A Thesis

entitled

Low-Observable Object Detection and Tracking Using Advanced Image Processing Techniques

by

Meng Li

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Master of Science Degree in Computer Science

Dr. Ezzatollah Salari, Committee Chair

Dr. Junghwan Kim, Committee Member

Dr. Jackson Carvalho, Committee Member

Dr. Patricia R. Komuniecki,
Dean College of Graduate Studies

The University of Toledo

May 2014
An Abstract of

Low-Observable Object Detection and Tracking Using Advanced Image Processing Techniques

By

Meng Li

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Master of Science Degree in Computer Science

The University of Toledo
May 2014

Over the past few years, digital image processing has been widely studied and used in various fields. Image processing uses computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. In this thesis, we are going to introduce three important algorithms dealing with digital images: image denoising, image enhancement and target detection and tracking. The proposed Genetic Algorithm (GA) can detect and track dim, low observable and point targets, mainly for remote monitoring applications. As a first step to detect and track objects more effectively, the input image is first denoised and enhanced. We use Total Variation (TV) technique to remove the noise and improve the Signal to Noise Ratio (SNR) of the input image. To further enhance the image for outdoor applications a foggy image enhancement technique is introduced which significantly benefits traffic and outdoor
visual systems. Foggy image enhancement is an important branch of digital image processing, which is used when the weather is foggy. To overcome the shortcomings of the existing foggy image enhancement algorithms, we have developed a method that combines Principal Component Analysis (PCA), Multi-Scale Retinex (MSR) and Global Histogram Equalization (GHE). Initially, a PCA transform is applied to the foggy image to split the input image into a luminance and two chrominance components. In the second step, the luminance and the chrominance components are individually enhanced by MSR and GHE, respectively. In the final stage, an inverse PCA is applied to combine the results of the three channels into a new RGB image.

To detect and track low observable targets in a digital image sequence, an encoding scheme along with genetic operation is designed to track the targets. To avoid missing any tracks, individual preservation method is introduced to maintain the more promising candidate tracks. Target trajectories are then confirmed by a multi-stage hypothesis testing scheme.
I sincerely thank my advisor Dr. Ezzatollah Salari for giving me the opportunity to pursue my research under his guidance. It has been a great learning experience working with him. I would also like to express my deepest gratitude towards him for offering instructive advice at all times, treating me with endurance, and constantly pushing me to do better every time.

Last, but not the least, I thank my family, and friends who have been a constant source of encouragement and support for me throughout my journey in the Master’s Degree program.
Table of Contents

Abstrack ........................................................................................................................................ iii
Acknowledgements ......................................................................................................................... v
Table of Contents .......................................................................................................................... vi
List of Tables ................................................................................................................................... ix
List of Figures ............................................................................................................................... x
List of Abbreviations ..................................................................................................................... xi
List of Symbols ................................................................................................................................ xii
1 Introduction ................................................................................................................................... 1
  1.1 Background ............................................................................................................................. 1
  1.2 Motivation of the Research ...................................................................................................... 3
  1.3 Organization of the Thesis ....................................................................................................... 4
2 Literature Review .......................................................................................................................... 5
  2.1 Existing Image Denoising Methods ......................................................................................... 5
  2.2 Existing Image Enhancement Methods .................................................................................. 7
  2.3. Review of Vehicle Detection and Tracking Algorithms .................................................... 12
3 Image Denoising using Total Variation ....................................................................................... 14
  3.1. The Mathematical Model of the Total Variation: ................................................................. 15
  3.2. The Implementation of the Total Variation: ......................................................................... 17
4 Image Enhancement Based on Principal Component Analysis .................................................. 19
4.1 Image Transform using Principal Component Analysis .............................. 21
4.2 Luminance Component Enhancement Using Multi-Scale Retinex .............. 23
4.3 Chrominance Component Enhancement Using GHE ................................. 26
5 Low Observable Targets Detection and Tracking Using Genetic Algorithm .... 29
  5.1 Encoding Scheme and Design of Genetic Operators .............................. 29
    5.1.1 The structure and storage of candidate tracks .............................. 29
    5.1.2 Encoding ............................................................................ 30
    5.1.3 Crossover ........................................................................... 31
    5.1.4 Mutation ............................................................................ 31
    5.1.5 Fitness function .................................................................... 31
    5.1.6 Qualified individual preservation ............................................ 32
    5.1.7 Selection operation ................................................................. 33
  5.2 Multi-Stage Hypothesis Testing .......................................................... 33
  5.3 Algorithm Description ....................................................................... 36
6 Simulation Results ....................................................................................... 38
  6.1 Simulation results for the Image Denoising .......................................... 38
  6.2 Simulation results for the Image Enhancement ...................................... 39
  6.3 Simulation results for the Target Detecting and Tracking ...................... 41
7 Conclusion and Future Work ...................................................................... 45
References ...................................................................................................... 47
A  Source Code for Total Variation Image Denoising .................................. 52
B  Source Code for Image Transform using PCA ....................................... 54
C  Source Code for Luminance Component Enhancement Using MSR ......... 59
D  Source Code for Chrominance Component Enhancement Using GHE.............. 63
E  Source Code for Target Detection.............................................................. 66
F  Source Code for Target Tracking............................................................... 70
List of Tables

6.1 Parameters associated with the genetic algorithm ........................................ 43
6.2 Other simulation parameters ......................................................................... 44
List of Figures

1-1 Example of a noisy sine and its denoised version ........................................ 2
3-1 Original image and its spatial distribution characteristics.......................... 15
3-2 Example of a real-valued continuous function to show total variation .......... 16
4-1 Flowchart of the image enhancement system ............................................ 21
4-2 Image decomposition using PCA algorithm ............................................. 23
4-3 Results for the GHE enhancement .......................................................... 28
5-1 A structure of chromosome ....................................................................... 31
5-2 Tree data structure of candidate tracks trajectory ..................................... 36
5-3 Flowchart of the target detection and tracking algorithm ......................... 39
6-1 The comparison of spatial distribution characteristic before and after denoising ...... 41
6-2 Results for the proposed image enhancement system .................................. 42
6-3 The initial tracks .......................................................................................... 44
6-4 The confirm tracks ....................................................................................... 44
6-5 The first frame image in the image sequence ............................................ 45
6-6 Detection results based on a single frame ................................................ 45
6-7 Detection results based on association of image frames ............................. 45
List of Abbreviations

CLAHE………………Contrast-Limited Adaptive Histogram Equalization
MAE………………Mean Absolute Error
MSR……………….Multi-Scale Retinex
NHTSA……………National Highway Traffic Safety Administration
NTSB………………National Transportation Safety Board
PCA……………….Principal Component Analysis
PSNR……………..Peak Signal to Noise Ratio
RGB……………….Red, Green and Blue
RMSE……………..Root Mean Square Error
SNR……………….Signal to Noise Ratio
SSR……………….Single-Scale Retinex
TV……………….Total Variation
GA……………….Genetic Algorithm
GHE………………Global Histogram Equalization
List of Symbols

\( f \) .................. Continuous function, desired clean image
\( f_0 \) .................. Noisy image
\( n \) .................. Random noise with a mean equal to 0 and variance equal to \( \sigma^2 \)
\( \Omega \) .................. The set of all pixels in the image
\( \frac{1}{|\nabla f|} \) .................. Diffusion coefficient
\( f_{i,j} \) .................. Gray value of the pixel in an image \( f \)
\( h \) .................. Spatial increment
\( \Delta t \) .................. Time step
\( n \) .................. Index-cycle
\( \lambda \) .................. Reciprocal of the image gradient threshold.
\( \sigma \) .................. Standard deviation for the Gaussian function
\( * \) .................. Convolution operation
\( T \) .................. Gray transformation function
\( P(p), P(q) \) .................. Probability of the gray value \( p, q \)
\( \text{Int} \) .................. Rounding operator
\( t_x, t_y \) .................. Coordinates of the initial point
\( m \) .................. Location of the next point on the track
\( \alpha \) .................. False alarm rate
\( \beta \) \text{ Miss detection rate}

\( N \) \text{ Gaussian probability density function}

\( \tau \) \text{ Threshold}

\( \Phi \) \text{ Standard normal distribution}
Chapter 1

Introduction

1.1 Background

Over the past few years, digital image processing has been widely studied and used in various fields. Image processing uses computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. In this thesis, we are going to introduce three important algorithms for image enhancement and tracking, namely, image denoising, image enhancement and target detection and tracking.

Fog and low stratus cloud is a hazard to public and personal transportation. Visibility under foggy conditions can be drastically reduced, and it will create dangerous situations for vehicles on roadways as well as airplanes, trains, boats, and other means of transport. From 1995 to 2005 the National Highway Traffic Safety Administration (NHTSA) determined an average of 38,700 vehicular accidents, resulting in about 16,300 injuries and 600 deaths, were directly related to fog each year in the United States. In the same time period, the National Transportation Safety Board (NTSB) reported an annual
average of 81 airplane crashes, 61 of which resulted in at least one fatality, caused by reduced visibility due to clouds, fog, or low ceiling. In addition, millions of dollars are lost each year by commercial airlines from cancellation, delay, and rerouting forced by low visibilities at airports due to fog and low stratus cloud.

