Multi-Mask Based Stabilization of Turbulence Degraded Videos Containing Moving Objects

by

Bhabesh Ray

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Electrical and Electronic Engineering
in the Department of Electrical and Electronic Engineering

Khulna University of Engineering & Technology
Khulna 9203, Bangladesh
November 2019

Declaration

This is to certify that the thesis work entitled "Multi-Mask Based Stabilization of Turbulence Degraded Videos Containing Moving Objects" has been carried out by Bhabesh Ray in the Department of Electrical and Electronic Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh. The above thesis work or any part of this work has not been submitted anywhere for the award of any degree or diploma.

Signature of Supervisor
(Dr. Kalyan Kumar Halder)

Bhabesh
Signature of Candidate
(Bhabesh Ray)

Approval

This is to certify that the thesis work submitted by Bhabesh Ray entitled "Multi-Mask Based Stabilization of Turbulence Degraded Videos Containing Moving Objects" has been approved by the board of examiners for the partial fulfillment of the requirements for the degree of Master of Science in Electrical and Electronic Engineering in the Department of Electrical and Electronic Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh in September 2019.

BOARD OF EXAMINERS

1. DHalder 26|11|19 ______________________ Chairman (Supervisor)
   Dr. Kalyan Kumar Halder
   Assistant Professor, Department of EEE
   Khulna University of Engineering & Technology

2. Rafiqul ______________________ Member
   Dr. Md. Rafiqul Islam
   Professor & Head, Department of EEE
   Khulna University of Engineering & Technology

3. 26/11/2019 ______________________ Member
   Dr. Mohiuddin Ahmad
   Professor, Department of EEE
   Khulna University of Engineering & Technology

4. 26/11/2019 ______________________ Member
   Dr. Md. Salah Uddin Yusuf
   Professor, Department of EEE
   Khulna University of Engineering & Technology

5. ______________________ Member (External)
   Dr. Muhammad Shahin Uddin
   Professor, Department of ICT
   Mawlana Bhashani Science and Technology University

Dedication

To

My Beloved Parents

and

Honorable Supervisor

Acknowledgement

At the beginning, I glorify Almighty God, who is the creator and the sustainer of this world and belongs to the supreme power. He has given me energy and spirit to fulfill all the requirements for the degree of Master of Science in Electrical and Electronic Engineering including this thesis work.

It is my great pleasure to express my profound sense of gratitude to my supervisor Dr. Kalyan Kumar Halder, Assistant Professor, Department of EEE, Khulna University of Engineering & Technology (KUET), for his kind co-operation, constructive advice and guidance, constant encouragement and valuable suggestions, and all other supports throughout this work.

Finally, 1 would like to express my gratitude to my parents Suprovat Ray and Anita Ray for their sacrifice and providing mental support and valuable advice on every steps of my life.

The thesis would not be possible to complete without the help of all of them. Thank you all, wholeheartedly!

Khulna, November 2019 Bhabesh Ray

Abstract

Stabilizing videos and detecting moving objects are important tasks in many computer vision applications though it becomes challenging because of the presence of atmospheric turbulence that causes random pixel shifting and blurring of the videos. Because of random change in magnitude and direction of hot air and winds present in the atmosphere, the refractive index of the medium changes nonuniformly. This causes deformation of pixels and makes the tracking system confused while detecting moving objects. This effect becomes more severe with the increase of imaging distance. This thesis proposes an improved method for correcting geometrical distortions of videos degraded by atmospheric turbulence while keeping moving objects unaltered. In this method, by taking the median of input frames, the background frame is estimated and three different techniques are used to generate three different masks. Then, by combining all three masks, a more accurate mask is generated. This refined mask is employed to properly detect the moving objects, and finally combining with the background a stabilized video output is obtained. Performance of this method is tested by applying it on different real-world datasets. A comparison with an existing method shows that the proposed method gives better moving objects detection and improved stabilization of distorted videos.

Contents

Page

List of Tables

List of Figures

List of Abbreviations

CCD	Charged coupled device
LRMD	Low-rank matrix decomposition
RGB	Red, Green, Blue
HSL	Hue, saturation, lightness
HSV	Hue, saturation, value
CMYK	Cyan, Magenta, Yellow, and key (Black)
HOT	Higher order terms
MEX	MATLAB Executable
MSE	Mean squared error
PSNR	Peak signal-to-noise ratio
SSIM	Structural similarity
ANN	Artificial neural network

Nomenclature

G_a, G_b, G_c	Three desired gray levels
G_{max}	Maximum value of gray level
k_1, k_2	Two different threshold values
N	Total number of pixels in an image
d_0 , d_1 , … , d_{n-1}	Quantization levels
d_k	One of the gray levels between 0 and 255
$h(d_k)$	Total number of pixels having same intensity level d_k
$f(x, y)$	Input image
$g(x, y)$	Thresholded image
T	Global threshold value
$I(x, y, t)$	Pixel intensity
Δx, Δy, Δt	Small movement of pixel
V_x, V_y	x and y component of the velocity
$\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, $\frac{\partial I}{\partial t}$	Derivatives of image pixel in corresponding directions
I_x, I_y, I_t	Derivatives of image pixel in corresponding directions
∇I	Image gradient
e_s	Measure of departure from smoothness
e_b	Error of optical flow for the rate of change of brightness
e	Total error
$D(x, y)$	Difference frame
$I(x, y)$	Input frame
$B(x, y)$	Background frame
T_{Il}	Low-level threshold
T_{Ih}	High-level threshold
g_h	Gain value
K_l, K_h	Low-level and high-level offset value
$\bar{\imath}$	Mean intensity of the difference frame
$M_I(x, y)$	Generated mask using intensity
P_i	Probability distribution of gray-level histogram

n_i	Each pixel
C_1, C_2	Pixels of two groups as foreground and background
P_1, P_2	Probability of class occurrence
M_1, M_2	Class means levels
M_g	Class means average
V_1, V_2	Class variances
T_{hist}	Threshold using histogram
$M_O(x,y)$	Generated mask using histogram
S_x	x-directional pixel shiftmap
S_y	y-directional pixel shiftmap
T_s	Threshold using optical flow
g_s	Gain
σ_s	Standard deviation of shiftmap
K_s	Offset value
$M_S(x,y)$	Generated mask using optical flow
$M_F(x,y)$	Final mask

CHAPTER I

Introduction

1.1 Overview

Video obtained through various video capturing devices are known to be degraded representations of the real videos due to imperfections in the imaging and capturing process. Such degradations can occur as a result of lens distortion, defocus, relative motion between the subject and the camera during the photographic exposure and due to atmospheric turbulence. Since the images captured or the recorded video through turbulence medium are usually degraded, they show some limitations when they are used for display and analysis. The problem of recovering the original stable frames from a degraded observed one is usually known as the problem of image restoration and video stabilization.

Moving objects detection and video stabilization are important tasks in computer vision applications. But it is challenging because of the presence of atmospheric turbulence throughout the imaging path for which the refractive index of the medium vary randomly that causes non uniform pixel shifting and blurring and this effect becomes severe with the increase of imaging distance. Endoscopy or underwater photography where video is acquired through liquids, defines another class of applications that imagery might be subjected to turbulence degradations. In all these cases, videos are very unstable due to degradation in their resolution and geometry because it is captured by optical sensors in the presence of turbulence. Visual analysis of such wavering output for reliable detection and tracking of moving objects is practically impossible. This strongly motivates development of real-time methods for stabilization of turbulent videos.

Generation of an effective mask for moving object detection plays an important role in many computer vision applications such as object detection, object recognition, location tracking for surveillance. But this is a challenging task due to image distortion effect as a result of the turbulence in the imaging path. As the problem of recovering an input signal from a degraded output is important in science and technology, image restoration and video stabilization has been an active research area.

1.2 Video Stabilization

Removal of unwanted motion in video sequences is known as video stabilization. This unwanted motion, often called jitter, is induced by vibrations or an unsteady platform. This is true for any system using a camera, such as robotic systems or handheld cameras. Removal of this undesired motion results in a stabilized and smoother video. This smoother video is a more viewable and user-friendly version of the degraded video.

