214	159	447
052_01_22_34	1583	605	44	962
052_01_36_06	1626	1600	90	1619

Table 2.14 Number of detections on day time MSI data

Collection name	Number of frames processed	HAAR	SVD	IBD
051_21_39_29	1641	0	11	6
051_21_45_56	1681	0	20	5
051_21_59_53	1977	0	18	12
051_21_19_58	1375	3	22	15
051_21_27_02	988	0	6	6
051_21_33_05	1202	0	10	8
051_22_19_49	1188	0	12	9
051_22_25_35	268	0	4	4
051_22_26_55	775	3	22	27
051_21_53_30	518	0	5	2
051_21_55_50	798	0	3	244
051_21_16_35	590	0	19	26
051_22_06_40	1631	0	21	28
051_22_12_56	1457	1	13	9
051_22_31_39	1081	8	92	27
051_22_38_24	1155	1	63	9
051_22_43_44	1113	1	74	4
051_22_50_27	1354	6	70	27

3. PSF ESTIMATION

In order to restore an image it is critical to identify the degradation function. A degradation function in the context of image de-blurring is more often referred to as Point Spread Function (PSF). Iterative image restoration techniques require the PSF to obtain the restored image [27]. In this thesis, the focus was on estimation of the PSF for motion blurred images. Section 2 briefly talked about the detection of motion blurred images and this section addresses estimation of PSFs. It is important to study the characteristics of these motion blurred images before attempting to estimate the PSF. The Fourier transform of the PSF is given by

$$H(\xi_x, \xi_y) = \iint h(x,y) e^{-2\pi i (x\xi_x + y\xi_y)} dxdy \tag{3.1}$$

where $h(x,y)$ denotes the PSF that caused the degradation and ξ_x, ξ_y represents the frequency components in the x and y directions respectively. During a motion blur, pixel intensity of the original uncorrupted image points are shared between these image plane points according to the relative motion in which the pixel is exposed to light from the original pixel [27]. In simple words, adjacent points in the image plane are exposed to the same point on the object plane during the exposure time [11]. In real time imaging, the exposure time is very small, of the order of a fraction of a second typically between $(1/10000\ \sec - 1/100\ \sec)$. The averaging of the pixel gray values over neighboring pixels in the motion direction acts as a low pass filter in the spatial domain.

Since the motion blur leaves a strong noticeable impression on the blurred image that is prominent in the direction of motion, it is required to identify the blur direction relative to the image axis. Consequently, image resolution is expected to decrease in the motion direction [27]. When a motion blurred image is convolved with a high pass filter in the direction of motion, it will result in maximum suppression of image intensities than if the same image was convolved with a high pass filter in other directions. The motion direction can be identified by measuring the direction where the power spectrum of the image derivative is lowest [27]. In the following, three different methods for the estimation of motion blur direction and two methods of blur length are discussed and subsequently the three methods are evaluated for synthetically blurred images and real blurred images.

3.1 AUTOCORRELATION METHOD

In [11], an auto-correlation function (ACF) method was developed to estimate the motion blur parameters. When the relative homogeneity of an image is high, a derivative of such a blurred image along the motion direction will suppress the homogeneous area inside the tracks of the smear and emphasize their edges. As a blurred image contains more homogeneous area image content is significantly suppressed. Also, the image derivative on such an image will yield information that can be further investigated to estimate the extent of blur.

When a derivative followed by auto-correlation is applied in the motion direction, a minimum can be expected in the autocorrelation function of the image derivative at a distance of blur extent from the zero-shift center of the ACF function.

3.1.1. Identification of Motion Direction. Motion direction identification is based on the principle of finding that angle where the image derivative is lowest. That is, on performing a convolution between the image and its directional derivative results in

$$\Delta f(i,j)_k = f(i,j) * D_k(i,j) \tag{3.2}$$

where $f\left(i,j\right)$ is the blurred image of size $M \times N$ under investigation and $D_k\left(i,j\right)$ is the derivative operator and * is the 2D convolution. $D_k\left(i,j\right)$, for any arbitrary direction k is given by

$$D_k\left(i,j\right) = \begin{bmatrix} -1 & 0 \\ 1-\tan(k) & \tan(k) \end{bmatrix} \tag{3.3}$$

The total intensity of the convolution in each direction is then given by equation (3.4).

$$I\left(\Delta f\left(i,j\right)_k\right) = \sum_{i=0}^{N-1} 1 \sum_{j=0}^{M-1} |\,\Delta f\left(i,j\right)_k\,|, \text{where, } 0 \geq k \geq -90 \tag{3.4}$$

The direction where the total image intensity in equation (3.4) results in a minimum, gives the blur direction.

As an example, an 8 pixel motion blur in the horizontal direction relative to the image axis was applied on natural image of Figure 3.1. The artificial motion blur was introduced by using the *fspecial* function available in MATLAB®. A 25dB noise was added to this blurred image to learn the effectiveness of this algorithm with noise.

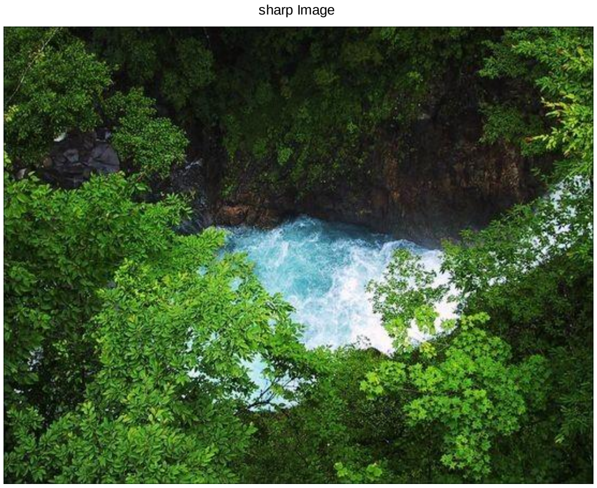


Figure 3.1 Test sharp image

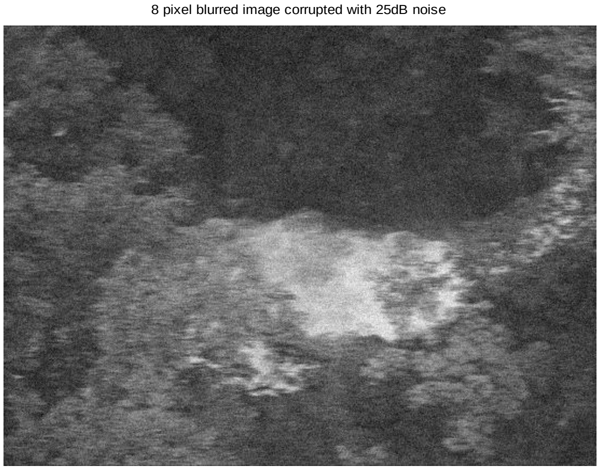


Figure 3.2 An 8 pixel blurred image corrupted with 25dB noise

On applying equations (3.2) through (3.4) a plot of direction vs normalized directional derivative was obtained as shown in Figure 3.3. It can be inferred from Figure 3.3 that a blur direction of 0° was estimated as seen from the minima of the normalized image derivative.

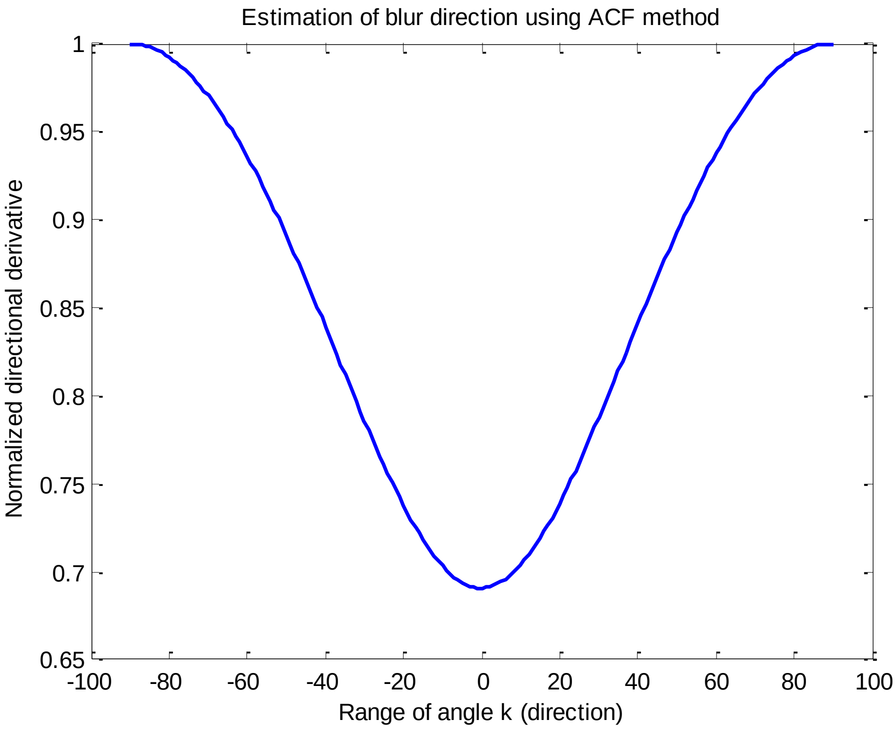


Figure 3.3 Estimation of blur direction using ACF method

Another example of direction estimation using ACF method is shown in Figure 3.4. The image of Figure 3.1 was motion blurred at an angle of 27° and at a length of 8 pixel blur at 25dB noise. ACF method estimated a 23° motion blur as seen in Figure 3.4.