It is well known to any scientist and engineer who work with a real world data that signals do not exist without noise, which might be negligible (i.e. high SNR) under certain conditions. However, there are many cases in which the noise corrupts the signals in a significant manner, and it must be removed from the data in order to proceed with further data analysis. The process of noise removal is generally referred to as signal denoising or simply denoising. An example of a noisy signal and its denoised version can be seen in Figure 1-1. It can be seen that the noise adds high-frequency components to the smooth original signal. This is a characteristic effect of noise.

![Figure 1-1 Noisy sine and its denoised version (solid line)](image)

Detecting and tracking techniques are widely used in GPS, NASA, traffic monitoring and other important fields. An accurate and effective detection method will help government save a lot of money and time.
1.2 Motivation of the Research

This thesis presents advanced techniques in image processing dealing with image enhancement, image denoising and object detection and tracking. In the literature many image enhancement techniques such as gamma correction, contrast stretching, histogram equalization, and Contrast-limited adaptive histogram equalization (CLAHE) have been discussed. These techniques have their limitations and will not be able to provide totally enhanced images and often give poor performance in terms of Root Mean Square Error (RMSE), Peak Signal to Noise Ratio (PSNR) and Mean Absolute Error (MAE). Most of these enhancement techniques will not recover exact true color of the images. Recently, Retinex, Homomorphic and Wavelet Multi-Scale techniques have been popular for enhancing images. These methods are shown to perform much better than those listed earlier.

Detection and tracking of low visible targets (dim objects) are considered to be very important areas in image and video analysis field. This technology can be used in many different fields such as surveillance (e.g. Oil spills), search and rescue, remote sensing, floating mine detection, and so on. Earlier research indicates that there are many effective algorithms for detection and tracking of large targets in a scene. However, there has been a limited effort for detection of smaller targets with low visibility.

In this thesis an effective and fast algorithm to detect and track tiny low observable targets in a digital image sequences has been proposed. To be more precise, we use image enhancing and noise reduction system as a filter to improve the Signal to Noise Ratio (SNR) and remove significant amount of noise before any further processing. We have also designed an encoding scheme and genetic operations to detect all possible
tracks. Finally, the individual preservation is implemented to avoid missing any target track. Target trajectories are then confirmed by a multi-stage hypothesis testing. The simulation result shows that the proposed scheme can efficiently detect and track small targets with an SNR value under 2db.

1.3 Organization of the Thesis

The primary goal of this thesis is to develop a detection and tracking system based on image processing techniques. The overall process consists of two modules: (1) image enhancement and noise reduction, (2) target detection and tracking. The rest of the thesis is organized as follows:

In Chapter 2, a brief review of some of the commonly used image enhancement, noise reduction, object detection, object tracking methods are presented.

In Chapter 3, a noise reduction system is proposed to denoise the input image sequence.

In Chapter 4, a PCA based image enhancement algorithm is proposed.

In Chapter 5, a GA based low observable target detection and tracking systems are presented.

In Chapter 6, simulation results of the above algorithms are given illustrating the effectiveness of the proposed systems.

In Chapter 7, conclusions are drawn.
Chapter 2

Literature Review

During the past few years, several technologies have been developed using a variety of concepts and approaches for noise reduction, image enhancement vehicle detection and vehicle tracking.

2.1 Existing Image Denoising Methods

Visual information transmitted in the form of digital images is becoming a major method of communication in the modern age. However, the image obtained after transmission are often corrupted by noise. Therefore, polluted images need to be processed before it can be used in the following steps. Image denoising includes the manipulation of the image data to produce a visually high quality image. This chapter reviews the existing denoising algorithms, such as filtering approach, wavelet based approach and provides a comparison of various approaches. Different noise models such as Gaussian noise, salt and pepper noise, speckle noise and Brownian noise as well as additive and multiplicative types are considered. How to select denoising algorithm is application dependent and it is based on the type of noise. Therefore, it is necessary to have knowledge about the noise present in the image so as to select the appropriate denoising algorithm. When the image is corrupted with salt and pepper noise, the filtering
approach has been proved to be the best. The wavelet and the total variation based approach are usually used in denoising images corrupted with Gaussian noise. In the case where the noise characteristics are complex, a more comprehensive approach can be used. We would like to list some methods which are widely used, Pei-Yung Hsiao presents a generic two-dimensional (2-D) Gaussian smoothing filter for noise image processing. The filter includes the power-of-two approximation arithmetic algorithm for the Gaussian coefficients and effective hardware design. The proposed generic Gaussian smoothing filter is able to provide various levels of noise smoothing and reduction, which are highly desired in early stages of the image processing. By using the power-of-two terms, the generic 2-D Gaussian filter can be implemented by using simple hardware shifters and adders. An embedded multi-access SRAM concept is also adopted to raise the hardware throughput. The synthesized hardware of such combination exhibits that the proposed filter can attain a real-time high frame-rate and high-resolution video processing [1]. Witkin introduced the scale-space technique which involves generating coarser resolution images by convolving the original image with a Gaussian kernel. This approach has a major drawback: it is difficult to obtain accurately the locations of the “semantically meaningful” edges at coarse scales. Pietro P. et al. suggested a new definition of scale-space and introduced a class of algorithms that is realized by using a diffusion process [2]. Rudin L. et al proposed a nonlinear total variation based noise removal algorithm. A constrained optimization type of numerical algorithm for removing noise from images is presented. The total variation of the image is minimized subject to the constraint which includes the statistics of the noise. The constraints are imposed using Lagrange multipliers and the solution is obtained using the gradient-projection method. This
amounts to solving a time dependent partial differential equation on a manifold determined by the constraints. As \( t \) approaches infinity, the solution converges to a steady state which is the denoised image. The numerical algorithm is simple and relatively fast. The results appear to be state-of-the-art for very noisy images. The method is noninvasive, yielding sharp edges in the image. The technique could be interpreted as a first step of moving each level set of the image normal to itself with velocity equal to the curvature of the level set divided by the magnitude of the gradient of the image, and a second step which projects the image back onto the constraint set [3]. In addition, other methods such as the Yaroslavsky, neighborhood filters [4-5] and an elegant variant, the SUSAN filter (Smith and Brady) [6], and the Wiener local empirical filter as implemented by Yaroslavsky [7]. Techniques based on wavelet thresholds have also been investigated [8-10].

### 2.2 Existing Image Enhancement Methods

The principal objective of image enhancement is to modify attributes of an image to make it more suitable for a given task and a specific observer. During this process, one or more attributes of the image are modified. The choice of attributes and the way they are modified are specific to a given task. Moreover, observer-specific factors, such as the human visual system and the observer's experience, will introduce a great deal of subjectivity into the choice of image enhancement methods.

In this section we present some of the recent image enhancement techniques for image processing from the point view of physical model and digital image processing. Some fundamental principles of typical methods are summarized and the state-of-the-art progress is presented. In the case of foggy image enhancement, some new haze removal
algorithms, both the perceptual visual effect and objective evaluation data are reviewed to illustrate their ability to remove the haze.

Image enhancement problem can be formulated as follows: given an input low quality image, it is required to provide a high quality output image for specific applications.

Image enhancement algorithms offer a wide variety of approaches for modifying images to achieve visually acceptable images. The choice of such techniques is a function of the specific task, image content, observer characteristics, and viewing conditions. There exist many techniques that can enhance a digital image which can broadly be divided in to the following two categories:

1. Spatial Domain Methods
2. Frequency Domain Methods

In spatial domain techniques, we directly deal with the image pixels. The pixel values are manipulated to achieve desired enhancement. In frequency domain methods, the image is first transferred in to frequency domain. It means that, the Fourier Transform of the image is computed first. All the enhancement operations are performed on the Fourier transform of the image and then the Inverse Fourier transform is performed to get the resultant image. These enhancement operations are performed in order to modify the image brightness, contrast or the distribution of the grey levels. As a consequence the pixel value of the output image will be modified according to the transformation function applied on the input values.

The point processing methods are most primitive, yet essential image processing operations and are used primarily for contrast enhancement.
Carrying out image enhancement of low quality images is a challenging problem because of the following reasons: Due to low contrast, we cannot clearly extract objects from the dark background and most color based methods will fail on this matter if the color of the objects and that of the background are similar. Moreover, Image enhancement is to improve the interpretability or perception of information in images to provide a better input for other automated image processing steps. The image acquired from natural environment with high dynamic range includes both dark and bright regions. Due to the high dynamic range of human eyes sensing, those image are difficult to be perceived by human eyes. Therefore, Image enhancement is a common tool used to improve the quality of those images in terms of human visual perception.

Cristobal et al. [11] presented a Gabor Filter based image enhancement method to remove degradations due to large amounts of space and frequency variant fractal noise. Images degraded by fog are examples of such a situation. The enhancement process is recursively applied at the different pyramid levels. The method has been tested by merging 90% of a fractal cloud cover with 10% of an aerial image, illustrating much of details practically invisible in the foggy image become clearly visible. Srinivasa G. et al. [12] addressed the question of de-weathering a single image using simple additional information provided interactively by the user. They exploited the Physics-based models described in prior work and develop three interactive algorithms to remove weather effects. They demonstrated effective color and contrast restoration using several images taken under poor weather conditions. Furthermore, they showed an example of adding weather effects to images. Their interactive methods for image de-weathering can serve as easy-to-use plug-ins for a variety of image processing software. Robby. et al. [13]
developed a foggy image enhancement method based on color and intensity information. It enhanced the visibility after estimating the color of skylight and the values of airlight. Its experimental results on real images showed the effectiveness of the approach. Kokkeong, et al. [14] presented a physics degradation image enhancement method in which the parameters required are estimated from the image itself. The proposed method is tested using synthetic images to explore the limitations and reliability of the method under different visibility conditions. Enhancement is performed on real images taken using an airborne camera at a height of approximately 1000 meters in hazy conditions for which the visibility is approximately 10 kilometers. Significant improvements in terms of contrast, visible range and color fidelity are evident when compared to existing methods.