Recently, several consumer and professional digital cameras incorporate anti-shake mechanisms. Those are using one of the following stabilization principles: digital image stabilization, optical image stabilization, and mechanical image stabilization [1].

1.2.1 Digital image stabilization

Digital image stabilization, also known as electronic image stabilization is a post-processing technique that is used in many video capturing devices. It reduces the level of blur or distortion in videos that can result from unsteady or shaky movement. This technique shifts the electronic image from frame to frame of a video, enough to neutralize the motion. From the digital input frame, the motion of the camera is estimated and movement correction is performed by digital processing.

In digital image stabilization system, there are two ways to reduce the perceived motion. One is by digitally zooming in on the image and then “pan and scan” within the image to counter the movement created by the vibration. Because of zooming in on the image the resolution of the image is decreases. The other method uses an oversized Charged Coupled Device (CCD). Vibration detection is the key to the effectiveness of the system. To detect shaky video digital image stabilization systems use one of two ways. Either they detect the actual movement of the camera or they detect the movement within the image as recorded on the CCD. The changes between the fields in each image are analyzed in the second method of detection. The odd and even fields of the video frames are stored by the special feature of camera and then look for changes between them. Changing image parts in one field respect to all other fields, this feature indicates that the subject present in the field of view is moving but the background is stable. Changing entire image from one field to the next field most likely means that camera vibration is present and the camera must correct the image. For correcting the camera vibration, electronics in the camera detect the moving direction and shifts the active field so that it meshes with the memorized field.

A major disadvantage of this system is that if large moving object is present in the frame, it may interpret this motion as the camera vibration and then the camera will attempt to stabilize the frame which causes reduction in image resolution and blurring in the image. Motion sensors can also be used in the camera for detecting the vibration of the camera. Movement of the camera is sensed by this method but not the images thus, the movement of a field in the image cannot fool it [1].

1.2.2 Optical image stabilization

Before the image gets to the CCD the optical image stabilization system manipulates it. The light rays from the fields are bent relative to the optical axis when the lens moves, which results in an unsteady image because of the deflection of the light rays [1].

1.2.3 Mechanical image stabilization

Instead of stabilizing just the image, mechanical image stabilization method stabilizes the entire camera. “Gyros” are used for this type of stabilization. Gyros consist of a gyroscope with a battery pack and two perpendicular spinning wheels. Gyroscopes are motion sensors. For maintaining stability a signal is sent to the motors to move the wheels when the gyroscopes sense movement. The gyros are attached to the tripod socket of camera and it acts as an “invisible tripod”. The sensor-based image stabilization suffers from the fact that the motion sensor is normally non-collocated with the actual image sensor. Therefore, any mechanical distortions (vibration, structural deformations, misalignment. etc.) affect the final quality of the corrected image due to errors of measurement in image motion [1].

1.3 Atmospheric Turbulence

A well-known source of distortion that can degrade the quality of images and videos acquired by cameras viewing scenes from long distances is known as atmospheric turbulence. In astronomy, it is a common phenomenon. For example, when stars of outer space are viewed through telescopes those appear blurred because of the degradation of image quality due to the Earth's atmosphere. The fluctuation in the refractive index of the medium is the physical cause of the turbulence. Many factors involve in these fluctuations such as elevation, temperature gradients, wind velocity, etc. Temperature variation is usually the dominant factor. In practice, measuring these factors is difficult, and thus the parameters associated with a true physical model of atmospheric turbulence are typically unknown. Consequently the task of compensating for turbulence distortion can be viewed as a blind restoration problem. In addition to blurring, geometric distortion is also associated with turbulence, which arises because the turbulence is time-varying. This effect shows up in video when stationary objects appear to waver. The central interest of this thesis is to mitigate or eliminate geometric distortions of the captured video [2].

1.3.1 Geometrical distortion in long-distance imaging

Irregular motion of atmosphere or unstable flow of liquid or gas is known as atmospheric turbulence. The Earth's atmosphere, in general, is always in a turbulent state due to the continuously existing winds and thermal currents. It can be regarded as a dielectric medium which affects the radiation that passes through it. In other words, the atmosphere can be thought of as a large group of eddies of different dimensions, each eddy being a parcel of air. Over these eddies, the temperature, air pressure, and humidity deviates from their means [2].

Energy is introduced into the turbulence via the largest of these eddies while for eddies whose sizes are less than some value of the outer scale of turbulence, turbulence effects are considered to be isotropic. The major sources of energy are the wind shear, convective heating from the ground, and wind over objects. The region between the energy receiving eddies and the energy dissipating eddies is called the inertial sub-range, in which energy is assumed to be passed on without loss.

Thus, turbulence contributes to the inhomogeneity of the atmospheric path in which the index of refraction is a function of time and space. When light propagates through such a turbulent medium, its interaction with the eddies produces random variations in the amplitude and phase of the signal. These variations lead to several effects such as jitter, wander, blurring and speckle, which tend to degrade the performance of imaging systems. In addition, the air is full of atmospheric particles even when it seems to be completely clear. Examples of such particles are: invisible solid and semisolid bits of matter, water droplets, ice crystals, dust, and smoke. Some atmospheric particles are so large that they could cause deflection of a light ray from its transmission path. This effect is commonly known as “scattering”, which eventually, imposes attenuation in the optical path.

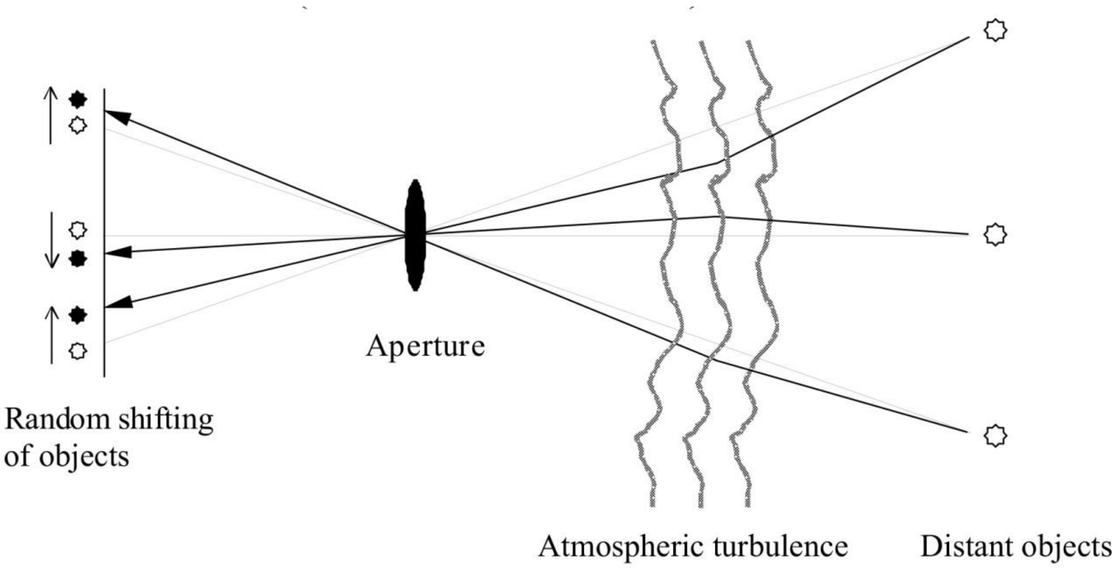


Figure 1.1: Illustration of turbulence effect in atmospheric imaging [3].

Figure 1.1 is a simple illustration of the geometric distortion problem in atmospheric imaging [3]. When light rays from distant objects (stars) pass through the atmosphere, the turbulence bends the rays by an angle from their true directions. At the receiving end, a camera captures these rays and generates the images of the remote objects. The apparent locations of these remote objects (stars in black) are recorded as randomly shifted from their true locations that could be otherwise observed in the absence of any turbulence [4].