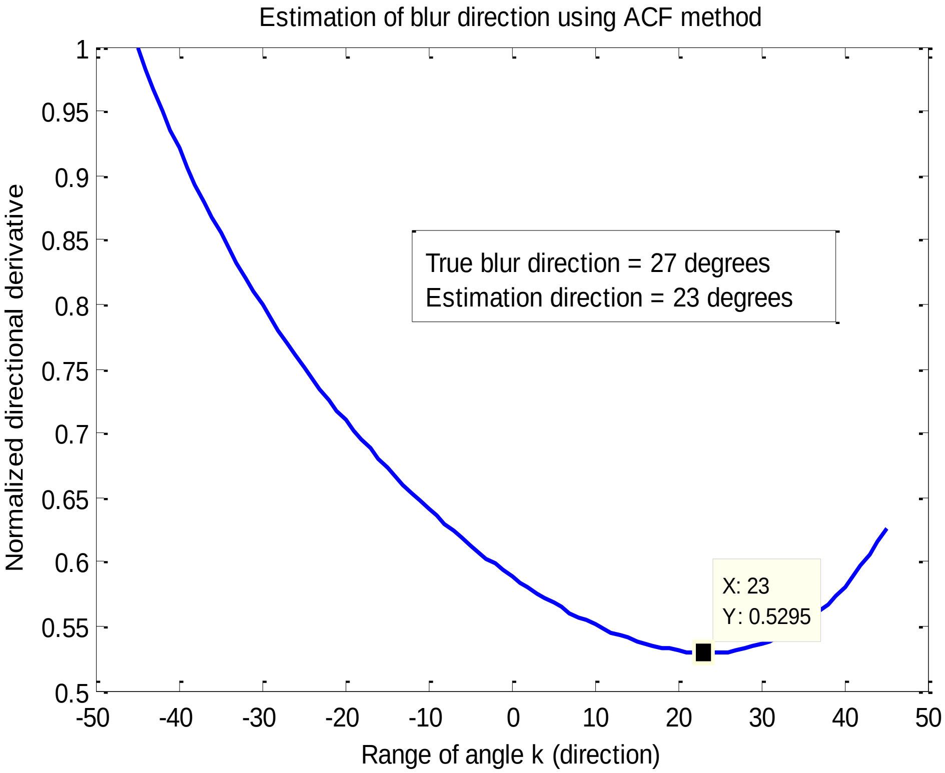


Figure 3.4 Estimation of direction using ACF method

3.1.2. Identification of Blur Extent. The ACF computed on the image derivative lines in the motion direction was calculated using

$$R_l(j) = \sum_{i=-M}^{M} l(i+j)\,l(i) \quad \text{where, } j \in \text{[-M,M], where } l(i) = 0\text{, for } i \notin \text{[0,M]} \qquad (3.5)$$

The blur extent was then calculated by noting the distance between the zero-shift of the auto-correlation function and the first minimum from that zero-shift center.

In continuation with the same example used for direction estimation, length estimation was performed using equation (3.5). A motion blur of 6 pixel length was introduced in the horizontal direction. ACF was computed on each row of the image derivative and then the mean of the ACFs was used for identification purpose. The mean of the ACFs computed on the image derivative lines is shown in Figure 3.5. It can be seen that a blur pixel length of 7 was detected for this example.

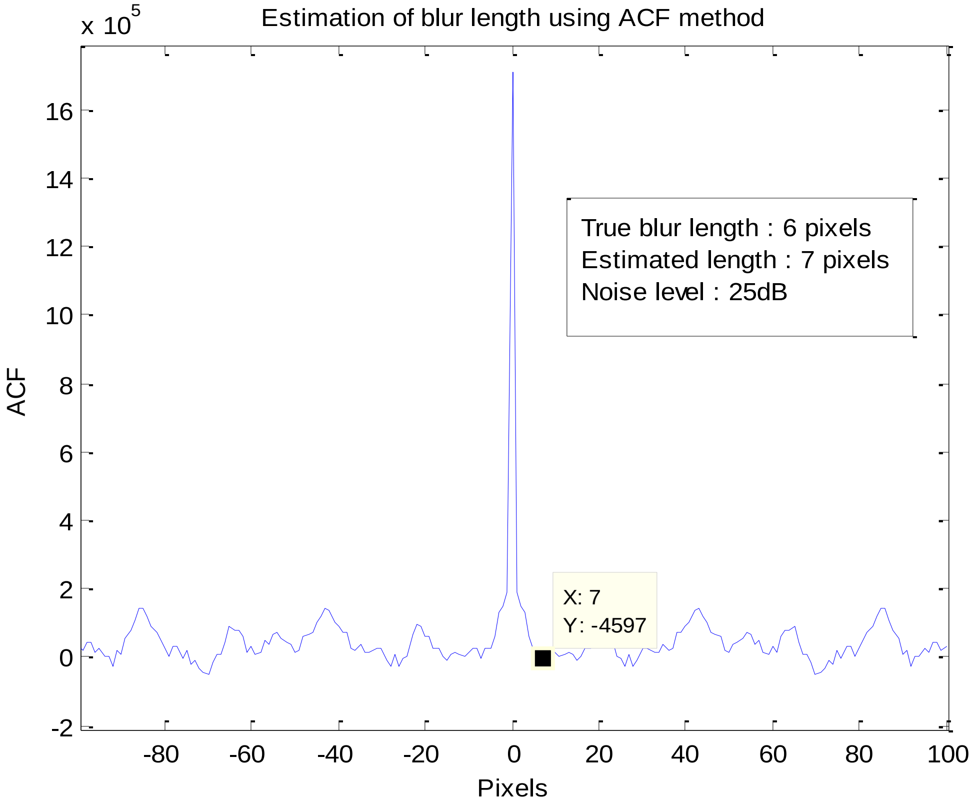


Figure 3.5 Length estimation using ACF method

To overcome ambiguous minima's, the blurred image was average filtered prior to computing the image derivative following which ACF algorithm was applied. Figure 3.6 represents the auto-correlation plotted on a single image derivative line for an average filtered blurred image derivative. The true blur length used was 8 pixels blur in the horizontal direction. In Figure 3.6, an 8 pixel blur was identified by the ACF algorithm. The reason for lesser number of peaks and valleys around the zero shift center is that the original image was filtered with a 3×3 low pass filter (LPF) prior to computing the image derivative. Thus, the task of finding the first dominant minima in the ACF function is easier in the latter case (LPF filtered). However, if the original image is not corrupted with noise, low pass filtering the image prior to the derivative will result in incorrect computation of image derivative and hence resulting in incorrect estimation of motion blur.

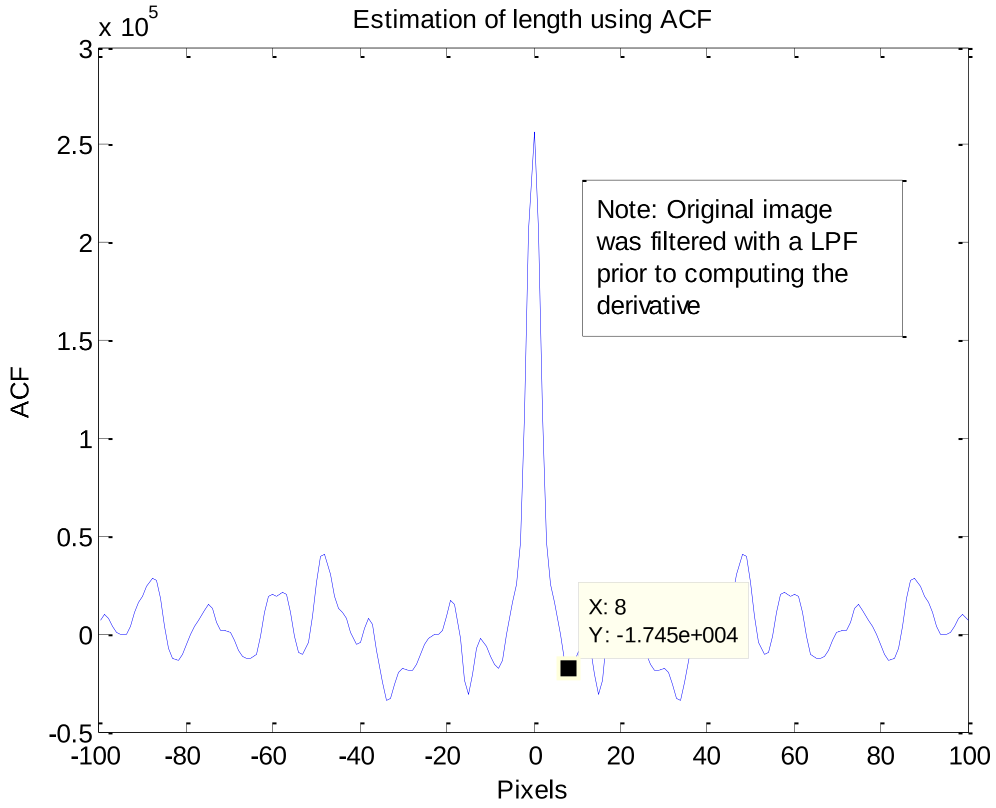


Figure 3.6 Estimation of ACF on an average filtered blurred image

3.2 HOUGH TRANSFORM METHOD

Auto-correlation based PSF estimation method is computationally expensive. This can be attributed to the fact that auto-correlation involves finding correlation between the neighboring pixels which will depend on the size of the image. The larger the image size, the longer is the time it takes to compute ACF. Image derivative gets affected with increased noise. Since ACF algorithm heavily depends on image derivative, the algorithm incorrectly estimates the blur extent in the presence of increased noise. Thus there was a need for improvement and relevant literature was studied. Hough transform was considered as an option as its computational cost is lesser than ACF algorithm. The underlying principle behind using Hough transform is detecting the line that is oriented in the direction of motion blur relative to the image axis.