Nicolas Hautiere et al. [15] presented an image enhancement assessment algorithm which consists of computing the ratio between the gradient of the visible edges between the image before and after contrast restoration. In this way, an indicator of visibility enhancement is provided based on the concept of visibility level, commonly used in lighting engineering. Finally, the same methodology is applied to contrast enhancement assessment and to the comparison of tone-mapping operators. Raanan Fattal et al. [16] proposed a new method for rendering high dynamic range images on conventional displays, which is conceptually simple, computationally efficient, robust and easy to use. They manipulated the gradient field of the luminance image by attenuating the magnitudes of large gradients. A new, low dynamic range image is then obtained by solving a Poisson equation on the modified gradient field. Their results demonstrated that the method is capable of drastic dynamic range compression, while preserving fine details and avoiding common artifacts, such as halos, gradient reversals,
or loss of local contrast. This method is also able to significantly enhance ordinary images by bringing out detail in dark regions. Hau Nge et al. [17] developed a visibility improvement system for helping drivers with poor vision during night and bad weather conditions. A novel image enhancement technique employing a nonlinear expansion function is developed for enhancing the fused images. The slope characteristics of the expansion function at an individual pixel location is based on the statistical properties of its neighborhood pixels. The expansion function is also capable of reducing the intensity of the overly bright image regions due to the presence of head lights from vehicles in opposite direction. The enhanced video stream is then subjected to a color restoration process to introduce natural color using the color information gathered from the CCD camera image. The entire image processing and analysis system was being installed in an FPGA environment for performing the processing in real-time. The processed video stream is displayed on a Head-Up Display located directly in front of the driver to assist him/her to drive safely in poor visibility environments. Preliminary results obtained in various experiments conducted with the proposed system are encouraging. M. Wilscy et al. [18] demonstrated a novel method that not only enhanced visibility but also maintained the color fidelity. This method consists of three phases. The first two phases consists of estimating a measure of degradation and its removal while the final phase used a novel wavelet fusion method to obtain the enhanced image. Performance analysis was carried out with the help of a contrast improvement index and sharpness measure. The experimental results on real images showed the effectiveness of the approach. Jing Yu et al. [19] proposed a novel fast defogging method from a single image of a scene based on a fast bilateral filtering approach.
Visible light image is a kind of data form which is most close to the human visual experience. Since its characteristic is that it is easy to access and does not require extra sensors, the image becomes the most common data input form for outdoor visual system. In recent years, the research on the atmospheric degradation of visible light image enhancement keeps heating up, and there has been dramatic achievement. Therefore, this thesis proposed a method based on PCA algorithm to enhance the foggy image. This method is focused on the research of visible light image enhancement and tries to find a better contrast enhancement algorithm to recover detail information in a scene from the degraded images. The complexity of our method is a linear function of the input image pixels which allows a very fast implementation. Results on a variety of outdoor foggy images demonstrate that our method achieves good restoration for contrast and color fidelity, resulting in a large improvement in image visibility.

2.3. Review of Vehicle Detection and Tracking Algorithms

There has been a significant amount of research in developing automated vehicle detection and tracking systems. Avidan [20] designed Support Vector Tracking system, which utilizes an offline-learned support vector machine as the classifier and embeds it into an optical-flow based tracker. Broggi et al. [21] developed a multi-resolution vehicle detection system which uses the symmetry features to locate the vehicle. Bertozzi et al. [22] described a stereo vision-based vehicle detection system that can be implemented on the ARGO vehicle. Zhu et al. [23] defined an object tracking system using SVM regression method. Rasekhi et al. [24] developed a supervised learning method based on wavelet transform to perform the airplane tracking.
Lim et al. [25] demonstrated monocular lane-vehicle detection and tracking system comprising of lane boundary detection, lane region tracking and vehicle detection with vertical asymmetry measurement. A critical survey of recent vision-based on-road vehicle detection system was presented by Sun et al. [26]. Song et al. [27] presented a monocular machine vision system capable of detecting vehicles in the front view or behind of a vehicle. Bensrhair et al. [28] presented a comparison of the model vehicle detection and the stereo vision based vehicle detection system. In traffic applications, Zhang et al. [29] proposed a framework to analyze the traffic video sequence using unsupervised vehicle detection method.
Chapter 3

Image Denoising using Total Variation

In this chapter, we describe an effective image denoising method. The method is based on total variation (TV) which was originally proposed by Rudin Osher and Fatemi. Aim at Gaussian noise, the denoising results of linear filter are better than mid-value filter, and the images denoised by TV algorithm is clearer than the images which are denoised by wavelet filter. Although the details of the edges in the denoised images appear to be blurred, the mean square error of TV algorithm is considerably low which means it has a better denoising function.

Reducing the total-variation of the image means filtering out noise while preserving edges. This will also remove textures and fine-scale details. It is assumed that the noise is a white Gaussian noise with a-priori known (or estimated) noise power (variance). The fidelity term used to the input image is calculated automatically so that the power of the noise is reduced.

Texture preserving TV with an adaptive fidelity term, reduces the total-variation of the image selectively, generalizing the process to adaptive power constraints. Denoising is strong in smooth regions and weaker in textured regions. Therefore, it preserves the texture and fine-scale details. This is a two-phase process where the noise
and textures are first isolated by scalar TV. The adaptive process then imposes local power constraints based on local variance measures of the first phase.

The SNR is very low in these images so the targets are hardly detectable by naked eye. This can be seen in the following figures where the two targets can hardly be noticed, even in the 3D distribution.

Figure 3-1 Original image and its spatial distribution characteristics

3.1. The Mathematical Model of the Total Variation:

In mathematics, the total variation indicates several similar concepts, which is related to the local or global structure of the codomain of a function. For a real-valued continuous function, $f$, shown in Figure 3-2, the total variation of a function can be simply explained as the length of the path travelled by that ball's projection on the y-axis when a ball travels on the graph of a given function.
Let $f$ be the desired clean image, $f_0$ be the noisy image. Therefore, the following relationship is true.

$$f_0(x, y) = f(x, y) + n(x, y), \quad (x, y) \in \Omega$$

where $n$ is the random noise with a mean equal to 0 and variance equal to $\sigma^2$ and $\Omega$ is the set of all pixels in the image, $(x, y) \in \Omega$. The denoising method based on total variation seeks for an image $f$ with a minimum total variation as follows,

$$\min TV(f) = \int_{\Omega} \sqrt{\nabla f^2} \, dx \, dy = \int_{\Omega} \sqrt{f_x^2 + f_y^2} \, dx \, dy$$

which is subject to the following mean and variance constraints,

$$\int_{\Omega} f \, dx \, dy = \int_{\Omega} f_0 \, dx \, dy$$

$$(3.3a)$$

$$\frac{1}{|\Omega|} \int_{\Omega} (f - f_0)^2 \, dx \, dy = \sigma^2$$

$$(3.3b)$$

Usually, the total variation of image with noise is higher than the total variation of image without noise, so minimizing total variation can remove the noise, in the image. The above equations can be reformulated as,

$$\min TV(f) = \frac{\lambda}{2} \int_{\Omega} (f - f_0)^2 \, dx \, dy + \int_{\Omega} \sqrt{f_x^2 + f_y^2} \, dx \, dy$$

$$(3.4)$$
In equation (3.4), the first part represents the fidelity maintaining the characteristic of the original image and reducing the distortion. The second part is the regularization term and it is used to provide a smooth result, and it is based on the SNR value. Following the Euler-Lagrange equation, we have the following,

$$-\mathbf{\nabla} \cdot \left( \frac{\mathbf{\nabla} f}{|\mathbf{\nabla} f|} \right) + \lambda (f - f_0) = 0$$  \hspace{1cm} (3.5)

In equation (3.5), $\frac{1}{|\mathbf{\nabla} f|}$ is the diffusion coefficient. Note that, at the edge points of the image, $|\mathbf{\nabla} f|$ is high and the diffusion coefficient is small. Therefore, after denoising, the edges could be retained. In contrast, at the smooth parts of the image, $|\mathbf{\nabla} f|$ is small and the diffusion coefficient is high, leading to the removal of the noise.