1.3.2 The process of underwater imaging

When compared to similar problems in the air, underwater imaging poses significant challenges at extended ranges. Propagation of light in the water is similar to that through the atmosphere, but in underwater imaging higher light attenuation due to scattering and absorption is the main problem [4].

Because of the presence of small suspended biological organisms or dissolved organic matter scattering occurs. The direction of the light beam changes due to this scattering. In addition, the beam energy reduces because of the absorption process.

Due to salinity fluctuations and random temperature the water remains inhomogeneous even if it is clear, i.e., free of the small observable suspended particles. The density of water changes because of these variations which causes a variation in the refractive index. This variation of refractive index is the most dominating factor for underwater turbulence [4].

When light rays travel from one medium to another they bend together and the bending amount is determined by the refractive index of the two medium. Figure 1.2 illustrates a common example of underwater imaging problem, where the camera from a static viewpoint observes an underwater object through the water surface with its optical axis perpendicular to the object. For this case, the two media are air (refraction coefficient 1.00) where the camera is placed and the water (refraction coefficient 1.33), as the observed object is under the water.

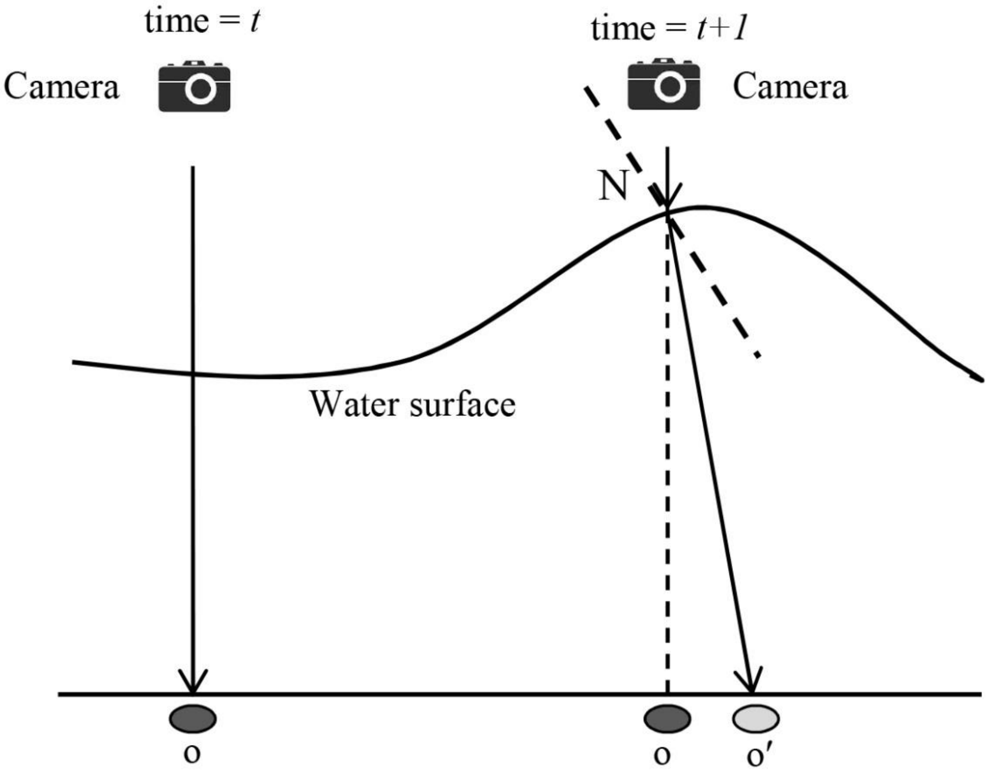


Figure 1.2: Illustration of turbulence effect in underwater imaging [4].

For the simplicity of analysis, it is assumed that both the object and the camera are stationary. Water is transparent and refractive. The distance between the object and the camera is assumed to be unknown. Furthermore, it is assumed that the water surface is large and the distribution of waves, of unknown magnitude and speed, on the surface is statistically stationary.

There are two cases here:

Case 1: When the water surface is still ($at\ time = t$)

The angle of the water surface normal N is zero. The camera therefore observes the object o at its original location. As a result, there will be no distortion in the captured image, except for the depth perception change due to the difference between the two refraction indices.

Case 2: When the water surface is wavy ($at\ time\ =\ t + 1$)

The normal to the water surface N is tilted by an angle. The apparent position of o changes to o' by a displacement factor with magnitude $||o - o'||$. The magnitude of this distortion is a function of the angle of the water surface normal at the point of incidence and the amplitude of the water waves.

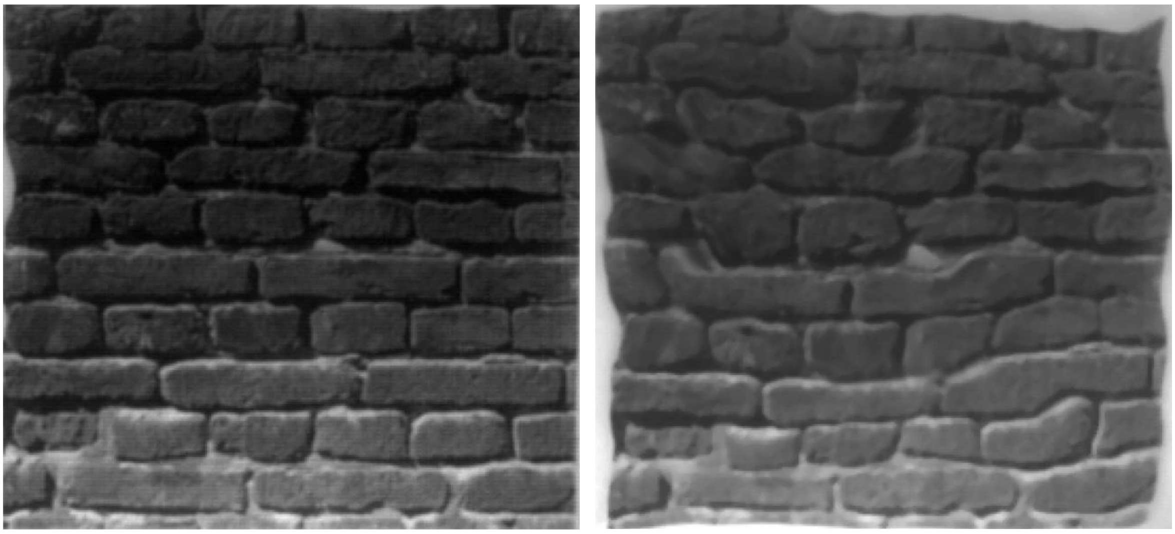


Figure 1.3: Acquired images through water surface: (i) when the water surface is still ($at\ time = t$); (ii) when the water surface is wavy ($at\ time = t + 1$) [9].

Figure 1.3 represents two images showing the consequence of refraction due to water waves at two different time t and $t + 1$. The captured image at time t appears correctly (undistorted), as noted from the normal text fonts on the image in Figure 1.3(i). The image at time $t + 1$ is shown in Figure 1.3(ii), which is, as expected, in a distorted form.

1.4 Motivation

In numerous applications, one would like to locate and track real moving objects in the scene. However, this task becomes difficult, when the scene contains motion not only due to the real moving objects, but also due to camera or turbulent light propagation medium motion.

Optical and mechanical video stabilization varies for each camera system and also expensive because it requires additional sophisticated hardware. On the other hand digital video stabilization uses previously captured sequences, and do not use any additional hardware and knowledge of physical motion of camera which makes it inexpensive and relatively easy to use.

The motivation behind the development of this video stabilization method is to create a general method that can effectively removes shakiness and turbulence effects by improving visual quality, and can be used for any camera system, with consideration for both robotic systems and handheld cameras. The goal is to provide a smoother video with turbulence removed, while unchanging moving objects present in it and also maintaining the desired quality. The smooth video should be more viewable without significantly affecting the understanding of the actual atmospheric condition.