Hough transform is widely used in automated image analysis and computer vision application for detecting lines and other specific shapes like circles, ellipses etc.

Basically, Hough transform is a feature extraction technique that is based on the principle of a voting procedure to confirm whether a given pixel is an edge pixel of a line.

For each pixel and its neighborhood, the Hough transform works by deciding if that pixel is an edge pixel. If the Hough transform determines that there is indeed enough evidence for a pixel to be an edge pixel, it then calculates the parameters of the line that the pixel falls in and increases the corresponding location in the accumulator array by 1.

Figure 3.7 represents the simplest case of Hough transform for detecting straight lines. The equation of the straight line is $y = mx + b$. In the Hough space, the Cartesian co-ordinate system is transformed to (r, θ) space. The parameter r represents the distance to the line from the origin and θ is the angle between the origin and the normal from the straight line to the origin. Using the parametric approach the equation of a straight line can now be written as

$$y = -\left(\frac{cos\theta}{sin\theta}\right)x + \left(\frac{r}{sin\theta}\right) \tag{3.6}$$

So for any given point (x_0, y_0) in the image plane, since an infinite number of lines can pass through this point, any line that does pass through this point must obey

$$r(\theta) = x_0 cos(\theta) + y_0 sin(\theta) \tag{3.7}$$

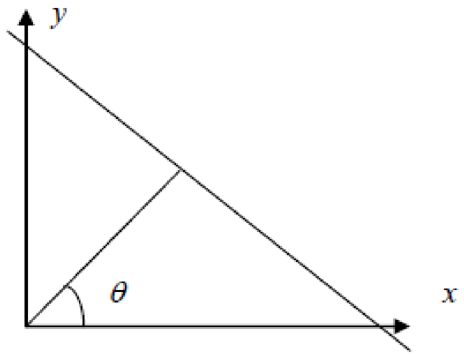


Figure 3.7 Hough parameter space

In [12], Hough transform based parameter estimation of motion blur has been developed. A specific instance of motion blur is given where in the scene moves relative to the sensor at a constant velocity V with an angle of ϕ degrees relative to the horizontal axis during the exposure interval [0, T], and then the motion blur PSF $h(x, y)$ for blur length L is given by

$$h(x, y) = \frac{1}{L}, if\ 0 \le |x| \le Lcos\phi; y = Lsin\phi \tag{3.8}$$

Estimation of motion blur parameters effectively involves finding the blur direction and then computing the amount of pixel blur extent. The estimation proceeds on the observation and assumption that the spectrum of an original non-blurred image is isotropic whereas that of a blurred image is anisotropic.

Hough transform line detector algorithm was used to find the best line among various detected lines. Since motion blurred image is biased in the direction of blur, a prominent effect can be seen in one particular direction than any other direction. For instance, the image frame on the left hand side of Figure 3.8 is blurred at 45° to the horizontal axis whereas the image on the right side of Figure 3.8 is blurred at 70° to the image axis.

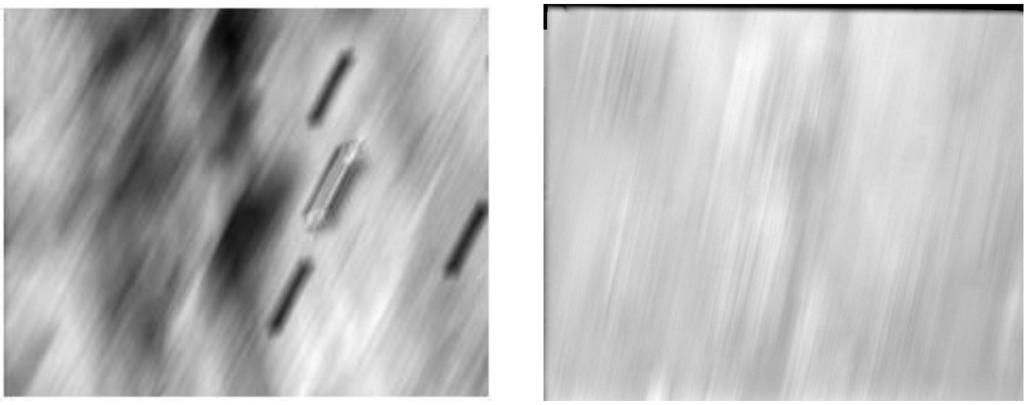


Figure 3.8 Image frames blurred at 45 degrees and 70 degrees, respectively

The Fourier transform of the blurred image was computed and log spectrum of the transform was obtained. In order to reduce the computational cost of Hough transform the spectrum was binarized. Mean intensity of the spectrum was obtained and used as the threshold value to obtain the binary image.

Hough transform was calculated on the binary image to arrive at the accumulator array. For a given point (x, y) in the image, the corresponding curve given by $r = x\cos\phi + y\sin\phi$ is entered in the accumulator and the count in the accumulator array was increased for each point along the curve. Accumulator array was built by an incremental voting process. The range of ϕ was computed in equal steps and the step size was passed into the computation block. Now each and every pixel in the image was subject to the accumulator array. After all the pixels pass through the accumulator array, the location where the maximum value in the accumulator occurs corresponds to the blur direction. The ϕ value corresponding to the maximum value of the accumulator array is the angle perpendicular to the blur direction. True blur direction was obtained by subtracting ϕ from 90°.

The next step is to calculate the blur length. As in the case of finding the motion direction, Fourier transform (FT) of the blurred image was calculated. The log spectrum of the Fourier transform was computed and converted to binary image. Now, this binary image was rotated in the direction opposite to the blur direction. The 2D image data was then converted into 1D data by computing averages along columns. The estimation of blur length is analogous to locating the frequency domain zeroes in a blurred image. The frequency domain zeroes appear at periodic intervals and the distance between any two zeroes gives the blur length. Similarly, the distance from the origin to the first zero also gives the blur length as the zeroes appear on both sides of the frequency spectrum. Then, inverse Fourier transform (IFFT) was computed on the 1D data and the first negative value on the real part was located. The location of the first negative value in the 1D array corresponds to blur pixel length.

Table 3.1 gives the (r,ϕ) corresponding to the maximum value in the Hough transform accumulator array.

Table 3.1 (r,ϕ) corresponding to maximum in the Hough accumulator array

True blur pixel length	Noise level (dB)	True blur direction	ϕ (max of accumulator array)	Estimated direction
8	1	0	90	0
8	5	39	46	44
5	25	79	23	67

Angle estimation was carried out by considering only the first location in the detected (r,ϕ) array. The blur direction was obtained by subtracting ϕ from 90°.

Length estimation was implemented as per the algorithm details given in Section 3.2. The inverse transform on the 1D data yielded the first negative real part at the 8th pixel. Thus, an 8-pixel blur was identified by Hough transform algorithm.

3.3 STEERABLE GAUSSIAN FILTER ANGLE ESTIMATION

Freeman and Adelson in [30] have provided a way to synthesize filters of different orientations from a combination of basis filters. Basically, Steer Gaussian filter method is an approach to calculate the response of the image by convolving with filters oriented at different angles and identifying the maximal response. All functions that are bandlimited in the angular frequency are steerable, that is they give enough basis filters [30]. Consider a 2D Gaussian function G given by

$$G(x,y)=e^{-(x^2+y^2)} \tag{3.9}$$

The 1st derivative of the above Gaussian function along the x-axis (0°) is given by

$$G_1^0(x,y)=\frac{\partial G(x,y)}{\partial x}=-2xe^{-\left(x^2+y^2\right)} \tag{3.10}$$

Now, after rotating the Gaussian function by 90° gives a derivative of

$$G_1^{90}(x,y)=\frac{\partial G(x,y)}{\partial y}=-2ye^{-\left(x^2+y^2\right)} \tag{3.11}$$

The G_1^{90} and G_1^0 are called the basis filters from which any filter G_1^{θ} at any arbitrary angle θ can be synthesized given by

$$G_1^{\theta}=\cos(\theta)G_1^0+\sin(\theta)G_1^{90} \tag{3.12}$$

From these basis filters, image response was found out by convolving the image filtered with G_1^{90} and G_1^0 giving

$$R_1^0=G_1^0*I \tag{3.13}$$

$$R_1^{90}=G_1^{90}*I \tag{3.14}$$

And then linear combination of these two filters at any given direction gives

$$R_1^{\theta}=\cos(\theta)R_1^0+\sin(\theta)R_1^{90} \tag{3.15}$$

The direction range for the purpose of direction estimation was chosen from 0° to 180°. The test image was convolved with Steerable Gaussian filter at different orientations and the response was calculated. It is to be noted that the response image in each direction is a 2D matrix. Absolute magnitude of the response image was calculated at each orientation. The orientation corresponding to the lowest magnitude sum was decided as the blur direction. For an image blurred at 60°, the estimated angle was 54° as shown in the Figure 3.9.