### 3.2. The Implementation of the Total Variation:

Let $f_{i,j}$ represents the gray value of the pixel $(x_i = i \cdot h, y_i = j \cdot h)$ in an image $f$ with $i, j = 0, 1, \ldots N$ and $h$ is the spatial increment. Further, let $f(x_i, y_i, t_n) = f_{i,j}^n$ represents the value in $n$-th index-cycle, where, $t_n = n \cdot \Delta t$ and $\Delta t$ is the time step. We use steepest descent method to implement TV denoising algorithm, the diffusion term can be written as,

$$\mathbf{\nabla} \cdot \left( \frac{\mathbf{\nabla} f}{|\mathbf{\nabla} f|} \right) = \frac{f_x^2 f_{xx}^n - 2 f_x f_y f_{xy}^n + f_y^2 f_{yy}^n}{f_x^2 + f_y^2}$$  \hspace{1cm} (3.6)

Using the difference equations to replace partial derivative, we have,

$$(f_x)_i^n = f_{i+1,j}^n - f_{i-1,j}^n$$
$$(f_y)_i^n = f_{i,j+1}^n - f_{i,j-1}^n$$
$$(f_{xx})_i^n = f_{i+1,j}^n - 2 f_{i,j}^n + f_{i-1,j}^n$$
$$(f_{yy})_i^n = f_{i,j+1}^n - 2 f_{i,j}^n + f_{i,j-1}^n$$
$$(f_{xy})_i^n = f_{i+1,j+1}^n + f_{i-1,j-1}^n - f_{i-1,j+1}^n - f_{i+1,j-1}^n$$  \hspace{1cm} (3.7)
Therefore, the discrete version of equation (3.5) can be expressed iteratively as shown below:

\[ f_{i,j}^{n+1} = f_{i,j}^n - \Delta t \lambda (f_{i,j}^n - f_{i,j}^0) + \Delta t \left( \nabla \cdot \left( \frac{\nabla f_{i,j}^n}{|\nabla f_{i,j}^n|} \right) \right) \]  

(3.8)

In equation (3.7), (3.8), \( n \) is the iteration number and \( i, j = 0, 1 \ldots N \). Boundary conditions satisfies \( f_{0,j}^n = f_{1,j}^n;  f_{N,j}^n = f_{N-1,j}^n \).

To simplify the calculations, we defined \( \lambda \) as the reciprocal of the image gradient threshold. In summary, the implementation of this algorithm involves the following steps:

(1) Input the noisy image \( f_0 \).

(2) Initialize parameters: \( n = 0, \Delta t = 0.25, \ h=I, \ f^0 = f_0, \ \nabla \cdot \left( \frac{\nabla f_{i,j}^n}{|\nabla f_{i,j}^n|} \right) = 0. \)

(3) While \( n < \) maximum index-cycle, repeat Steps 3.1 and 3.2.

   Step 3.1: \( n = n + 1 \), calculate \( f_{i,j}^n \) based on equation (3.8).

   Step 3.2: using equations (3.6) and (3.7),

   \[ \text{calculate diffusion term } \nabla \cdot \left( \frac{\nabla f_{i,j}^n}{|\nabla f_{i,j}^n|} \right). \]

(4) End.

The result in the last cycle is the denoised image.
Chapter 4

Image Enhancement Based on Principal Component Analysis

In this chapter, we present an image enhancement which is capable of compensating for poor visibility caused by adverse and changing weather conditions that can result in particularly severe traffic accidents involving many vehicles and causing many injuries. Although not the only cause of reduced visibility, fog is the most familiar one. Fog is a common natural phenomenon, which can even take place in sunny summer weather, because of the evaporation of water. It can reduce the visibility in the atmosphere and influence aviation, which can cause accidents to almost every kind of transportation modes. Physically speaking, fog negatively influences human perception of visibility due to scattering atmospheric particles. Part of the reflected light from objects is reflected and absorbed by water molecules, leading to the attenuation of the incident light reaching the observer. Fog can also make the images blur when captured by optical equipment; thus information extraction from the images will be affected. In foggy images, the dark components have higher gray values meanwhile the light components have lower gray values. Thus the distribution of the pixel gray values is too concentrated and the contrast of images is reduced. Additionally, in the image transfer process, inevitably, image quality will be deteriorated by the influence and interference of various
noises, such as fog and rain. Therefore, there is an urgent need for an efficient foggy image enhancement algorithm.

Image enhancement is not based on the principle of fidelity. Therefore, without considering degradation, an image can simply highlight certain information according to a specific requirement. Meanwhile, the unnecessary information is removed. The main purpose for this procedure is to improve the discernibility degree of image, make it more suitable for human vision and computer recognition systems. However, it is difficult to implement the image enhancement method in the existing outdoor visual systems due to the variety of noises.

The proposed method for foggy image enhancement as depicted in Figure 4-1 consists of the following three main modules:

1. The PCA algorithm is applied to transform the RGB image into a more independent luminance and chrominance components.
2. Apply MSR to enhance the luminance component and GHE to the two chrominance components, respectively.
3. Inverse PCA transform is used to reconstruct the output image.
4.1 Image Transform using Principal Component Analysis

Principal Component Analysis (PCA) is a powerful tool for data analysis and can be used to express the data in such a way as to highlight their similarities and differences. PCA is a useful statistical technique that has found application in fields such as face recognition and image compression. Furthermore, it is a common technique for finding patterns in data of high dimension, where the luxury of graphical representation is not available.

PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called first principal component), the second greatest variance on the second coordinate and so on. Thus, PCA is concerned with explaining the
variance and covariance structure of a high-dimensional random vector through a few linear combinations of the original component variables. The objective of PCA is to reduce the dimensionality (the number of variables) of the dataset while retaining most of the original variability in the data.

A common way to find the principal components of a data set is by calculating the eigenvectors of the data covariance matrix. These eigenvectors give the directions in which the data distribution is stretched most. The projections of data on the eigenvectors are the principal components. To proceed with the PCA transformation, the input image is considered after subtracting its mean value. Following this step, the covariance matrix, eigenvalues and eigenvectors are calculated. The eigenvectors are re-arranged in descending order of eigenvalues. A selected set of eigenvectors can be used to multiply the original data matrix to generate the final matrix.

In the proposed algorithm, the PCA is used as an image transformation tool instead of a dimension reduction tool. In this way the image is decomposed into luminance and chrominance descriptors, which can be used separately to further enhance the image. Figures 4-2 (a, b, c, d) shows an example of image decomposition using PCA algorithm, where, Figure 4-2 (a) is the original image, Figure 4-2 (b) is the luminance component, and Figure 4-2 (c) and (d) are the two chrominance components.
Figure 4.2 image decomposition using PCA algorithm

(a) Original image (b) Luminance component (c, d) Chrominance component

4.2 Luminance Component Enhancement Using Multi-Scale Retinex

The Retinex theory of color, proposed by Edwin Land in 1971, offered an explanation of our ability to perceive color in ambient-colored environments. Under a perfectly uniform illumination, the color perceived by the human eye can be considered to be the product of the object reflectance and the incident light. Therefore, the reflectance can be calculated by estimating the illuminant component from the perceived color. However, this is impossible in the case of real scenes, because the illuminant is never perfectly uniform and needs to be regionally estimated.
The main concept of Retinex theory is to replace the original image to the reflectance component, which can be calculated by the difference between the original image and the Gaussian filtered image in logarithmic space. The procedure of this algorithm is defined as following:

The Retinex theory first assumes that an image is composed of the incident component and the reflection component, as expressed in the following equation,

\[ O(i, j) = R(i, j) \cdot I(i, j) \]  \hspace{1cm} (4.1)

where \( O(i, j) \) is the original image, \( R(i, j) \) and \( I(i, j) \) represents the reflection and incident components, respectively.

The luminance estimate value \( I_{low}(i, j) \) is obtained through preprocessing \( I(i, j) \) by a Gaussian filter in the logarithm domain as shown in equation (4.2) and (4.3).

\[ I(i, j) \approx I_{low}(i, j) = O(i, j) * F(i, j) \]  \hspace{1cm} (4.2)

\[ F(i, j) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i^2 + j^2)}{2\sigma^2}\right) \]  \hspace{1cm} (4.3)

where \( F(i, j) \) is the Gaussian filter, \( \sigma \) represents the standard deviation for the Gaussian function, and the symbol ‘\(*\)’ denotes the convolution operation. Determining the standard deviation of the Gaussian filter, \( \sigma \), is very important, as the performance of a single-scale Retinex depends on this parameter. The small scale \( \sigma \) can dramatically enhance the detail in the dark part of the image, but it suffers from the color distortion. In contrast, the large scale \( \sigma \) leads high level of color fidelity but it is weak in enhancement.

Finally, the reflection component \( R'(i, j) \) is:

\[ R'(i, j) = O(i, j) / I_{low}(i, j) \]  \hspace{1cm} (4.4)

Based on the theory of the single scale Retinex, the idea of a Multi-Scaled Retinex (MSR) was introduced to improve the enhancement result. MSR can use multiple
scale values to enhance the image, which can improve both the image dynamic range compression and the quality of color in the reproduction. The Result of MSR is a Gaussian filter with various scales which are averaged with different weights using the following computation:

\[ R(i,j) = \sum_{i=1}^{k} w_i \left[ \log O(i,j) - \log [O(i,j) \ast F_t(i,j)] \right] \]  \hspace{1cm} (4.5)

where \( w_i \) represents the weight for the \( i \)th scale, \( k \) is the number of scales (normally defined as 3). While the result of a single Retinex using a small scale Gaussian filter only includes the detail with graying-out, the result of a multi scale Retinex using a large scale Gaussian filter includes more chromaticity information. Thus, the local contrast and color rendition can be simultaneously obtained based on a weighted summation of these results.