1.5 Contributions

In this thesis work, the aim of the proposed method is to generate a systematic and effective mask by reducing the defect and erroneous noise due to turbulence in the medium. In this regard, three different masks are generated by applying three different thresholding techniques, and fusing them together a final and more accurate mask is generated to reduce the unwanted and erroneous portions present in each of the generated mask.

This approach mainly solves two different problems of imaging through turbulence: restoration of degraded videos and detection of moving objects present in this video. The main contribution of this thesis is the development of a turbulence degraded video stabilization method of high accuracy and low complexity.

For achieving high accuracy and low complexity, the development of proper mask generation technique that can be applied for any real-world turbulence degraded videos is another contribution in the field of image processing.

The proposed method is much more usable and extendible, and may have direct impact in actual real-time systems. Also the results of the thesis are relevant to current needs of the scientific community.

1.6 Objectives of the Thesis Work

This thesis focuses on developing a video stabilization method for processing real-world videos acquired in unknown environmental conditions for better presentation and characterization. The main objectives of this thesis are:

- To develop an improved model to improve the performance of moving object detection and video correction with negligible distortions.
- To evaluate the performance of the proposed method by applying them on different video sequences.
- To compare the performance of the proposed method with others.
- To experimental validate the proposed method by capturing real-world videos in the presence of turbulence and applying the method on them.

1.7 Outline of the Thesis Work

This thesis is structured in five chapters, starting with introduction that gives an overview of thesis. Chapter 1 gives a brief introduction to the atmospheric turbulence, video stabilization, and the purpose of doing the current work. The next subsection gives insight into the motivation for choosing the specific area. This chapter concludes with the outline of the thesis.

Chapter 2 provides a brief description of related research and theories. Chapter 3 discusses the proposed method. It includes frames extraction, difference frame generation, threshold determination, mask generation, and video stabilization. In chapter 4 experimental results are discussed and compared with those of an existing method to evaluate its performance. Chapter 5 concludes the thesis works with a direction of future research.

CHAPTER II

Literature Review

2.1 Introduction

Detecting moving targets flawlessly is very challenging because of several reasons and among them atmospheric turbulence is the most significant one, especially in long-range imaging [5]. Because of change in temperature in air and continuously flowing winds which randomly vary in direction and magnitude, the index of refraction of the medium throughout the atmospheric path varies randomly causing both the irregular movements and deformations of the pixels and makes the system distracted at the time of detecting objects [6-7].

The process of locating the desired object within some frame sequences, from its first appearance to its last is known as detection. Type of desired object within the system depends on the application. Sometime the desired object may be occluded by fixed obstacles or by the other objects within the scene. A detection system should be able to predict the position of any occluded objects. There is a diversity of methods which address the problem of restoring geometrically distorted videos containing real motions, and several others which do not consider the presence of moving objects.

2.2 Related Works

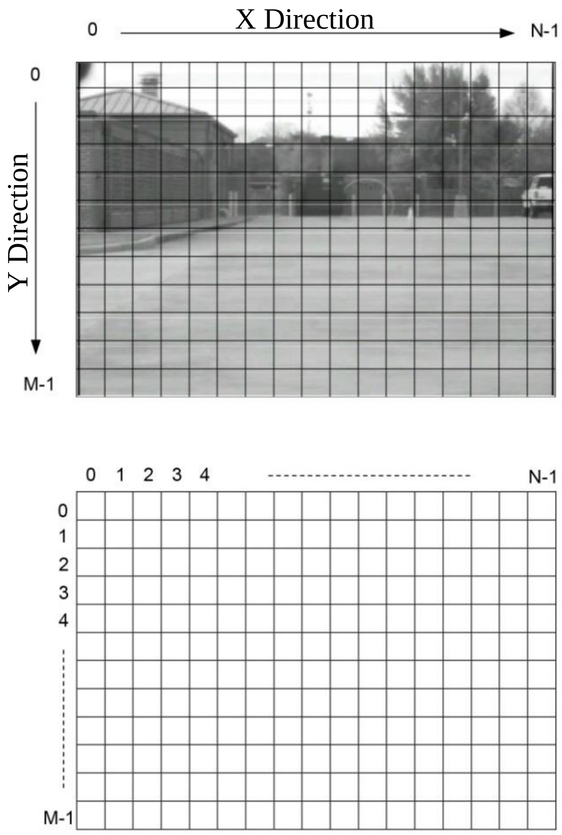
Among numerous methods that have been proposed for target detection, a few take background distortions into account caused by atmospheric turbulence. A method of stabilizing videos is proposed in [8] that can reconstruct a non-uniformly distorted video without altering the moving objects present in it. Firstly, a median filter is used to determine the background of a distorted video. Then, a mask is generated using the difference image, angle and amplitude of the motion vectors of each frame. For picking out the moving objects from the video sequence, the generated masks are used. Oreifej *et al.* [9] also proposed an approach to stabilize non-uniformly distorted videos and find out moving objects present in it. Low-rank matrix decomposition (LRMD) is included in this approach that gives three components such as background, moving objects and turbulence by dividing each frame and then processes them individually. Lower turbulence-induced motion is effectively separated by this method but it has a high computational cost. LRMD method uses a single Gaussian function to model turbulence thus it may not work under severe turbulence which is frequent in long range surveillance. From turbulence degraded images moving objects detection is done in [10] by extending decomposition of 2D cartoon-texture into 3D vector fields. The algorithm is implemented in curvelet spaces to efficiently explore the movement flow geometry. Zhang *et al.* [11] proposed a two stage restoration method that alleviates uncertain motions by conserving the moving objects present in videos. Representing each frame using a monogenic signal and producing a stable background using a low-pass filter are done in the first stage. Second stage involves the mask generation to preserve moving objects. Zhu and Milanfar [12] proposed an approach, where non-rigid image registration and blind deconvolution algorithm are used to alleviate atmospheric turbulence. Mao and Gilles [13] developed a unified framework using optical flow scheme and nonlocal-TV based regularization process for correcting distortion. To estimate the geometrical distortion, optical flow scheme is used and then original observed scene is recovered by regularization process. Lou *et al.* [14] proposed a method to reconstruct a sharp latent image and unwarp the video by applying the Sobolev gradient method. To sharpen the distinct frames, the Sobolev gradient method is applied and temporal distortions are mitigated by using Laplace operator. A software-based distortion-correction approach is proposed in [15] that could be applied for mitigating turbulence for both land and underwater imaging. In this method, the motions of large objects and turbulence-induced shifts are calculated by using an optical flow scheme. To get a stabilized video output, these motions are then used to reduce turbulence. Halder *et al.* [16] proposed a method for correcting geometric distortion that uses optical flow technique and a centroid algorithm. Optical flow technique is used for estimating motion vector of input frames and then a centroid algorithm is employed for correcting frames. Islam *et al.* [17] proposed a mask generation technique considering intensity and histogram. In this method, other properties like optical flow of distorted frames have not been considered. Elahi and Halder [18] considered intensity and optical flow for generating masks to detect moving objects but other properties like histogram have not been considered.

In this thesis work, three different properties (intensity, histogram and optical flow) are considered to generate a more accurate mask. Use of higher number of masks increases the computational complexity and the processing time. Thus, choosing appropriate mask generation technique is necessary for achieving research goal.

2.3 Image Sampling

Before manipulating an image using various image processing techniques, it needed to be sampled spatially. The process of applying a two-dimensional grid to a spatially continuous image to discretize it into a two-dimensional array of elements is known as the process of image sampling.

Figure 2.1 shows a sampled image having $N \times M$ sampled elements using a two dimensional rectangular grid. Any type of sampling grid can be used, but the rectangular grid is mostly used because it has a relationship to two-dimensional arrays. Picture element which is typically referred as a pixel is the fundamental unit of a sampled image.