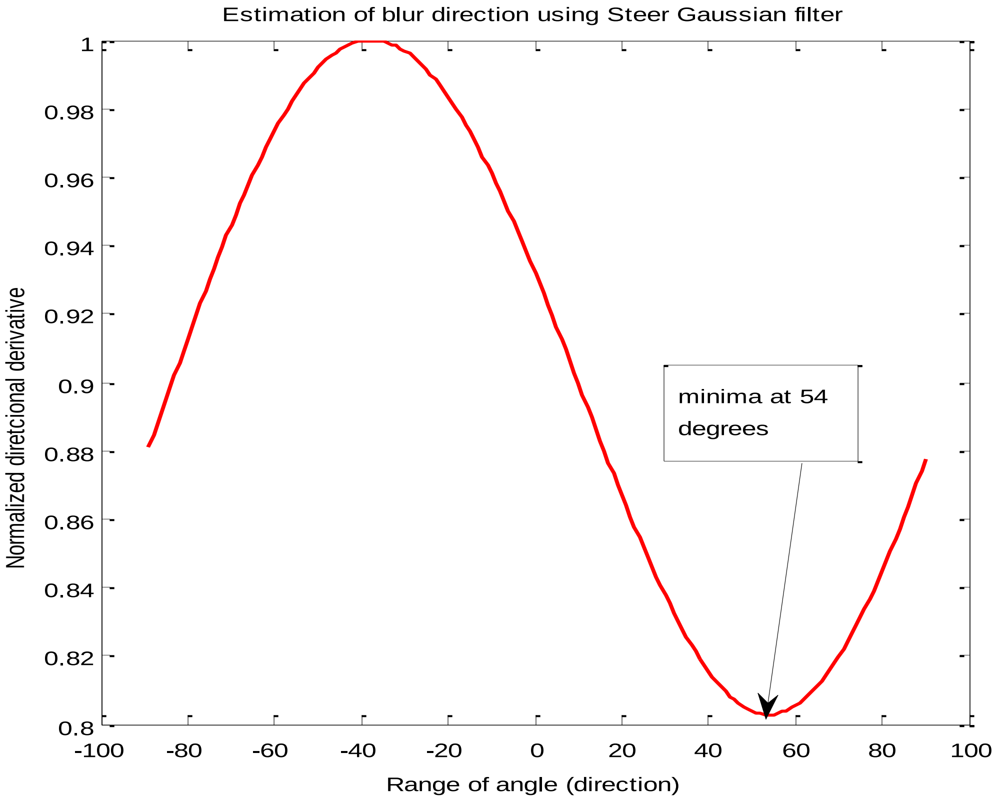


Figure 3.9 Direction estimation using steerable Gaussian filter

The length estimation followed the same approach as auto-correlation method given in Section 3.1. The angle obtained in steerable Gaussian filter was used as input to calculate image directional derivative in ACF method. The angle estimation error distribution using SGF method at different noise intervals is shown in Figure 3.10. 100

randomly chosen angles were tested at different noise levels 1dB, 5dB, and 25dB. The range of the angles was [-90,90].

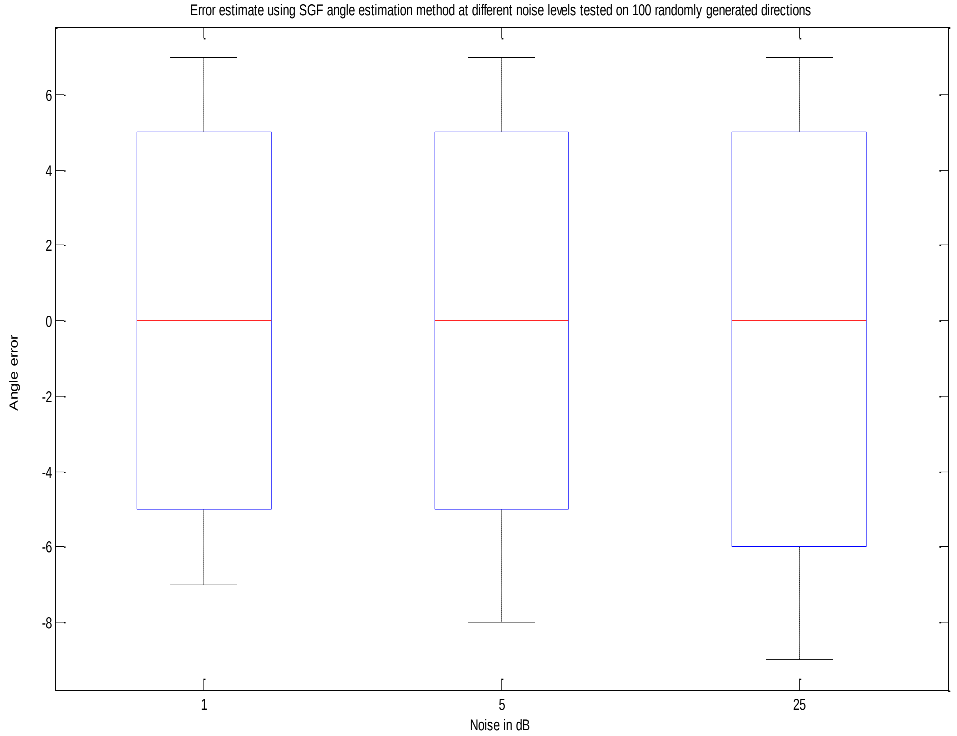


Figure 3.10 Angle error estimate using SGF method at different noise variance

3.4 PSF ESTIMATION RESULTS

The above described algorithms were tested on synthetically blurred images and the results are tabulated. All the algorithms were implemented in MATLAB® running on Intel Core 2 Duo machine with 2.40 GHz processor with 2 GB of RAM memory. Since the study pertained to motion blurred images, different sharp images were artificially blurred by introducing motion blur of different lengths and different orientations. Then, the artificially blurred image was corrupted with Gaussian noise at different levels of noise variance namely 25dB, 5dB and 1dB, respectively.

Tests were also performed on naturally motion blurred images for the data set of 2009 provided by NVESD.

Before presenting the results it is critical to mention the assumptions and conditions under which the algorithms were implemented. In calculating the accumulator array, angle step of 10° was used and the segment size used was 3×3. The blur direction was obtained by locating the direction corresponding to the maximum value in the accumulator array.

Since the direction of motion is perpendicular to the one obtained using Hough transform, angle computed from the accumulator array was subtracted from 90°. The binary image used to input into the Hough transform was calculated by thresholding the log Fourier transform of the given image. Threshold value was obtained dynamically by calculating the mean value of the log Fourier transform. The minimum distance between the lines was restricted to $10^{\circ}$and the angle interval ranged from $-\pi/2$ to $\pi/2$ in increments of $\pi/180$.

On detecting the blur direction, the binary image was rotated by an amount of blur direction in counter clockwise direction. Mean of the rotated image was computed along each column resulting in a row vector. Inverse Fourier transform (IFFT) was then computed on the 1D row vector. The first location where the real part of the IFFT resulted in a negative value was declared the blur pixel length.

For the auto-correlation method, the angle estimated using the steer Gaussian filter was used as input to compute the directional derivative in the motion direction. In this thesis, the auto-correlation (ACF) was computed on the center line of the derivative image resulting in a 1D row vector. The distance from the zero shift of ACF to the location of the 1st minimum was declared as the blur pixel length. Figure 3.11 presents sample image used for study.

3.4.1. Analysis of PSF Estimation on Artificially Blurred Images. The test image of Figure 3.11 was used to motion blur at a given angle and given length.

Figure 3.11 Uncorrupted sharp image for PSF estimation tests

The direction range was chosen from [-90, 90] and the pixel length was chosen between 0 and 8. In the test results shown below, the true blur lengths used were 8 pixel blur during one test and 3 pixel blur during the second test. Eight blur directions were chosen between [-90, 90]. The test was conducted at different levels of additive Gaussian noise to study the effectiveness of the algorithm at 25dB, 5dB and 1dB noise variance, respectively.

The experiment was repeated for 20 times in order to study the confidence measure of the estimated PSF values. The number in the X-axis of the chart indicates the noise level. Figure 3.12 presents the results of PSF estimation using Hough transform based method carried out for 20 trials at different noise levels. The quantity on the Y-axis is the estimation error in terms of number of pixels. Error in this context refers to the difference between the true blur pixel length and the length estimated using estimation algorithm (Hough algorithm) in this case.