The selection of scale parameters is a crucial part in the MSR algorithm, which has been discussed in many related articles. The four common types of selections list as follows: Daniel Jobson et al. [30] defined the middle-scale as \( 50 < \sigma < 100 \) and ignore the small and big scale. Bo Sun et al. [31, 32] defined the three scales values as 5, 20, 240. Li Tao et al. [33] and Qian Liu et al. [34] define that 1-5% of original images use small-scale, 10-15% of original images use middle-scale, 30-50% use large-scale. Peng Jiaqi et al. [35] define the first scale as the mean-standard deviation of the original image meanwhile the second scale is the result of mean-standard deviation adding 0.5.

Solving the above problems requires strict mathematical deduction, demonstration, and an abundance of experiments. We conducted a series of tests starting with only two scales and adding further scales as needed. The key to choosing the parameters is making a balance between the processing time and quality. Based on our experimental results, the parameters of the proposed method are 15, 80, 235.
4.3 Chrominance Component Enhancement Using GHE

Histogram Equalization (HE) is a technique that generates a gray map which changes the histogram of an image and redistributing all pixels values to be as close as possible to a desired value. This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram which increases the contrast value in the low contrast area. Since it is computationally fast and simple to implement, histogram equalization is a technique commonly used for image contrast enhancement.

An image which is captured in foggy weather usually has low contrast and its distribution of histogram is more centralized than image captured in normal weather. After histogram equalization, the original image is changed into a new image with uniform gray value distribution. Unlike Retinex algorithm, Global Histogram Equalization (GHE) is based on the brightness constancy, so compared with color restoration, GHE is better at sharpening edges and increasing details. Therefore, we choose GHE to enhance the chrominance component.

The procedure of GHE is discussed as following: first, let’s define $f(x, y)$ as original image, $g(x, y)$ as enhanced image, $T$ is the transformation function.

$$g(x, y) = T[f(x, y)]$$  \hspace{1cm} (4.6)

Assume $p$ is the gray value of $f(x, y)$ in the original image $f$, and $q$ is the gray value of $g(x, y)$ in the enhanced image $g$ and $T$ is the gray transformation function according to gray distribution of the original image. Equation (4.6) can be rewritten as,

$$q = T(p)$$  \hspace{1cm} (4.7)
The inverse transformation of (4.7) is,

\[ p = T^{-1}(q) \]  

(4.8)

Further, let’s assume the probability of the gray value \( p \), \( q \) be \( P(p) \) and \( P(q) \) respectively. The purpose of GHE is trying to find out the \( T \) which makes the \( P(q) \) uniform, while \( T \) needs to satisfy the condition in which \( T(p) \) is a monotonically increasing function, \( 0 \leq T(p) \leq 1 \) for \( 0 \leq p \leq 1 \).

To find the proper \( T \) value, first, we need to find the number of pixels in different gray scales, \( n_k \), in the original image with \( k=0, 1, \ldots, L-1 \), and \( L \) is the total number of the gray scales.

Secondly, the histogram of original image is calculated as follows,

\[ P(p_k) = \frac{n_k}{n} \]  

(4.9)

Then, the cumulative distribution function is calculated as follows, which \( k=0, 1, \ldots, L-1 \),

\[ q_k = T(p_k) = \sum_{j=0}^{k} n_j / n = \sum_{j=0}^{k} P(p_j) \]  

(4.10)

Finally, the output gray value \( q_k \), as shown in equation (4.11) is calculated. Note ‘\( Int\)’ is the rounding operator.

\[ q_k = Int[(L - 1)q_k + 0.4] \]  

(4.11)

The proposed global histogram equalization method is very effective not only in enhancing the entire image but also in enhancing the textural details. It also makes the change of the order of gray levels of the original image completely controllable. Therefore, it can enhance the contrast of the images more effectively.

In our experiments, the original foggy image we used has low brightness, low contrast, low visibility and unclear boundaries between scenes. After processing with
GHE, the gray value is uniformly distributed in the [0-255] interval. Moreover, the brightness, contrast and object's outline are improved. However, this algorithm also suffers from some disadvantages, such as the loss of some detail texture information. Figure 4-3 shows the result of GHE processing.

![Figure 4-3 (a) Original image (b) Result of GHE](image)

Due to the results of experiments, we know that the GHE algorithm is simple and can effectively improve the image resolution. Although this algorithm is effective for gray intensity image processing but it hardly deals with the color information due to the color distortion. The essence of histogram equalization is expanding the quantization interval rather than increasing the number of quantity. After histogram equalization processing, the original image's pixels which have different gray values may become the same grayscale, which easily leads pseudo profile appearance.
Chapter 5

Low Observable Targets Detection and Tracking Using Genetic Algorithm

In this chapter, we present an effective and fast algorithm to detect and track low observable targets in a digital image sequence. First, an encoding scheme along with genetic operation is designed to track the targets. To avoid missing any tracks, the individual preservation method is introduced to maintain more promising candidate tracks. Target trajectories are then confirmed by a multi-stage hypothesis testing scheme. The simulation results show that the proposed scheme can efficiently detect and track small targets with an SNR value under 2db.

5.1 Encoding Scheme and Design of Genetic Operators

Genetic algorithm is an optimized searching algorithm which can effectively be used to detect and track targets in image sequences. Generally speaking, the time of convergence and computational cost of a genetic algorithm highly depends on the encoding scheme and genetic operations. The following encoding scheme and genetic operation are designed to track the low SNR targets.

5.1.1 The structure and storage of candidate tracks

The targets used in our method are too small to be detected in one frame, therefore, motion characteristics and searching along the track are used to locate the
targets. Since the location, speed and direction of targets are unknown, the candidate path need to be calculated based on the range of the speed of the target. Assuming linear motion with a constant velocity and the velocity of the object being equal to or less than one pixel per frame, the number of children-nodes generated from a random point \( k \) (initial point) is equal to 9. Therefore, a track tree for each possible point beginning at \( k \) will be constructed. In addition, in order to save time, codes of each target and the coordinate offsets of the starting point will be saved. Candidate tracks and initial track will be built and searched from candidate track trees.

### 5.1.2 Encoding

![Figure 5-1 A structure of chromosome](image)

In the genetic algorithm (GA), a chromosome of an individual is a solution candidate to be optimized. In other words, chromosomes specify parameters which represent coordinates and scaling of an object to be explored on the target video frame. Figure 5-1 represents an encoding of a chromosome in which \( t_x, t_y \) are coordinates of the initial point and \( m \) is the location of the next point on the track. In this case, the chromosome is expressed as a binary string for simplicity. The parameters are coded in a total of 24 bits with the following range values.

\[
0 \leq t_x \leq \text{width of the video frame}
\]

\[
0 \leq t_y \leq \text{height of the video frame}
\]

\[
0 \leq m \leq 8
\]
These ranges are decided by a presumed motion in the general frame rate of about 30 frames per second. Note, m represents the location of the next point one the track in the range of [0, 8] which indicates that the speed of the target is relatively slow in video frames.

5.1.3 Crossover

The purpose of this step is to implement crossover operations between two chromosomes. Using coordinate crossover, every pixels in the image could be an initial point at least once.

5.1.4 Mutation

The mutation operation includes initial point mutation and children-node mutation to make sure every point will be searched. In addition, mutation of m assures that all adjacent pixels of the initial points will be considered as a children point.

5.1.5 Fitness function

It is assumed that the gray value of the target is higher than the average gray value of the noise. Let \( y_i \) represents the gray value of a random point in \( i \)-th frame. Note that the point represented by \( y_i \) could be an actual target or simply a noise point. Assume \( H_1 \) represents the hypothesis for the presence of the target and \( H_0 \) represents the absence of the target, as shown below,

\[
H_0: y_i = m_0 + n \\
H_1: y_i = m_1 + n
\]  

(5.1)

In equation (5.1), \( m_0 \) is the average gray value of the background, \( m_1 \) is equal to the gray value of the target plus the average gray value of the background; and \( n \) is the Gaussian noise with mean value \( 0 \) and variance \( \sigma^2 \). Equation (5.1) can be rewritten as,
\[
H_0: y_i \sim N(m_0, \sigma) \\
H_1: y_i \sim N(m_1, \sigma)
\] (5.2)

In equation (5.2), \(N\) represents the Gaussian probability density function. The candidate tracks which start from one point in the first frame could consist of target and noise points. If this candidate track consists of all target points, the sum of \(M\) points in this track can be expressed as,

\[
\sum_{i=1}^{M} y_i \sim N(M \times m_1, \sqrt{M} \sigma)
\] (5.3)

The fitness function \(f\) is defined as,

\[
f = \sum y_i
\] (5.4)

It is obvious that as the value of \(M\) increases, the SNR value raises. This makes finding the actual track easier, but requires more computational power. On the contrary, when \(M\) decreases, the SNR value decreases. Therefore, the probability of finding actual track decreases. In this case, the genetic algorithm might get a false track partly consists of noise. Therefore, qualified individual preservation is proposed in order to avoid tracking the noise.