$N \times M$ array of intensity values

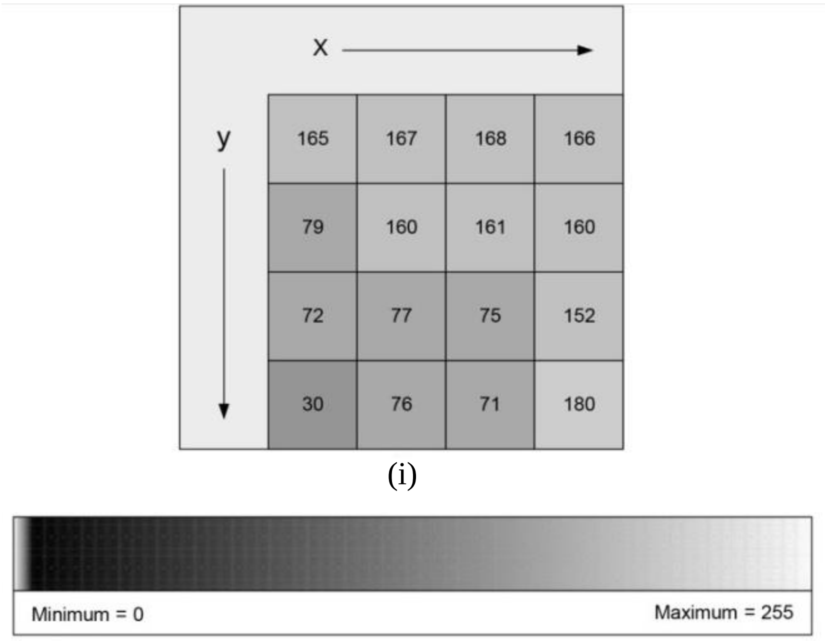
Figure 2.1: Sampled image containing $N \times M$ pixels [1].

The average intensity of the continuous spatial image covered by a pixel is the value of that pixel. The result of sampling produces a two-dimensional array of numbers that are directly proportional to the intensity levels of the continuous spatial image. Real-time video data is usually digitized over a 320×240, 640×480, 768×525, or 1600×1200 grids according to the context.

2.4 Quantization

Digitization of intensity value of each pixel into a finite set of numbers is necessary after spatial sampling. Digitization process converts analog intensity value of each pixel into some finite digital numbers. The quality of digitization is determined by the quantity of numbers used for representing pixel intensity. This number set is the gray levels of an image.

Since the spatial distribution of light energy is represented by an image, pixel values of digitized image take only positive values. A 4×4 sub-image of an entire image is shown in Figure 2.2 (i) and Figure 2.2 (i) shows the grayscale values from 0 to 255. The intensity value 0 indicates the absence of light i.e., black and 255 indicates highest intensity i.e., white.



165	167	168	166
79	160	161	160
72	77	75	152
30	76	71	180

(i)

(ii)

Figure 2.2: Example of (i) a sampled and digitized 4×4 sub-image and (i) corresponding gray-scale [1].

2.5 Converting Gray-Scale Images to Binary Image Using Thresholding

Converting a grayscale or color image to a binary image based upon a thresholding value different thresholding techniques are used in image processing. If the intensity value of a pixel in the image is less than predetermined threshold value k (i.e., $f(x,y) < k$), the corresponding pixel in the resulting image is set to 0 (black). On the other hand, if the intensity value of a pixel is greater or equal to the threshold intensity k (i.e., $f(x,y) \geq k$), the intensity value of corresponding pixel is set to 255 (white). In this way, it is used to convert a grayscale image to a binary image or an image with only 2 colors, black (0) and white (255). This can be formulated as follows:

$$f(x) = \begin{cases} 0, & f\,(x,y) < k \\ 255, & f\,(x,y) \geq k \end{cases} \tag{2.1}$$

This equation can be written in the form as follows:

$$f(x) = \begin{cases} G_a, \ f\,(x,y) < k \\ G_b, \ f\,(x,y) \geq k \end{cases} \tag{2.2}$$

where G_a and G_b are two desired gray levels for the image. For reducing a multilevel image to a two gray-level (G_a and G_b) image, thresholding process described by equation (2.2) can be used. For introducing multilevel threshold technique equation (2.2) can be expanded as follows:

$$f(x) = \begin{cases} G_a & 0 \leq f(x) < k_1 \\ G_b & k_1 \leq f(x) < k_2 \\ G_c & k_2 \leq f(x) < G_{max} \end{cases} \tag{2.3}$$

where maximum intensity of the image $f(x,y)$ is represented by G_{max}. For the 8-bit gray image this value is 255. And k_1, and k_2 are two different threshold values.

2.6 Histogram

Brightness histogram concisely describes the brightness characteristic of an image. Frequency distribution of pixels in an image is described by the brightness histogram. It provides a graphical representation of how many pixels within an image fall into a given intensity level.

A histogram is a graph having the horizontal axis from 0 to 255 (for 8-bit gray image) and vertical axis represents the “number of pixels”. For a particular brightness, total number of pixels having that intensity level can be found by simply looking up the intensity on the horizontal axis and reading off the vertical axis. Since each pixel has some brightness value to define them, the vertical axis adds up the number of pixels having same intensity in the image.

For the simplicity consider a digitized image having N pixels. Each of these pixels has been quantized into n levels in the range $d_0, d_1, \dots, d_{n-1}$. Figure 2.3 shows the histogram of this image.

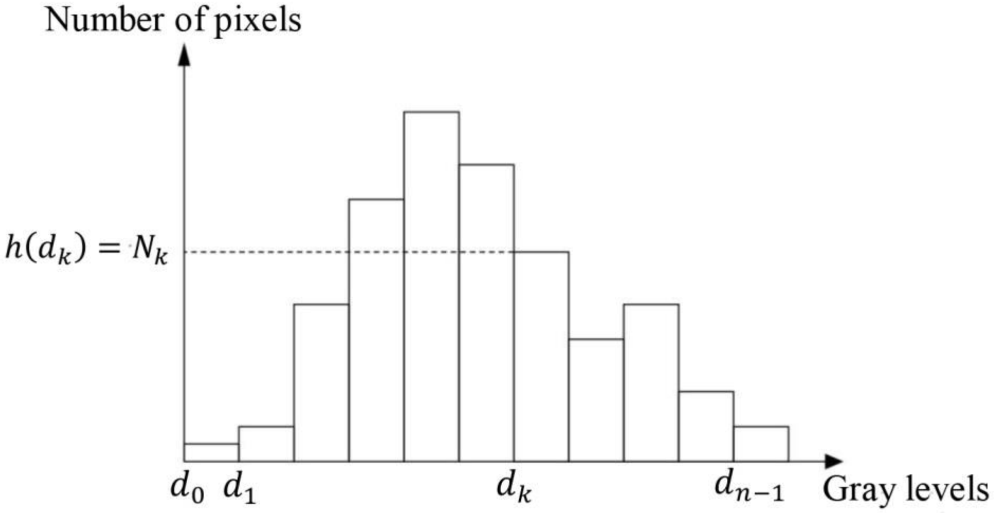


Figure 2.3: Simple representation of image histogram [1].

In the figure, $h(d_k)$ = the total number of pixels having same intensity level d_k and this can be written as:

$$h(d_k) = N_k, \tag{2.4}$$

where d_k is one of the gray levels between 0 to 255, and total number of pixels having the value d_k is represented by N_k.

2.7 Cumulative Histogram

The modified version of histogram is the cumulative histogram. In cumulative histogram vertical axis gives the total number of pixel for a specific intensity level added up with the number of pixels having intensity level lower than that intensity level.

Considering a digitized image having N pixels and each of these pixels has been quantized into n levels in the range $d_0, d_1, \ldots, d_{n-1}$. Figure 2.4 shows the cumulative histogram of an image.