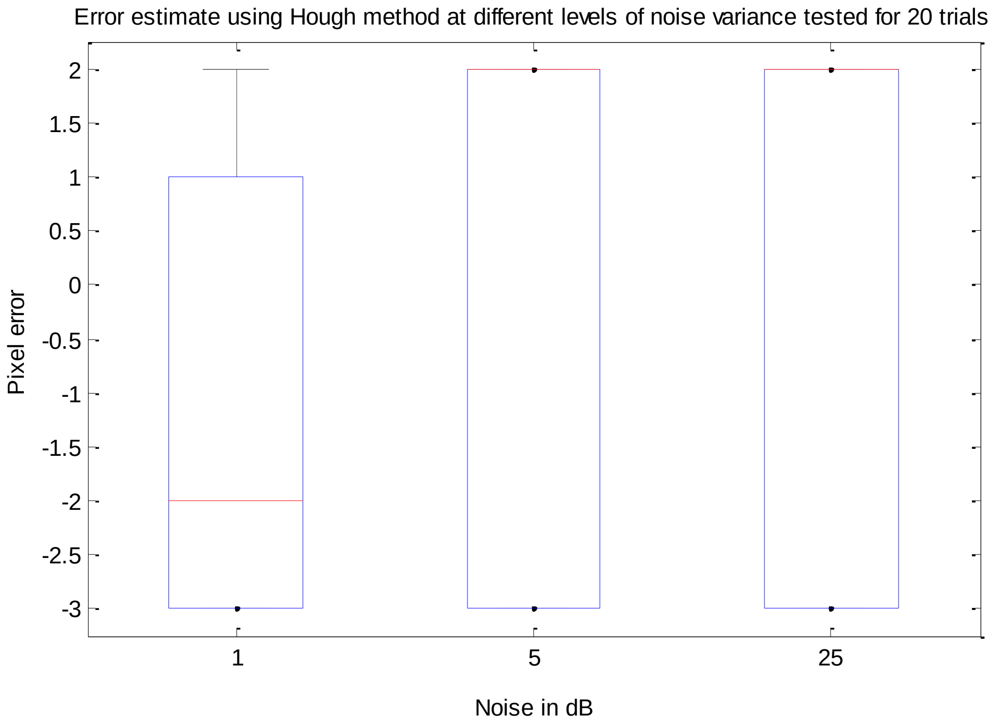


Figure 3.12 Blur length estimation error using Hough method

Figure 3.13 presents the results of study of estimation error on the same image of Figure 3.10 blurred at different angles and lengths using ACF method. It can be inferred from Figure 3.13 that with the increase in noise level, the error between the true and estimated blur length also increased. Also, it can be seen that there is an outlier at 5dB noise in Figure 3.13. Outliers indicate bad estimation. Outliers signify that the difference between true and estimated blur pixel length is high. Figure 3.14 shows the results of estimation error using SGF method. As mentioned earlier, the direction computed from the steer Gaussian filter was used to compute the image directional derivative and then ACF method was used for length estimation. Although both SGF and ACF method use the same algorithm for length estimation, it can be observed from Figure 3.13 and Figure 3.14 that ACF method resulted in lesser estimation error compared to SGF method.

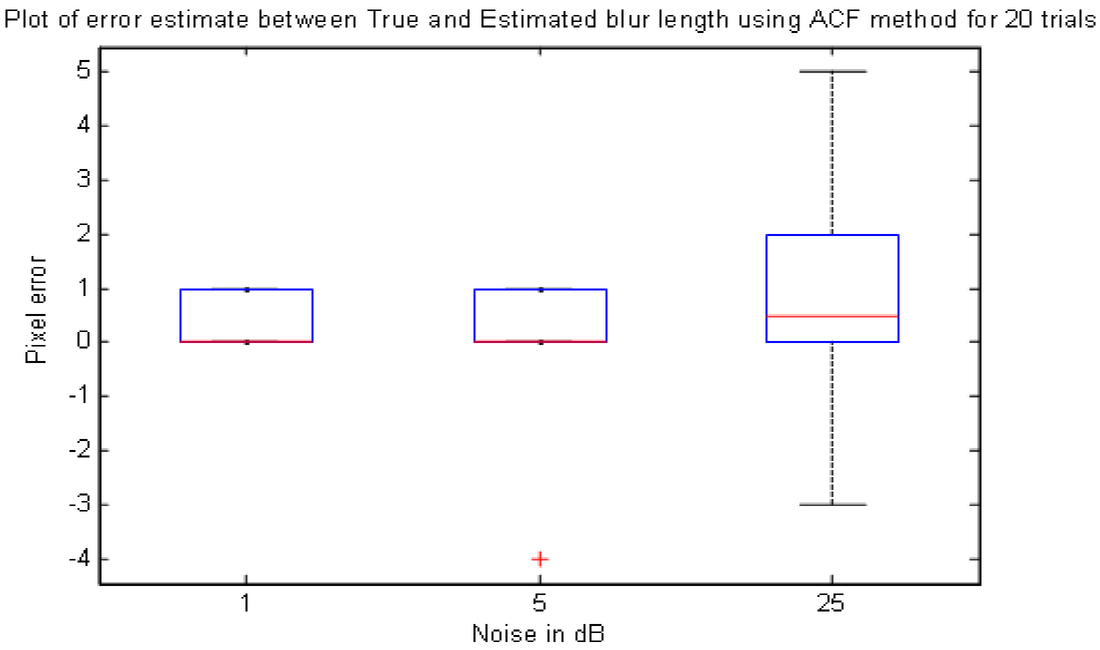


Figure 3.13 Blur length estimation error using ACF method

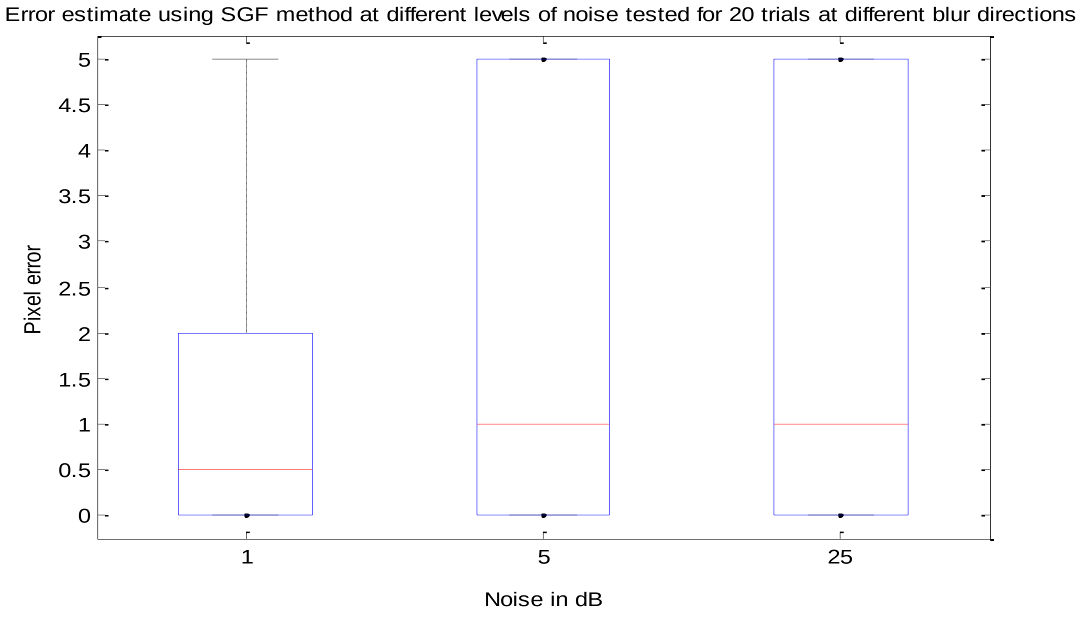


Figure 3.14 Blur length estimation error using SGF method

The distribution of pixel error estimate tested at different blur directions using Hough algorithm is given in Figure 3.15. True blur lengths used were 2 pixel blur and 8 pixel blur. The algorithm was tested at all three noise levels viz. 1dB, 5dB and 25dB for each direction.

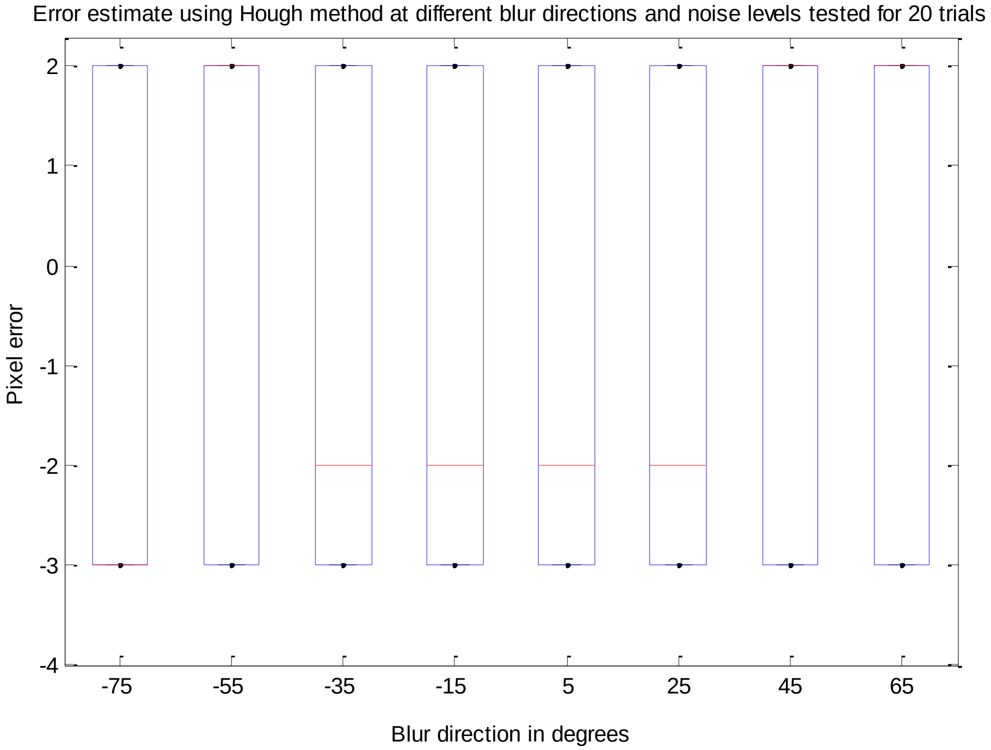


Figure 3.15 Blur length estimation error using Hough method at different directions

The distribution of pixel error estimate tested at different blur directions using ACF method is given in Figure 3.16. True blur lengths were 2 pixel blur and 8 pixel blur. The same set of directions as Hough method was used for ACF method in order to maintain uniformity.