5.1.6 Qualified individual preservation

Choose an individual chromosome whose fitness function value is the highest in the population and higher than a threshold \(\gamma\), move it into the initial track. Fitness function, \(f\), should satisfy equation (5.5):

\[
(\sum_{i=1}^{j} y_i > \gamma) \cap (\sum_{i=1}^{j} y_i > j \times m_1), \quad \gamma = \sqrt{j \sigma \Phi^{-1}(\alpha_1) + j \times m_0}
\] (5.5)

Note, \(\alpha\) is a given low false alarm rate, \(\Phi\) is the standard normal distribution. When the cycle-index ends, \(\alpha_i\) will be updated to a higher value of \(\alpha_2\) from which a new threshold \(\gamma\) is calculated.
Qualified individual preservations ensure that the best individual chromosome in one generation will not be split by mutation and crossover operations.

5.1.7 Selection operation

From the algorithm discussed above, we are able to find the best individual chromosome. However, the best individual chromosome is not always the target. Comparing with the number of target tracks, the number of candidate tracks is considerably high. Therefore, some impossible tracks are removed by changing the coordinates to make sure that the fitness function is equal to or greater than zero. Using the best individual and qualified individual preservation, the converging time will be reduced and the track with noise will be mostly avoided.

When the SNR is very low, the initial tracks searched by the genetic algorithm contain many tracks which are a combination of noise and target. Therefore, in order to further avoid the noise in the tracking process a multi-stage hypothesis testing scheme is used.

5.2 Multi-Stage Hypothesis Testing

Multi-stage hypothesis testing algorithm is one of the methods which is used to perform tracking first followed by testing. Assume the image contains independent Gaussian white noise and there is one or more tree data structure can be used to represent the candidate tracks. We use the multi-stage hypothesis testing to remove and modify trees which fail to pass the test in each layer of the tree data structure. Therefore, multi-stage hypothesis testing is one of the effective and efficient filters. All initial tracks are examined with multi-stage hypothesis testing to find and confirm a true track.
The basic idea is to start from every potential point and organize the candidate tracks into a tree data structure. Assuming the target moves at most one pixel per frame, therefore, the moving range is a 3 x 3 neighborhood in the next frame. The number of candidate tracks grows exponentially as the number of frames increases leading to an enormous amount of computations. To decrease the computations, a multi-stage hypothesis testing algorithm is used to test every candidate tree. A typical tree structure representing a set of candidate tracks is depicted in Figure 5-2.

![Diagram of a tree data structure of candidate tracks](image)

Figure 5-2 Tree data structure of candidate tracks trajectory

Figure 5-2, shows five candidate tracks starting from node 1 in the first frame. These candidate tracks are 1-2-3-6, 1-2-4-7, 1-2-4-8, 1-2 5 9, 1-2-5-10, in which the real track is among them. The underlying idea of constructing the candidate tracks in the
adjacent frames is either targets do not move or move one pixel per frame. We use the following rules to remove unnecessary branches.

Rules: along the tracks, accumulate the gray value of pixels, and compare it with two thresholds $T_1, T_2$, where $T_1$ is less than $T_2$. The ones which are higher than high threshold $T_2$ are accepted as targets tracks; the ones which are lower than the low threshold $T_1$ are regarded as false tracks. Those which are between these two threshold values are kept and their merit will be evaluated in the following frames until the last frame.

There are two cases to be considered. In case 1, all points on the track are the target, i.e., the sum of gray values on the track is greater than $T_2$, where,

$$T_2 = \sum_{i=1}^{j} y_i, \text{ where } y_i \sim N(i \times m_1, i \times \sigma^2) \quad (5.6)$$

However, in case 2, the points on the track are to be noise points, i.e., the sum of gray values is lower than $T_1$, where,

$$T_1 = \sum_{i=1}^{j} y_i, \text{ where } y_i \sim N(i \times m_0, i \times \sigma^2) \quad (5.7)$$

Multi-stage hypothesis could improve the accuracy, and avoid false alarm rate caused by the small difference between $m_0$ and $m_1$. Assume $\alpha$ is the false alarm rate and $\beta$ is the miss detection rate, i.e.,

$$P(H_1|H_0) = \alpha$$
$$P(H_0|H_1) = \beta \quad (5.8)$$

The multi-stage hypothesis testing follows the Neyman-Pearson rule given below,

$$\begin{align*}
\sum_{k=1}^{i} y_k &\geq a_i & \text{accept } H_1 \text{ hypothesis} \\
\sum_{k=1}^{i} y_k &\leq b_i & \text{accept } H_0 \text{ hypothesis} \\
b_i &< \sum_{k=1}^{i} y_k < a_i & \text{add a new sample, continue testing}
\end{align*} \quad (5.9)$$
In the above equation, \( a_i = i \frac{m_1 + m_0}{2} + \mu_1 \frac{\sigma^2}{m_1 - m_0}, \) \( b_i = i \frac{m_1 + m_0}{2} + \mu_0 \frac{\sigma^2}{m_1 - m_0}, \) where

\[ \mu_0 = \ln \frac{\beta}{1 - \alpha}, \mu_1 = \ln \frac{1 - \beta}{\alpha}. \]

\( \mu_0 \) and \( \mu_1 \) are determined by a given false alarm rate and missing detection rate.

When \( i \) is reached \( N \) (number of image frames in the sequence), a decision is made whether to accept \( H_0 \) or \( H_1 \) in the following way. Note, \( N \) should be higher than \( N_i \), which is average number of frames required, \( N_i = [\Phi^{-1}(\alpha) + \Phi^{-1}(\beta)]^2 \sigma^2 / (m_1 - m_0)^2 \).

\[
\begin{align*}
\sum_{j=1}^{N} y_j & \geq \tau, \text{ accept } H_1 \text{ hypothesis} \\
\sum_{j=1}^{N} y_j & < \tau, \text{ accept } H_0 \text{ hypothesis}
\end{align*}
\]

where threshold \( \tau \) can be determined as,

\[
\tau = \sqrt{N_1 [m_1 \Phi^{-1}(\alpha) + m_0 \Phi^{-1}(\beta)] \sigma^2 / (m_0 - m_1)}
\]

Note, \( \Phi \) is the standard normal distribution.

5.3 Algorithm Description

The low SNR target detection and tracking method as was discussed can be summarized in the following steps:

1. Use Total Variation technique to denoise the image.
2. Randomly construct a candidate track tree, the root can be any point in the input image.
3. Set the initial population; initializes the starting track area; randomly select starting points and children-nodes of the starting points.
4. Based on the candidate track trees, build the candidate tracks whose length is \( N \), then use genetic algorithm to search in the initial tracks.
(5) Test initial tracks with the multi-stage hypothesis testing, move the confirmed tracks to the target track area.

The flowchart of this algorithm is shown in the following Figure 5-3,

![Flowchart of the Algorithm](image)

Figure 5-3 Flowchart of the Algorithm
Chapter 6

Simulation Results

This chapter describes the implementation and simulation results for the image denoising, enhancement and target detection and tracking. In the image denoising experiment, I used MATLAB program to simulate the total variation algorithm and its effectiveness in noise removal. Then, the simulation results comparing the image enhancement proposed in this thesis and other common algorithms are presented. The last part of this chapter is the simulation results for the targets detection and tracking.

6.1 Simulation results for the Image Denoising

Figure 6-1 is the comparison of spatial distribution characteristic before and after denoising. Figure 6-1 (a) is the image before denoising and (b) is after denoising. Our 3D result is an easy way to see the point representing the target. Although the target is covered by the noise, we can easily distinguish it by naked eyes. This step greatly facilitates the next step dealing with the detection and tracking.
6.2 Simulation results for the Image Enhancement

The proposed algorithm has been implemented in MATLAB, and its performance and simulation results are presented in this section. Figure 6-2 (a) is the original foggy image and Figure 6-2 (b) represent the enhanced image after the application of the proposed method. For the purposes of comparison, the results of the Retinex enhancement applied to luminance and the GHE process applied to chrominance individually are shown in Figure 6-2 (c) and (d), respectively.
The simulation results show that the proposed algorithm having the advantages of both the Retinex and GHE methods outperform the other two algorithms when applied individually. The Retinex algorithm is effective at enhancing the luminance factor but suffers from weak contrast in color components. Meanwhile, the GHE algorithm can drastically increase the contrast of the foggy image at the expense of losing some luminance information.

Therefore, the proposed algorithm uses PCA to decompose the original image to two separate channels: luminance and chrominance. Then, we apply the Retinex
algorithm to the luminance channel to enhance the light factor and apply the GHE algorithm to the chrominance channel to enhance the color factor. Finally, the inverse PCA is applied to combine the two channels together to generate the final image.

### 6.3 Simulation results for the Target Detecting and Tracking

The target detection and tracking algorithms are implemented in MATLAB, and the results of its performance are presented in this section. The input video in this simulation is taken from an in-car camera, which is 30 FPS, and the length of it is 10 seconds. We use 20 frames, $32 \times 32$ images in the simulation. The noise in the image is independent white noise whose mean value is equal to 0 and its variance equals to 12. The gray values of the targets in various image frames are about 2. Therefore, the targets are basically independent white noise whose mean value is equal to 2 with variance 12.