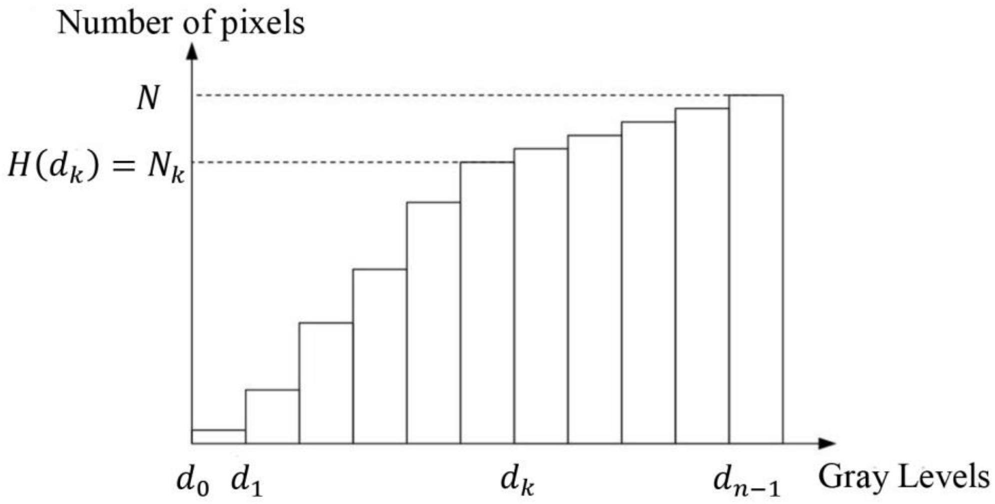


Figure 2.4: Cumulative histogram [1].

In the figure, $H(d_k)$ = the total number of pixels having equal or less than intensity level d_k. Hence,

$$H(d_k) = \sum_{i=0}^{k} h(d_i) = \sum_{i=0}^{k} N_i \tag{2.5}$$

Both the cumulative histogram and histogram are step functions.

Since the total number of pixel in the image is $\sum_{i=0}^{n-1} N_i = N$, the cumulative histogram $H(d_k)$ increases from 0 to N.

2.8 Background Subtraction

In digital image processing, the common and most popular approach for motion estimation is background subtraction. Taking a reference image as background that is updated with time and subtracting this background from each image gives difference image. This process is known as background subtraction. Difference image is calculated as

$$D(x, y) = |I(x, y) - B(x, y)|, \tag{2.6}$$

where *D(x,y), I(x,y),* and *B(x,y)* represent the difference image, input image, and reference background image, respectively.

For stationary cameras without any type of turbulence effect the subtraction process gives only stationary objects including silhouette region of that object. This is simple approach for real-time systems but for the case of changing brightness and presence of turbulence in dynamic scene this approach is extremely sensitive. Thus, good background maintenance is necessary to get better processing performance. Automatically updating the background from the incoming video frame is the main problem and it should be able to overcome the following problems [19]:

- **Motion in the background:** Non-stationary background such as leaves and branches of trees, waving flag in wind, water flowing, should be identified as part of the background.
- **Illumination changes:** It is necessary to adapt the the background model over a period of time with the change in illumination.
- **Memory:** Use of memory and computing power by the background model should be less.
- **Shadows:** Due to moving object some shadow also creates. This shadows should be consider as background.
- **Camouflage:** For the similar pixel characteristics of moving object and background the object should be detected.
- **Bootstrapping:** The background model should be able to maintain background even if the background training is absent.

2.9 Frame Difference

Pixel-wise differencing between two consecutive frames is known as frame difference. It is important for detecting moving object region in an image or video sequence. Change in difference frames is determined by threshold function and it depends on the speed of object motion. Maintaining the quality of segmentation is hard if the object speed changes significantly. Developing holes inside moving object is very common in frame differencing technique but is very adaptive to dynamic environments.

2.10 Thresholding

Setting all pixels to a foreground value whose intensity values are above a threshold and to a background value for all the remaining pixels to segment an image is called thresholding.

Conventional thresholding technique uses global threshold for all pixels but in adaptive thresholding, the value of threshold changes dynamically over the image considering the change in lighting condition in the image.

Based on the manipulating information of the algorithm Sankur and Sezgin classify thresholding techniques into following six groups [20]:

- **Histogram shape**-based methods, where the curvatures, valleys and peaks of the histogram are analyzed.
- **Clustering-**based methods, where the gray-level samples are clustered in two parts as foreground (object) and background.
- **Entropy**-based methods, where the entropy of the foreground and background regions are used in algorithms.
- **Object Attribute**-based methods, where a measure of similarity between the gray-level and the binarized images, such as edge coincidence, fuzzy shape similarity, etc. are searched by the algorithms.
- **Spatial** methods use correlation between pixels and/or higher-order probability distribution.
- **Local** methods use local image characteristics to adapt the threshold value on each pixel. In this technique, a different threshold value is selected for each pixel in the image.

Global and adaptive thresholding are the main two types of thresholding techniques. In global thresholding technique only one threshold value is selected based on the image histogram for the entire image. When local properties of some regions in an image are considered to determine threshold value then it is called local thresholding. If the selection of local threshold values is independent for each pixel or for a group of pixels the thresholding is called adaptive thresholding or dynamic thresholding.

2.10.1 Global thresholding

The simplest property that pixels in a region of an image can share is intensity. Separating light and dark regions through thresholding is the natural way to segment an image. A gray-scale image converts into binary image through thresholding technique. Pixels below the threshold value turn into zero and pixels above that value turns into one.

Applying a global threshold T in an image $f(x,y)$, the thresholded version $g(x,y)$ can be written as,

$$g(x,y) \ = \begin{cases} 1 & if\ f(x,y) \geq \ T \\ 0 & otherwise \end{cases} \tag{2.7}$$

2.10.2 Adaptive thresholding

Taking a color or gray-scale image as input and generating a binary output image for representing the segmentation is done in adaptive thresholding. For each pixel in the image, a threshold has to be calculated and pixel having the value above the threshold is set to foreground value. On the other hand pixel having the value below the threshold is set to foreground value.

Two main approaches for finding the threshold are [21]:

i. Chow and Kaneko approach
ii. Local thresholding.

For the both methods it is assumed that smaller image regions are of uniform illumination which is more suitable for thresholding. Chow and Kaneko divide an image into an array of overlapping sub-images and then find the optimum threshold for each sub-image by investigating its histogram. The threshold for each single pixel is found by interpolating the results of the sub-images. Computational complexity is the main drawback of this method and it is inappropriate for real time application [21].

Examining the neighborhood pixels intensity value of each pixel is an alternative approach for finding local threshold. The most appropriate statistic depends largely on the input image. Simple and fast functions include the mean of the local intensity distribution,

$$T = \text{mean}$$

the median value,

$$T = \text{median}$$

or the mean of the minimum and maximum values,

$$T = \frac{\text{max} + \text{min}}{2}$$

If the size of the neighborhood is not large enough to cover sufficient background and foreground pixels, a poor threshold will be chosen. On the other hand, choosing large regions violate the assumption of uniform illumination. This method gives better result than Chow and Kaneko approach for some application due to less computational burden.

2.11 Hysteresis Thresholding

For the segmentation of monochrome image one of the widely used techniques is histogram thresholding.

Assuming the images as the combination of different gray-level ranges the histogram can be separated into a number of peaks. Each of the peaks corresponding to one region and there exists a threshold value corresponding to valley between the two adjacent peaks. By splitting the gray-level ranges into equal-sized classes histograms are constructed. Then for each class, the number of points from the data set that fall into each class is counted.

Consider an image $f(x, y)$ which is composed of dark background and light object on this background. The image $f(x, y)$ is composed in such a way that background pixel and object have gray levels grouped into two dominant modes and the gray-level histogram corresponds to this image. Selecting a threshold T is necessary to separate two modes and extract the desired object from the background. Since the object is lighter than the background thus, for any point (x, y) whose gray-level intensity is greater than the threshold value T is considered as the pixel of object. On the other hand if the gray-level intensity of a pixel is lesser than the threshold value T is considered as a background pixel. The major drawback of this process is that: there exist only two classes and this method cannot deal with multichannel images. To overcome this problem different thresholding technique have been introduced that segments an image by using the information based on local intensities and connectivity.

i. **Bimodal Histogram:** When only two different classes are present in an image then only one threshold is enough for making partition between classes and generated histogram by using this single threshold is known as bimodal histogram.

ii. **Multimodal Histogram:** If, there exist many different classes in an image then the histogram will have no clear valley. It means that several object pixels and background pixels have similar gray-level value. In this case single threshold is not enough for separating classes. Pixels below the low threshold value and above the high threshold value are classified as object and background, respectively. For the remain case the pixels are considered as object if they are adjacent to other object.