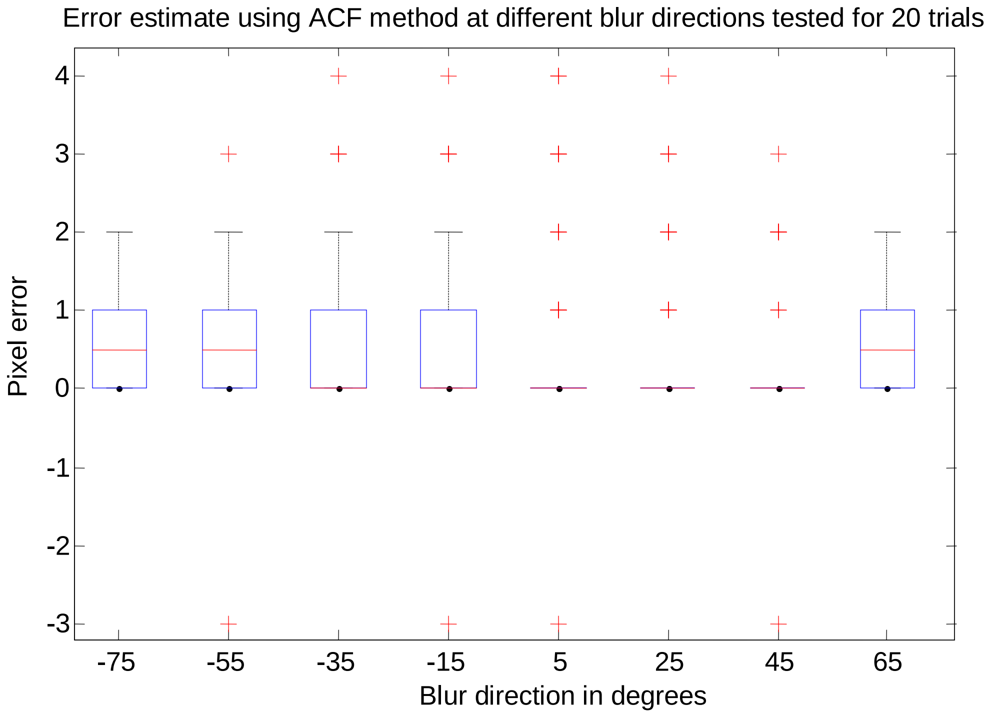


Figure 3.16 Blur length estimation error using ACF method at different blur directions

The distribution of pixel error estimate using SGF method tested at different blur directions is given in Figure 3.17. The test conditions were the same as ACF method and Hough method where the true blur lengths were 8 pixel blur at one instance and 2 pixel blur during another instance. The noise level used was 1dB, 5dB and 25dB noise in each direction.

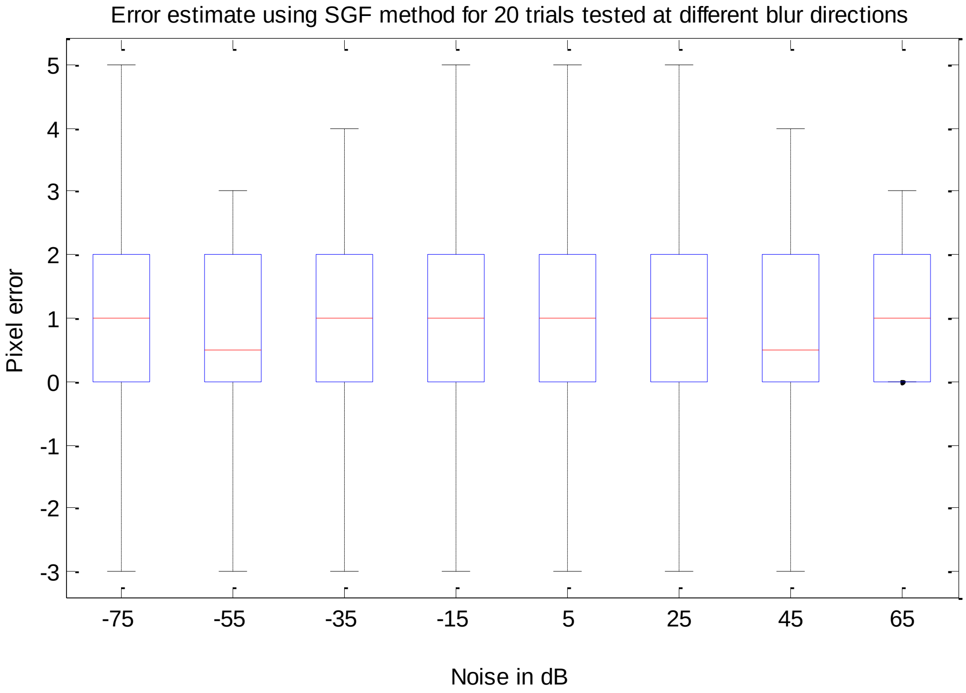


Figure 3.17 Blur length estimation error using SGF method at different blur directions

3.4.2. Results of PSF Estimation on Naturally Blurred Images. The data set of 2009 provided by NVESD was used for the tests. Tests were performed on chosen motion blurred IR and MSI collections. Some examples of naturally motion blurred images are given in Figure 3.18 and Figure 3.19.

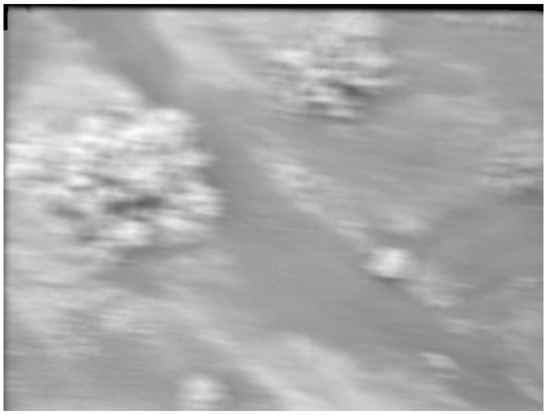


Figure 3.18 Representative low motion blurred frame (062_19_25_30_12)

Figure 3.19 Representative high motion blurred frame (062_19_25_30_20)

Certainly Figure 3.19 is more blurred than Figure 3.18. Hough algorithm computed a 3 pixel blur on the frame shown in Figure 3.18 and an 8 pixel blur on the frame shown in Figure 3.19. Another example of naturally motion blurred frames is given in Figure 3.20 and Figure 3.21. In Figure 3.20, a 6 pixel blur extent was identified by Hough algorithm whereas a 4 pixel blur was detected by auto-correlation based algorithm. The frame shown in Figure 3.21 was detected as a 7 pixel blur by Hough estimate and 11 pixel blur by auto-correlation estimate.

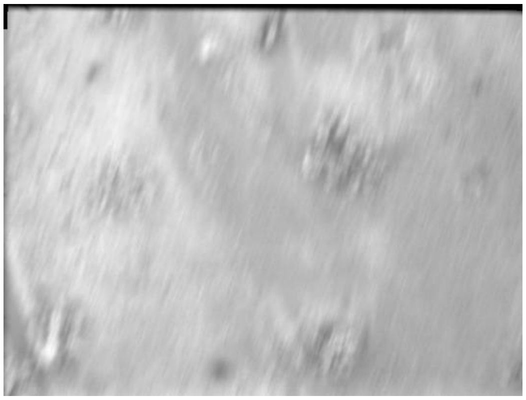


Figure 3.20 Representative low motion blurred frame (062_19_25_30_40)

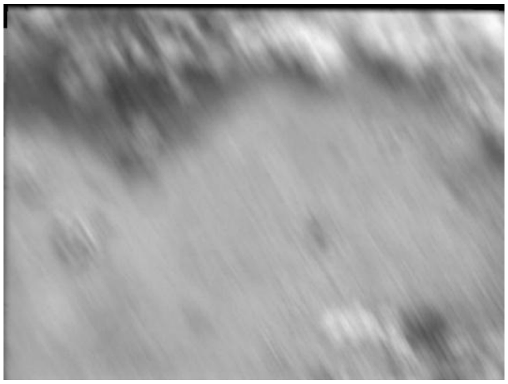


Figure 3.21 Representative high motion blurred frame (062_18_32_40_27)

Overall results of data night time III for collection 062_19_25_30 and data IR 32_40 is given in Table 3.2 and Table 3.3, respectively. The results of PSF estimation applied on 25_30 and 32_40 MSI data are given in Table 3.4 and Table 3.5, respectively.