There are a number of parameters in Genetic Algorithm that should be properly selected in the training stage. Table 6.1 lists the values of the parameters that have been selected in our implementation. These parameters are population size, cycle-index, and the probability of performing crossover and mutation operation.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Parameter values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>Cycle-index</td>
<td>30</td>
</tr>
<tr>
<td>Probability of performing crossover (PC)</td>
<td>0.3</td>
</tr>
<tr>
<td>Probability of mutation (PM)</td>
<td>0.3</td>
</tr>
</tbody>
</table>

The number of candidate tracks from a point in each frame is $32 \times 32 \times 9 = 9216$. 

41
The tracking parameter values of the simulation are set as shown in table 6.2.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Symbols</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>False alarm rate</td>
<td>$\alpha_1$</td>
<td>0.03</td>
</tr>
<tr>
<td>False alarm rate</td>
<td>$\alpha_2$</td>
<td>0.06</td>
</tr>
<tr>
<td>Missing detection rate</td>
<td>$\beta$</td>
<td>0.08</td>
</tr>
<tr>
<td>Length of candidate tracks</td>
<td>$N_1$</td>
<td>8</td>
</tr>
<tr>
<td>Threshold of qualified individual reservation</td>
<td>$\gamma$</td>
<td>25</td>
</tr>
</tbody>
</table>

The initial tracks are shown in Figure 6-3, and the result after multi-stage hypothesis test is shown in Figure 6-4.

In both Figure 6-3 and 6-4, X-axis represents the horizontal coordinate of the pixels and the Y-axis represents the vertical coordinate of the pixels in the image.
Due to the lack of information, the use of the genetic algorithm for target detection using single frame will also introduce a lot of false targets as shown in the following figures,

Therefore, to increase the accuracy of the algorithm, the targets are detected based on the information among a sequence of images. In this way, the false targets are removed and only the real targets are detected.

Figure 6-7 shows that the use of information from an image sequence can detect the real targets avoiding the detection of false targets by using a single frame. This is
evidenced by comparing Figure 6-6 and 6-7 in which the presence of seven targets in fourth frame is reduced to only one real target.
Chapter 7

Conclusion and Future Work

There have been significant advances in the area of image enhancement, denoising, target detection and tracking. In this thesis a unique and effective method for foggy image enhancement has been presented. The design of an enhancement algorithm should serve a particular application area and is the crucial factor in obtaining a better simulation results. In the area of foggy image enhancement, both luminance and chrominance components play an important role. Therefore, the proposed method chooses the Retinex and GHE algorithms to deal with the luminance and chrominance components. The separation of these two components is achieved by using a PCA transformation. It should be mentioned that both Retinex and GHE algorithms have their own advantages and weaknesses. Consequently, using the PCA transformation to decompose the foggy image and dealing with the two components separately is very important. The simulation results show that the proposed method can effectively increase the definition of a foggy image.

For the object detection and tracking system, this thesis presents a procedure for detection and tracking of low signal-to-noise ratio target in sequence of images. The proposed scheme is based on a genetic algorithm involving encoding and corresponding
genetic operators followed by qualified individual reservation to detect and track objects. The simulation results show that the algorithm is effective to detect and track targets even at low SNR value of 2db. Comparing with the traditional algorithm, the new algorithm proposed in this thesis involves less computations and relatively simple to implement. In addition, it also avoids false tracks detection when the target signal-to-noise ratio is low. Based on genetic algorithm and hypothesis testing algorithm, we use the genetic algorithm to search possible tracks, test certain length assumption track and prevent missing detection.

Future research for image enhancement will investigate the use of other image transformation algorithms and the combination of other imaging techniques, such as the YCbCr transform, the Adaptive Histogram Equalization (AHE), and Wavelet transform. For the detection and tracking part, further research is needed to obtain missing tracks which happens when the size of population and/or the cycle-index of the number of iterations in the genetic algorithm are small.
References


Appendix A

Source Code for Total Variation Image Denoising

tic;
close all;
clear all;
cle;

I=imread('1.tif'); % load image
I=double(I(20:120,10:105)); % cut a piece, convert to double

% params
iter=80;
% dt=0.2; eps=1;

%%%% Add noise
std_n=20; % Gaussian noise standard deviation
In = randn(size(I))*std_n; % White Gaussian noise
I0 = I + In; % noisy input image
% show original and noisy images

close all

figure(1); imshow(uint8(I)); title('Original')

figure(2); imshow(uint8(I0)); title('Noisy image')

% denoise image by using tv for some iterations

J=tv(I0,iter);

figure(3); imshow(uint8(J)); title('Denoised image')
Appendix B

Source Code for Image Transform using PCA

close all;
clear all;
clc

function y = pca(mixedsig)
if nargin == 0
    error('You must supply the mixed data as input argument.');
end
if length(size(mixedsig))>2
    error('Input data can not have more than two dimensions. ');
end
if any(any(isnan(mixedsig)))
    error('Input data contains NaN"s.');
end

#———remove mean———

%———remove mean———
meanValue = mean(mixedsig');
mixedsig = mixedsig - meanValue * ones(1,size(meanValue,2));
[Dim,NumofSampl] = size(mixedsig);
oldDimension = Dim;
fprintf('Number of signals: %d
',Dim);
fprintf('Number of samples: %d
',NumofSampl);
fprintf('Calculate PCA...');
firstEig = 1;
lastEig = Dim;
covarianceMatrix = cov(mixedsig',1);  % covariance matrix
[E,D] = eig(covarianceMatrix);
% eigenvalue and eigenvalue vector of covariance

%The number of eigenvalue of covariance which is higher than the threshold
rankTolerance = 1e-5;
maxLastEig = sum(diag(D)) > rankTolerance;
lastEig = maxLastEig;

% descending sort eigenvalue
eigenvalues = flipud(sort(diag(D)));

% remove lower eigenvalue
if lastEig < oldDimension
lowerLimitValue = (eigenvalues(lastEig) + eigenvalues(lastEig + 1))/2;
else
    lowerLimitValue = eigenvalues(oldDimension) - 1;
end
lowerColumns = diag(D) > lowerLimitValue;

% remove highest eigenvalue (optional)
if firstEig > 1
    higherLimitValue = (eigenvalues(firstEig - 1) + eigenvalues(firstEig))/2;
else
    higherLimitValue = eigenvalues(1) + 1;
end
higherColumns = diag(D) < higherLimitValue;

% choose eigenvalue
selectedColumns = lowerColumns & higherColumns;

% show
fprintf('Selected[ %d ] dimensions.
',sum(selectedColumns));
fprintf('Smallest remaining (non-zero) eigenvalue[ %g ]
',eigenvalues(lastEig));
fprintf('Largest remaining (non-zero) eigenvalue[ %g ]
',eigenvalues(firstEig));
fprintf('Sum of removed eigenvalue[ %g ]
',sum(diag(D).*(~selectedColumns)));
% choose eigenvalue
E = selcol(E,selectedColumns);
D = selcol(selcol(D,selectedColumns)',selectedColumns);

% whiten matrix
whiteningMatrix = inv(sqrt(D)) * E';
dewhitenMatrix = E * sqrt(D);

% main component
y = whiteningMatrix * mixedsig;

% separate image into two components
function newMatrix = selcol(oldMatrix,maskVector)
if size(maskVector,1)~ = size(oldMatrix,2)
    error('The mask vector and matrix are of incompatible size.);
end
numTaken = 0;
for i = 1:size(maskVector,1)
    if maskVector(i,1) == 1
        takingMask(1,numTaken + 1) == i;
        numTaken = numTaken + 1;
    end
end
newMatrix = oldMatrix(:,takingMask);
Appendix C

Source Code for Luminance Component Enhancement Using MSR

clc

clear

I=imread('fig5.jpg');
fr=I(:, :, 1);
fg=f(:, :, 2);
fb=I(:, :, 3);     %RGB
mr=mat2gray(im2double(fr));
mg=mat2gray(im2double(fg));
mb=mat2gray(im2double(fb));     %Data type normalization

alf1=1458; %Define standard deviation:  alf=a^2/2  a=54
n=161;%Define the template size
n1=floor((n+1)/2);%Calculate center

for i=1:n
    for j=1:n
        b(i,j) =exp(-(i-n1)^2+(j-n1)^2)/(4*alf1)/(pi*alf1); %Gaussian function
    end
end

nr1 = imfilter(mr,b,'conv', 'replicate');

ng1 = imfilter(mg,b,'conv', 'replicate');

nb1 = imfilter(mb,b,'conv', 'replicate');%Convolution filtering

ur1=log(nr1);

ug1=log(ng1);

ub1=log(nb1);

tr1=log(mr);

tg1=log(mg);

tb1=log(mb);

yr1=(tr1-ur1)/3;

yg1=(tg1-ug1)/3;

yb1=(tb1-ub1)/3;

alf2=53.38; %Define standard deviation: alf=a^2/2 a=10.3325

x=31;%Define the template size

x1=floor((n+1)/2);%Calculate center

for i=1:n

    for j=1:n

        a(i,j) =exp(-((i-n1)^2+(j-n1)^2)/(4*alf2))/(6*pi*alf2); %Gaussian function

    end

end

nr2 = imfilter(mr,a,'conv', 'replicate');