2.12 Multiband Thresholding

For separating color images each of the RGB components is thresholded by different threshold and then AND operation is employed to combine them. This reflects the working of the camera and data storing in the computer. But it is different from color recognition by people. Since the HSL (hue, saturation, lightness) and HSV (hue, saturation, value) color models which are alternative representations of the RGB color model are more often used and hue is a circular quantity thus it requires circular thresholding. It is also possible to use the CMYK (cyan, magenta, yellow, and key (black)) color model [22].

2.13 Optical Flow

One of the most important parts of this thesis is optical flow estimation and it is used in mask generation process. Image motion at each pixel because of the variation of brightness is known as optical flow. It can also be defined as the distribution of apparent velocities of movement of brightness pattern in an image. Due to relative motion between viewer and objects, optical flow can arise. The information about the arrangement of the objects viewed and the rate of change of this arrangement can get from optical flow.

Optical flow cannot be calculated locally. The flow velocity has two components but for a point in the image sequence only one independent measurement is available. Thus it required a second constraint to calculate optical flow. Unfortunately, motion field cannot be observed directly, since the movement of the image projection of a 3D point is unknown. On the other hand, only image points can be observed. Under certain assumption, it can be said that an image point moves from here to there, which indicates optical flow. It is obvious that,

$$\text{motion field} \neq \text{optical flow}$$

Consider a perfectly uniform sphere as in Figure 2.5. Some shadows also present on the surface of the sphere. At the time of rotation of the sphere the shading pattern remains unchanged and thus the optical flow is zero but there exists a motion field. On the other hand, if the lighting source moves but the sphere remains still then the shading pattern will change. In this case, the motion field is zero but the optical flow is not zero. A stationary sphere with a moving light source produces drifting intensities.

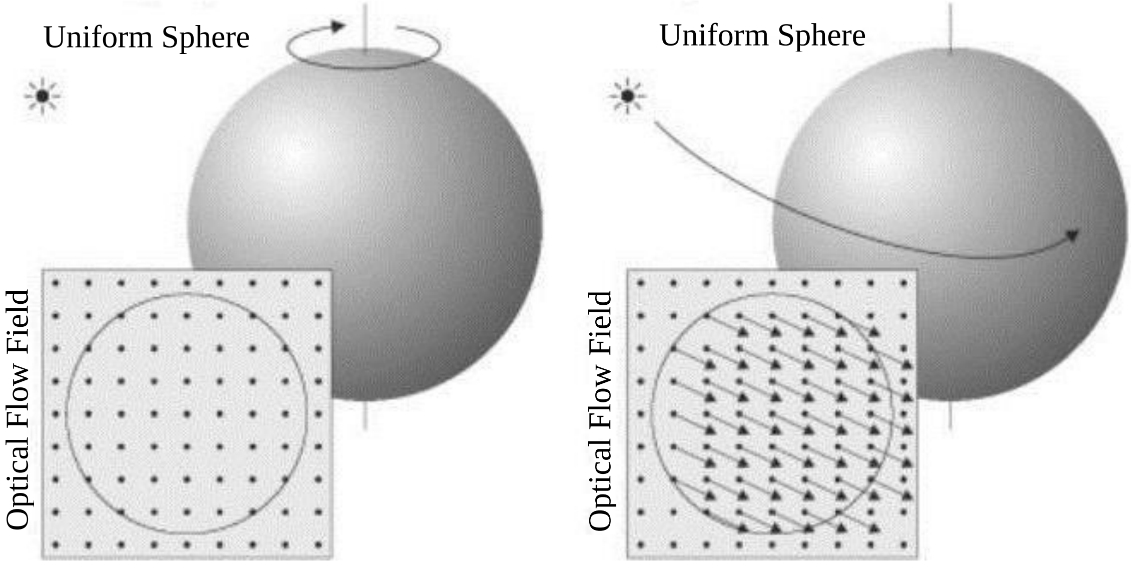


Figure 2.5: Optical flow for a uniform sphere.

Since optical flow can be observed, it is expected that optical flow is not too different from the motion field.

2.13.1 Optical flow constraint equation

Consider two image frames taken at times t and $t + \Delta t$. The motion between these image frames at every pixel position is needed to calculate by optical flow method. Local Taylor series approximations of the image signal are used in this method and they use partial derivatives with respect to the spatial and temporal coordinates.

For a $2D + t$ dimensional case (n-D cases are similar) a pixel at location (x, y, t) having intensity $I(x, y, t)$ will have moved by Δx, Δy and Δt between the two image frames. For this assumption the brightness constancy constraint can be given as:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{2.8}$$

Considering small movement for pixels between two frames and the brightness changes smoothly with x, y and t, the image constraint at $I(x, y, t)$ by expanding the right-hand-side by Taylor series we get:

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + H.O.T.$$

$$\approx I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t \qquad (2.9)$$

From these equations it can be written that:

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = I(x+\Delta x, y+\Delta y, t+\Delta t) - I(x,y,t)$$

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t \approx 0$$

$$\frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t} \approx 0$$

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} \approx 0, \qquad 2.10)$$

where x and y component of the velocity or optical flow of $I(x,y,t)$ is represented by V_x and V_y, respectively. And the derivatives of the image at (x,y,t) in the corresponding directions are denoted by $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, and $\frac{\partial I}{\partial t}$. These derivatives can be represented by I_x, I_y, and I_t.

Thus,

$$I_xV_x + I_yV_y + I_t \approx 0 \qquad (2.11)$$

In the limit as Δx and Δy go to zero, this becomes exact

$$I_xV_x + I_yV_y + I_t = 0$$

$$\nabla I\,.\begin{bmatrix}V_x & V_y\end{bmatrix}^T + I_t = 0$$

$$\nabla I\,.V_m + I_t = 0, \qquad (2.12)$$

where the image gradient at pixel $[x,y]^T$ and velocity of this pixel are represented by $\nabla I = \begin{bmatrix}\frac{\partial I}{\partial x} & \frac{\partial I}{\partial y}\end{bmatrix}^T$ and $V_m = \begin{bmatrix}V_x & V_y\end{bmatrix}^T$, respectively. $I_t = \frac{\partial I}{\partial t}$ and image gradient ∇I can be obtained from images. This equation is the optical flow constraint equation with two unknowns.

Apparently, for each pixel, there exist one constraint equation, but it is needed to solve two unknowns, i.e., V_x and V_y, which means that only from such constraint equation optical flow cannot be determined. This is known as "aperture problem" of the optical flow algorithms. Another set of equations by some additional constraint are required to find the optical flow.

From the optical flow constraint equation we can write

$$V_y = -\frac{I_x}{I_y}V_x - \frac{I_t}{I_y}$$
$$V_x = -\frac{I_y}{I_x}V_y - \frac{I_t}{I_x} \tag{2.13}$$

When $V_x = 0$ then the value of $V_y = -\frac{I_t}{I_y}$ and when $V_y = 0$ then the value of $V_x = -\frac{I_t}{I_x}$. This helps to draw the constraint line. Thus the component of the movement in the direction of the brightness gradient (I_x, I_y) which is known as the normal flow can be calculated as:

$$\text{Normal Flow} = \frac{I_t}{\sqrt{{I_x}^2+{I_y}^2}} \tag{2.14}$$

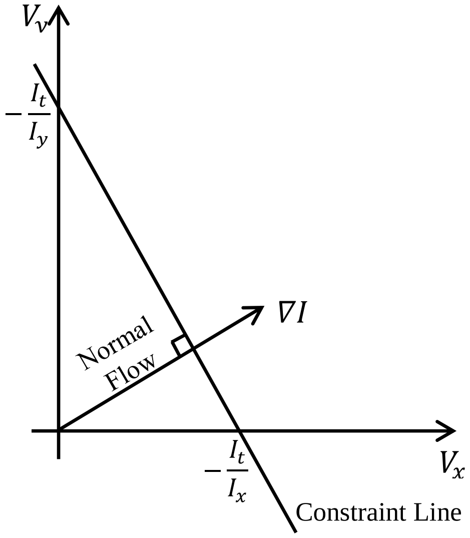


Figure 2.6: Geometrical explanation of the optical flow constraint equation [23].