Table 3.2 Estimated blur length on night time IR III data collection 062_19_25_30

Collection frame name	Hough Estimate	Auto-correlation Estimate
062_19_25_30_ 000001.rawl	5	1
062_19_25_30_ 000002.rawl	8	3
062_19_25_30_ 000003.rawl	7	2
062_19_25_30 _000006.rawl	7	2
062_19_25_30_ 000007.rawl	6	1
062_19_25_30_ 000009.rawl	5	2
062_19_25_30_ 000010.rawl	8	1
062_19_25_30_ 000011.rawl	6	1
062_19_25_30_ 000012.rawl	3	2
062_19_25_30 _000013.rawl	3	1
062_19_25_30 _000015.rawl	8	2
062_19_25_30_000016.rawl	9	1
062_19_25_30_000017.rawl	3	2
062_19_25_30_000018.rawl	7	1
062_19_25_30_000019.rawl	9	1
062_19_25_30_000020.rawl	8	5
062_19_25_30_000022.rawl	8	2
062_19_25_30_000023.rawl	5	8
062_19_25_30_000025.rawl	6	6
062_19_25_30_000026.rawl	9	4
062_19_25_30_000028.rawl	8	1
062_19_25_30_000033.rawl	8	4

Table 3.2 Estimated blur length on night time IR III collection 062_19_25_30 (contd.)

062_19_25_30_000035.rawl	6	4
062_19_25_30_000036.rawl	8	4
062_19_25_30_000037.rawl	8	5
062_19_25_30_000038.rawl	5	2
062_19_25_30_000039.rawl	5	4
062_19_25_30_000040.rawl	6	4
062_19_25_30_000041.rawl	7	1
062_19_25_30_000042.rawl	8	3
062_19_25_30_000043.rawl	6	2
062_19_25_30_000045.rawl	7	3
062_19_25_30_000048.rawl	3	1
062_19_25_30_000049.rawl	8	3
062_19_25_30_000055.rawl	8	2
062_19_25_30_000056.rawl	7	1
062_19_25_30_000057.rawl	6	1
062_19_25_30_000058.rawl	7	1
062_19_25_30_000059.rawl	6	1
062_19_25_30_000060.rawl	8	3
062_19_25_30_000062.rawl	6	5

Table 3.3 Estimated blur length on data IR collection 32_40

Collection frame name	Hough Estimate	Auto-correlation Estimate
062_18_32_40_000001.rawl	9	5
062_18_32_40_000003.rawl	8	2
062_18_32_40_000006.rawl	7	1
062_18_32_40_000007.rawl	8	7
062_18_32_40_000008.rawl	6	1
062_18_32_40_000009.rawl	8	4
062_18_32_40_000010.rawl	8	1
062_18_32_40_000011.rawl	6	2
062_18_32_40_000012.rawl	5	2
062_18_32_40_000013.rawl	3	2
062_18_32_40_000015.rawl	5	1
062_18_32_40_000016.rawl	9	1
062_18_32_40_000017.rawl	5	2
062_18_32_40_000027.rawl	7	11
062_18_32_40_000028.rawl	8	6
062_18_32_40_000029.rawl	9	2
062_18_32_40_000030.rawl	6	6
062_18_32_40_000031.rawl	9	3
062_18_32_40_000032.rawl	8	7

Table 3.4 Estimated blur length on MSI collection 25_30 band1

Collection frame name	Hough Estimate	Auto-correlation estimate
062_19_25_30_000001.rawl	6	5
062_19_25_30_000002.rawl	6	1
062_19_25_30_000005.rawl	6	3
062_19_25_30_000006.rawl	6	7
062_19_25_30_000007.rawl	5	2
062_19_25_30_000008.rawl	6	1
062_19_25_30_000009.rawl	6	2
062_19_25_30_000010.rawl	6	6
062_19_25_30_000011.rawl	6	1
062_19_25_30_000012.rawl	6	2
062_19_25_30_000013.rawl	6	3
062_19_25_30_000017.rawl	6	6
062_19_25_30_000018.rawl	6	6
062_19_25_30_000020.rawl	6	2
062_19_25_30_000021.rawl	6	7
062_19_25_30_000023.rawl	6	4
062_19_25_30_000025.rawl	6	2
062_19_25_30_000026.rawl	6	5
062_19_25_30_000027.rawl	6	11
062_19_25_30_000031.rawl	6	8
062_19_25_30_000033.rawl	6	3
062_19_25_30_000035.rawl	6	7
062_19_25_30_000037.rawl	6	2
062_19_25_30_000038.rawl	6	8
062_19_25_30_000039.rawl	6	3
062_19_25_30_000042.rawl	6	6
062_19_25_30_000043.rawl	6	5
062_19_25_30_000045.rawl	6	3
062_19_25_30_000047.rawl	6	3
062_19_25_30_000048.rawl	6	8
062_19_25_30_000049.rawl	6	2
062_19_25_30_000050.rawl	6	1
062_19_25_30_000052.rawl	6	2
062_19_25_30_000053.rawl	6	4
062_19_25_30_000056.rawl	6	9
062_19_25_30_000057.rawl	6	1
062_19_25_30_000058.rawl	3	2
062_19_25_30_000059.rawl	6	2
062_19_25_30_000060.rawl	6	8
062_19_25_30_000061.rawl	3	8
062_19_25_30_000062.rawl	6	3
062_19_25_30_000063.rawl	6	7
062_19_25_30_000066.rawl	6	1
062_19_25_30_000067.rawl	6	2

Table 3.5 Estimated blur length on MSI collection 32_40 band1

Collection frame name	Hough Estimate	Auto-correlation estimate
062_18_32_40_000002.rawl	6	2
062_18_32_40_000003.rawl	6	2
062_18_32_40_000005.rawl	6	2
062_18_32_40_000006.rawl	6	2
062_18_32_40_000007.rawl	6	7
062_18_32_40_000008.rawl	3	4
062_18_32_40_000009.rawl	3	3
062_18_32_40_000010.rawl	6	1
062_18_32_40_000013.rawl	6	2
062_18_32_40_000015.rawl	6	4
062_18_32_40_000018.rawl	6	2
062_18_32_40_000020.rawl	5	4
062_18_32_40_000021.rawl	6	2
062_18_32_40_000022.rawl	6	4
062_18_32_40_000023.rawl	6	2
062_18_32_40_000025.rawl	6	2
062_18_32_40_000026.rawl	6	4
062_18_32_40_000027.rawl	6	2
062_18_32_40_000028.rawl	6	5
062_18_32_40_000029.rawl	6	1
062_18_32_40_000030.rawl	6	3
062_18_32_40_000031.rawl	5	6
062_18_32_40_000032.rawl	6	1
062_18_32_40_000033.rawl	6	2
062_18_32_40_000035.rawl	6	5
062_18_32_40_000036.rawl	6	2
062_18_32_40_000037.rawl	5	2
062_18_32_40_000038.rawl	6	1
062_18_32_40_000040.rawl	6	4
062_18_32_40_000041.rawl	6	3

4. CONCLUSION AND FUTURE WORK

This thesis considers the problem of blur detection as a critical image quality metric. The problem of blur detection has been pursued by computer vision researchers for a very long time. The ability of a computer to detect between a good and bad image data is challenging task especially in time critical real time applications. The problem of automatic blur detection finds application in modern digital cameras that can detect for bad images and automatically discard them, time critical image inspection, biomedical applications in detecting images that even an experienced cardiologist will not be able to say whether an image is good or not. A large number of airborne images that was captured in 2009 suffered motion blur and some of them suffered defocus blur. Manual inspection of the massive data set consumed several man hours and hence an automatic detection for blur in these airborne images became inevitable.

In this thesis, a direct form of blur detection has been used that looks for distinct features in the image. Edges in the image get severely affected on the event of an image blur and lose their sharpness and some of the edges tend to completely disappear. Based on [7], this study has focused on identifying the different edge types and the effect of blur on these different edge types viz. Dirac-structure, Roof-structure and Step-structure edges. Identifying the edges in an image has always been a challenging area in image processing and continues to be so.

Another blur detection algorithm exploited the fact that the variation in pixel intensities of neighboring pixels in a sharp image is much higher than a blurred image. It is intuitive that any attempt to further blur an already blurred image would result in lesser variation between neighboring pixel intensities than an attempt to blur an original sharp image which would result in higher pixel variations. Blur extent was obtained by arriving at a factor of maximum between the horizontal blur and vertical blur. The intentional blurring method resulted in a much faster test. However, large number of false alarms was obtained due to scenes of featureless terrains and low contrast images.

Singular value decomposition was performed on 3rd level Haar decomposition. Edge strength factor was evaluated on an image obtained from the horizontal and diagonal details of the 3rd level Haar decomposition. Blur decision was then arrived by comparing a ratio of singular values to a threshold value. Singular value decomposition method based on Haar wavelet transform resulted in capturing much of the detection that were missed by the original Haar wavelet method. This modified method using SVD had the least number of false alarms among the three blur detection algorithms evaluated here.

The second part of the thesis focused on estimating the point spread function from a motion blurred image. Two algorithms were pursued for the purpose of study. The algorithms were validated first on synthetically blurred images by comparing the actual blur parameters to the estimated values. The NVESD data set of 2009 was used to make visual verification on selected airborne data. The two methods that were researched and studied for PSF estimation were auto-correlation based method and Hough transform based method. The success of both these algorithm in detecting the true blur lengths depends on the identification of motion direction. Steerable Gaussian filter approach was used to determine the motion direction which was later used to detect the blur length in Auto-correlation method.