ng2 = imfilter(mg,a,'conv', 'replicate');
nb2 = imfilter(mb,a,'conv', 'replicate');%Convolution filtering
ur2=log(nr2);
ug2=log(ng2);
ub2=log(nb2);
tr2=log(mr);
tg2=log(mg);
tb2=log(mb);
yr2=(tr2-ur2)/3;
yg2=(tg2-ug2)/3;
yb2=(tb2-ub2)/3;
alf3=13944.5; %Define standard deviation: alf=a^2/2 a=167
l=501;%Define the template size
l1=floor((n+1)/2);%Calculate center
for i=1:n
    for j=1:n
        e(i,j) =exp(-((i-n1)^2+(j-n1)^2)/(4*alf3))/(4*pi*alf3); %Gaussian function
    end
end
nr3 = imfilter(mr,e,'conv', 'replicate');
ng3 = imfilter(mg,e,'conv', 'replicate');
mb3 = imfilter(mb,e,'conv', 'replicate');%Convolution filtering
ur3=log(nr3);
ug3=log(ng3);
ub3 = \log(nb3);
tr3 = \log(mr);
tg3 = \log(mg);
tb3 = \log(mb);
 yr3 = (tr3 - ur3) / 3;
yg3 = (tg3 - ug3) / 3;
yb3 = (tb3 - ub3) / 3;
 dr = yr1 + yr2 + yr3;
dg = yg1 + yg2 + yg3;
 db = yb1 + yb2 + yb3;

cr = im2uint8(dr);
cg = im2uint8(dg);
cb = im2uint8(db);
z = cat(3, cr, cg, cb);
figure, imshow(z)
Appendix D

Source Code for Chrominance Component Enhancement Using GHE

clear all;
close all;
clear all

%%% load image
PS=imread('1.jpg');
imshow(PS)
title('input image')
imwrite(rgb2gray(PS),'PicSampleGray.bmp');
PS=rgb2gray(PS);

%%% draw histogram
[m,n]=size(PS);
GP=zeros(1,256);
for k=0:255
GP(k+1)=length(find(PS==k))/(m*n);
end
figure,bar(0:255,GP,'g')
title('original histogram')
xlabel('gray value')
ylabel('probability')

%%%% histogram equalization
S1=zeros(1,256);
for i=1:256
    for j=1:i
        S1(i)=GP(j)+S1(i);
    end
end

S2=round((S1*256)+0.5);
for i=1:256
    GPeq(i)=sum(GP(find(S2==i)));
end
figure,bar(0:255,GPeq,'b')
title('equalized histogram')
xlabel('gray value')
ylabel('probability')

%% GHE
PA=PS;
for i=0:255
    PA(find(PS==i))=S2(i+1);
end

figure, imshow(PA) % show the enhanced image by GHE
title('GHE enhanced image')
imwrite(PA,'PicEqual.bmp');
Appendix E

Source Code for Target Detection

RGB = imread('111.jpg');
I=rgb2gray(RGB); %Convert to Grey level image
[x,y]=size(I);
BW=edge(I);
figure;imshow(I);title('Original Image')
figure;imshow(BW);title('Edge Detection Image')

rho_max=floor(sqrt(x^2+y^2))+1;

accarray=zeros(rho_max,180);

Theta=[0:pi/180:pi];

for n=1:x,
    for m=1:y
        if BW(n,m)==1
for k=1:180
    rho=(m*cos(Theta(k)))+(n*sin(Theta(k)));
    rho_int=round(rho/2+rho_max/2);
    accarray(rho_int,k)=accarray(rho_int,k)+1;
end
end
end
end

%figure;colormap gray;
%imagesc(accarray);title('hough±ää»»ºµÄÍ¼')
%xlabel('theta'), ylabel('rho');

%accarray=uint8(accarray);
%figure;imshow(accarray);title('hough±ää»»ºµÄÍ¼')
%xlabel('theta'), ylabel('rho');
%axis on, axis normal, hold on;

K=1;
for rho_n=1:rho_max
    for theta_m=1:180
        if accarray(rho_n,theta_m)>=10
            case_accarray_n(K)=rho_n;
        end
    end
end

case_array\_m(K) = theta\_m;
K = K + 1;
end
end
end

I\_out = zeros(x, y);
I\_jiao\_class = zeros(x, y);
for n = 1:x,
    for m = 1:y
        if BW(n, m) == 1
            for k = 1:180
                rho = (m * cos(Theta(k))) + (n * sin(Theta(k)));
                rho\_int = round(rho/2 + rho\_max/2);
                for a = 1:K-1
                    if rho\_int == case\_array\_n(a) & k == case\_array\_m(a)
                        I\_out(n, m) = BW(n, m);
                        I\_jiao\_class(n, m) = k;
                    end
                end
            end
        end
    end
end
end
end
end

figure;imshow(I_out);title('ÀûÓÑÔ±ëåª³òåçë³áä³ëë±ã»»ÎáÈëµÄÉ¾Ìí');

[m,n]=size(I_out);
for i=1:ceil(2*m/7)
  for j=1:n
    I_out(i,j)=0;
  end
end

figure
imshow(I_out);

%=============hough transform toolbox================%

% [H,T,R] = hough(BW,'RhoResolution',0.5,'ThetaResolution',0.5);
%
figure;imshow(H,'XData',T,'YData',R,'InitialMagnification','fit');title('hough±ë»»½ØÔó')
% xlabel('theta'), ylabel('rho');
% axis on, axis normal, hold on;
Appendix F

Source Code for Target Tracking

close all;
clear all;
clc

h = [40 30]; %Define the object size
I = imread('frame0001.jpg'); R = I;
imshow(I); hold on
pt = ginput(1); y(1) = round(pt(2));
y(2) = round(pt(1));

r = 10;

I(y(1), y(2) - r:y(2) - 3,:) = 255; I(y(1), y(2) + 3:y(2) + r,:) = 255; %Left side of the rectangle
I(y(1) - r:y(1) - 3,y(2),:) = 255; I(y(1) + 3:y(1) + r,y(2),:) = 255; %Right side of the rectangle
I(y(1), y(2),:) = 255;
I(y(1), y(2) - round(h(1)/2):y(2) + round(h(1)/2),:) = 255;
I(y(1) - round(h(1)/2),y(2) - round(h(2)/2):y(2) + round(h(2)/2),:) = 255;
I(y(1) - round(h(1)/2), y(2) - round(h(2)/2): y(2) + round(h(2)/2),:) = 255;
I(y(1) - round(h(1)/2): y(1) + round(h(1)/2), y(2) - round(h(2)/2),:) = 255;
I(y(1)-round(h(1)/2):y(1)+round(h(1)/2),y(2)+round(h(2)/2):)=255;

%Draw rectangle box

mov = avifile('Object Tracking.avi','fps',15,'quality',100);
F=im2frame(I);
mov = addframe(mov,F); %Add first frame
[~ ~ pic]=TemplateTrans(R,y(1),y(2),h);
x=y(2);xb=x;
y=y(1);yb=y;

%Define Search area

py=0;ty=12;
px=0;tx=12;
output(1)=100;
tic
num=1;
for Frame=1:120
    filename = sprintf('%3.3d.jpg', Frame);
    ImageName=strcat('frame0',filename);
    I =imread(ImageName);
    R=rgb2gray(I);
R=im2double(R);

[Temp detect]=Transform(R,y,x,h);

TeF=Temp.*Aim;

Out=abs(ifft2(TeF))

op=max(max(Out(1:64,1:64)));

output(Frame)=op;

[y1 x1]=find(Out==op);

y1=y1(1)-2;

x1=x1(1)-2;

if y1>32 & x1>32
  y1=y1-64;
  x1=x1-64;
elseif y1>32 & x1<32
  y1=y1-64;
elseif y1<32 & x1>32
  x1=x1-64;
end

y=y+y1;

x=x+x1;

num=num+1;

if num>3
  if Frame>4
    if op<output(Frame-2)
if op>1/2*output(Frame-2)

    [Gy,Gx]=modify(Aimer);

    y=y+Gy;
    x=x+Gx;
    Aimer=UpImage(R,y,x,h);

    [H1,L1]=size(Aimer);
    R1=round((64-H1)/2);
    R2=round((64-L1)/2);
    Amp=zeros(64,64);
    Amp(R1:R1+H1-1,R2:R2+L1-1)=Aimer;
    Aim=conj(fft2(Amp));
    num=0;
    end
  end
end
end

%%%Redraw the rectangle box

I(y+round(h(1)/2),x-round(h(2)/2):x+round(h(2)/2),:)=255;
I(y-round(h(1)/2),x-round(h(2)/2):x+round(h(2)/2),:)=255;
I(y-round(h(1)/2):y+round(h(1)/2),x-round(h(2)/2),:)=255;
I(y-round(h(1)/2):y+round(h(1)/2),x+round(h(2)/2),:)=255;
I(y,x-r:x-3,:) = 255; I(y,x+3:x+r,:) = 255;
I(y-r:y-3,x,:) = 255; I(y+3:y+r,x,:) = 255;
I(y,x,:) = 255;

xtp(Frame) = x;
ytp(Frame) = y;

% I = bitmapplot(ytp, xtp, I, struct('LineWidth', 1, 'Color', [1 0 0 1]));
imshow(I), drawnow
F = im2frame(I);

mov = addframe(mov, F);
Frame
end

\( t = \text{Frame/toc} \)
figure, plot(xtp, ytp, '*'), title('tracking path'), grid on

clear all