ACF method takes advantage of the fact that the homogeneity and smoothness in the motion direction is greater than any other direction in the image. As a result, derivative followed by autocorrelation in the motion direction, will result in a minimum in the autocorrelation function (ACF) at a distance of blur extent from the zero shift center of the ACF. In the tests carried out in this thesis it was observed that the ACF method was able to identify the blur extent correctly when the motion direction was parallel to the image axis even at Gaussian noise level as high as 25dB.

Hough transform based estimation method is a frequency domain approach to estimate the PSF. Identification of the motion direction is based on the fact that the spectrum of a non-blurred image is isotropic i.e. spreads out in all directions equally whereas the spectrum of a blurred image is anisotropic i.e. biased in the direction of the blur. Hough transform PSF estimation method works on an incremental voting method to compute the direction based on which blur length was calculated. When compared to ACF method, Hough estimate method was less affected by noise whereas the former suffered severe degradation in detection performance.

Further work could be simultaneous decision based on detection and estimation and discarding the bad frames dynamically. Also, detection can be extended to give more specific details as to whether the image suffered a motion blur or defocus blur and accordingly carry out the estimation part.

BIBLIOGRAPHY

1. Ouni, S., Chamnah, M., Herbin, M., Zagrouba, E., "Are Existing Procedures Enough? Image and Video Quality Assessment: Review of Subjective and Objective Metrics," in proceedings of SPIE, Farnand, S.P., Gaykema, F., Image Quality and System Performance V, 6808, pp. 68080Q-1 – 68080Q-11, 2008.

2. Joshi, N., Szeliski, R., Kreigman, D.J., "PSF estimation using sharp edge prediction," in IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8, 2008.

3. Cohen, E,. Yitzhaky, Y., "No-Reference assessment of blur and noise impacts on image quality," published in Springer London, Signal, Image and Video Processing, 2009.

4. Marzialono, P., Dufaux, F., Winkler, S., Ebrahimi, T., "A no-reference perceptual blur metric," in proceedings of IEEE international conference on Image Processing, Vol. 3, pp. 57-60, 2002.

5. Marichal, X., Ma, W.Y., Zhang, H., "Blur determination in the compressed domain using DCT information," in proceedings of IEEE international conference on Image Processing, Vol. 2, pp. 386-390, 1999.

6. Chong, R.M., Tanaka, T., "Image extrema analysis and blur detection with identification," in proceedings of IEEE international conference on Signal Image Technology and Internet based systems, pp. 320-326, 2008.

7. Tong, H., Li, M., Zhang, H., Zhang, C., "Blur detection for digital images using wavelet transform," in proceedings of IEEE international conference on Multimedia and Expo, Vol. 1, pp. 17-20, 2004.

8. Andrews, H,C., Hunt, B.R., "Digital Image Restoration," Englewood Cliffs, NY, Prentice-Hall, 1997.

9. Savakis, A.E., Easton Jr, R.L., "Blur identification based on higher order spectral nulls," in proceedings of SPIE, Image Reconstruction and Restoration, Vol. 2302, 1994.

10. Pao, T.L., Kuo, M.D., "Estimation of the point spread function of a motion-blurred object from autocorrelation," in proceedings of SPIE, Lance T.Wu., Visual Communications and Image Processing conference., Vol. 2501., 1995.

11. Yitzhaky, Y., Kopeika, N.S., "Identification of Blur Parameters from Motion Blurred Images," in Graphical Models and Image Processing, Vol. 59, No. 5, pp. 310-320, 1997.

12. Lokhande, R., Arya, K.V., Gupta, P., "Identification of Blur Parameters and Restoration of Motion Blurred Images," in proceedings of ACM symposium on applied computing, pp. 301-305, 2006.

13. Dash, R., Sa, P.K., Majhi, B., "RBFN based motion blur parameter identification," in proceedings of IEEE international conference on Advanced Computer Control, pp. 327-331, 2009.

14. Moghaddam, M.E., "A Mathematical Model to Estimate Out of Focus Blur," in proceedings of 5th IEEE international symposium on Image and Signal Processing and Analysis, pp. 278-281, 2007.

15. Moghaddam, M.E.," A Robust Noise Independent method to Estimate Out of Focus Blur," in proceedings of IEEE international conference on Acoustics, Speech and Signal Processing, pp. 1273-1276, 2008.

16. Nargesian, F., Darabi, A.A., Jamzad, M., "Center of Confsuion Estimation for Out of Focus Images based on Bispectrum," in IEEE international conference on Computational Intelligence and Multimedia Applications, Vol. 3, pp. 501-506, 2007.

17. Sakano, M., Suetake, N., Uchino, E., "A robust Point Spread Function estimation for Out-of-Focus Blurred and Noisy Images based on a distribution of Gradient vectors on the polar plane," in the Journal of Optical Society of Japan, co-published with Springer-Verlag GmbH, Vol. 14, No. 5, pp. 297-303, 2007.

18. Rooms, Filip., Pizurica, A., Philips, W., "Estimating image blur in the wavelet domain," in IEEE international conference on Acoustic, Speech and Signal Processing, Vol. 4, pp. , 2002.

19. Jain, A.K., "Fundamentals of Digital Image Processing," Prentice Hall International, Inc.

20. Mallat, S., Hwang, W.L., "Singularity detection and processing with wavelet," IEEE transactions on Information theory, pp. 617-643, 1992.

21. Tang, Y.Y., Yang, L., "Characterization and detection of edges by Lipschitz exponents and MASW wavelet transform," 14th international conference on pattern recognition, pp. 1572-1574, 1998.

22. Gonzalez, R.C., Woods, R.E., "Digital Image Processing," Prentice Hall, 2007.

23. Talukder, K.H., Harada, K., "Haar wavelet based approach for image compression and quality assessment of compressed image," published in the international journal of applied mathematics, Vol.36, No. 1, 2007.

24. Agarwal, S., “EE-345,Digital Image Processing, DFTs and Image transforms,” Department of Electrical Engineering, course material, 2009.

25. Tang, Y.Y., Yang, L., Liu, J., “Characterization of Dirac-Structure Edges with Wavelet Transform,” IEEE transactions on Systems, Man and Cybernetics, Part B, Vol.30, No. 1, pp. 93-109, 2000.

26. Crete, F., Dolmiere, T., Ladret, P., Nicolas, M., “The Blur Effect: Perception and Estimation with a New No-Reference Perceptual Blur Metric,” published in SPIE Human Vision & Electronic Imaging, Vol.6492, 64920I, pp. 64920I-1-64920I-011,2007.

27. Yitzhaky, Y., Mor, I., Lantzman, A., Kopeika, N.S., “Direct method for restoration of motion blurred images,” published in the journal of optical society of America, Vol.15, No.16, pp.1512-1518, 1998.

28. Agarwal, S., “Modeling and performance estimation for airborne minefield detection system, “ M.S thesis, Department of Electrical Engineering, University of Missouri, Rolla, MO, 2007.

29. Sanaka, A., “Spectral and spatial analysis of false alarms in airborne minefield data,” M.S thesis, Department of Electrical Engineering, Missouri University of Science & Technology, Rolla, MO, 2009.

30. Ganju, R., “Expectation maximization and its application in modeling, segmentation and anomaly detection,” M.S thesis, Department of Electrical Engineering, University of Missouri, Rolla, MO, 2006.

31. Juan, D., Yinglin, Yu., Shengli Xie., “ A new blind image quality assessment based on HVS,” published in the Journal of Electronics (China), Vol. 22, No.3, pp. 315-320., 2005.

32. Burningham, N., Pizlo, Z., Allebach, J.P., “Image quality metrics,” Image processing, Image quality, Image capture systems conference, 2003.

33. Li, X., “Blind image quality assessment,” proceedings of the international conference on image processing,” Vol.1, pp. 449-452, 2002.

34. Marais, I.v.Z., “On board image quality assessment for a satellite,” PhD dissertation, Department of Electrical and Electronic Engineering, University of Stellenbosch, Matieland, South Africa, 2009.

35. Marais, I.v.Z., Steyn, W.H., “Robust defocus blur identification in the context of blind image quality assessment,” published in the journal of Signal Processing : Image communication 22, pp. 833-844, 2007.

36. Rekleitis, I., "Visual motion estimation based on motion blur Interpretation," M.Sc thesis, School of computer science, McGill University, Montreal, Canada, 1995.

37. Hsu, P., Chen, B,Y., "Blurred image detection and classification," Advances in multimedia modeling, Vol. 4903, pp. 277-286, 2008, published by Springer Berlin/ Heidelberg, ISBN 978-3-540-77409-9.

38. Fabian, R., Malah, D., "Robust identification of motion and out-of-focus blur parameters from blurred and noisy images," published in CVGIP: Graphic models and Image processing, Vol.53, No.5, pp. 403-412, 1991.

39. Ouni, S., Chamnah, M., Herbin, M., Zagrouba, E., "Are Existing Procedures Enough? Image and Video Quality Assessment: Review of Subjective and Objective Metrics," in proceedings of SPIE, Farnand, S.P., Gaykema, F., Image Quality and System Performance V, 6808, pp. 68080Q-1 – 68080Q-11, 2008.

VITA

Harish Narayanan Ramakrishnan was born in Madras, India on January 22, 1986. In May 2007, he received his B.Tech degree in Electronics and Instrumentation Engineering from Vellore Institute of Technology University, Tamil Nadu, India. He started his Master’s degree in Electrical Engineering during the fall semester of 2008 and completed his Master’s degree in July 2010.