Figure 2.6 gives a geometrical explanation of the constraint equation. For a given image pixel the optical flow can be any point on the constraint line. And only "normal flow" can be determined from this constrain equation, i.e., the flow along the direction of image gradient, but we cannot determine the component of movement in the tangent direction of the iso-brightness contour, i.e., the direction perpendicular to the image gradient. As a consequence, the flow velocity (V_x, V_y) cannot be computed locally without introducing additional constraints.

2.13.2 Aperture problem

From optical flow constraint equation it cannot possible to determine the direction of movement of a pixel perpendicular to the image gradient which is known as the aperture problem. This indicates the movement in the tangent direction of the iso-brightness contour [23]. The variation of perceived motion from the actual motion is known as aperture problem. For example shown in Figure 2.7 consider, the contours are iso-brightness. There are many points around p that have the same intensity. In this case which point p' on C' match point p on C is impossible to determine.

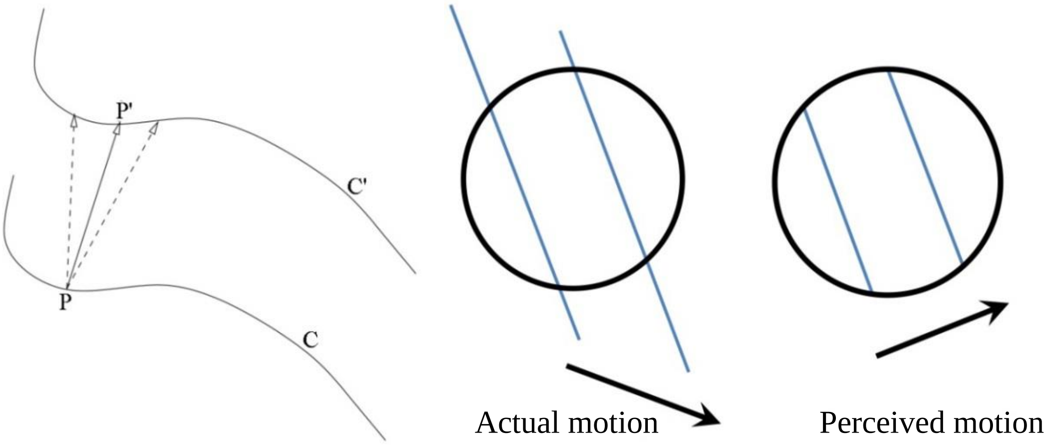


Figure 2.7: Aparture problem of optical flow [23].

As a result, some other constraints are needed to determine the optical flow uniquely. Two commonly used approaches are described below: the first one introduces a global smoothness constraint, and the second one employs a local constraint for each pixel.

2.13.3 Horn-Schunck's method

Independent movement of brightness pattern for every pixel makes it impossible to recover velocities. Considering similar velocities for neighboring points on the object the brightness pattern varies smoothly almost everywhere in the image. If one object occludes another there is a probability of discontinuity [24].

One way to express the additional constraint is to minimize the square of the magnitude of the gradient of the optical flow velocity:

$$\left(\frac{\partial V_x}{\partial x}\right)^2 + \left(\frac{\partial V_x}{\partial y}\right)^2 \text{ and } \left(\frac{\partial V_y}{\partial x}\right)^2 + \left(\frac{\partial V_y}{\partial y}\right)^2 \tag{2.15}$$

Another measure of the smoothness of the optical flow field is the sum of the squares of the Laplacians of the x- and y-components of the flow. The Laplacians of V_x and V_y are defined as

$$\nabla^2 V_x = \frac{\partial^2 V_x}{\partial x^2} + \frac{\partial^2 V_x}{\partial y^2} \text{ and } \nabla^2 V_y = \frac{\partial^2 V_y}{\partial x^2} + \frac{\partial^2 V_y}{\partial y^2}$$

For simple situation both Laplacians are zero but if the viewer travels orthogonally to the surface, then the second partial derivatives of both V_x and V_y will be vanished.

The measure of departure from smoothness can be written by:

$$e_s = \iint \left(\|\nabla V_x\|^2 + \|\nabla V_y\|^2\right) dx\, dy$$

$$= \iint \left(\left(\frac{\partial V_x}{\partial x}\right)^2 + \left(\frac{\partial V_x}{\partial y}\right)^2 + \left(\frac{\partial V_y}{\partial x}\right)^2 + \left(\frac{\partial V_y}{\partial y}\right)^2\right) dx\, dy \tag{2.16}$$

The error in the equation of optical flow for the rate of change of image brightness is

$$e_b = \iint \left(\nabla I . V_m + \frac{\partial I}{\partial t}\right)^2 dx\, dy \tag{2.17}$$

The total error to be minimized is

$$e = e_b + \lambda e_s$$

$$= \iint \left(\left(\nabla I . V_m + \frac{\partial I}{\partial t}\right)^2 + \lambda\left(\|\nabla V_x\|^2 + \|\nabla V_y\|^2\right)\right) dx\, dy \tag{2.18}$$

By finding suitable values of (V_x, V_y) the minimization is accomplished. Using the calculus of variation we get

$$I_x\left(I_xV_x + I_yV_y + I_t\right) + \lambda(\nabla^2V_x) = 0$$
$$I_y\left(I_xV_x + I_yV_y + I_t\right) + \lambda\left(\nabla^2V_y\right) = 0 \tag{2.19}$$

Since, $(i + 1, j)$, $(i, j + 1)$, $(i - 1, j)$, $(i, j - 1)$, are the 4-neighbor of an image pixel (i, j) thus, the discrete version of smoothness error can be written for each pixel. The discrete version of the Laplacian's is calculated from the difference of the velocity at a point and the average of its neighbors as

$$I_x\left(I_xV_x + I_yV_y + I_t\right) + \lambda\left(V_x - V_{x_{av}}\right) = 0$$
$$I_y\left(I_xV_x + I_yV_y + I_t\right) + \lambda\left(V_y - V_{y_{av}}\right) = 0 \tag{2.20}$$

From this two equations $V_x\ and\ V_y$ can be calculated as

$$V_x = V_{x_{av}} - I_x\frac{P}{D}$$
$$V_y = V_{y_{av}} - I_y\frac{P}{D}, \tag{2.21}$$

where $P = I_xV_x + I_yV_y + I_t$ and $D = \lambda + {I_x}^2 + {I_y}^2$.

For iterative scheme these can be written as

$${V_x}^{k+1} = {V_x}^{k} - I_x\frac{P}{D}$$
$${V_y}^{k+1} = {V_y}^{k} - I_y\frac{P}{D} \tag{2.22}$$

At the beginning V_x and V_y both are zero. Initializing with $k = 0$ and increasing the value of k to repeate until it converges.

2.13.4 Lucas-Kanade's method

From optical flow equation it can be written that

$$I_xV_x + I_yV_y = -I_t \tag{2.23}$$

In matrix form it can be written as

$$[I_x \quad I_y]\begin{bmatrix}V_x\\V_y\end{bmatrix} = -[I_t] \tag{2.24}$$

Considering smoothly varied brightness pattern and using $N \times N$ windows it gives N^2 equations per pixel as

$$\underset{\substack{A\\N\times 2}}{\begin{bmatrix}I_x(P_1) & I_y(P_1)\\ I_x(P_2) & I_y(P_2)\\ \vdots & \vdots\\ I_x(P_N) & I_y(P_N)\end{bmatrix}} \underset{\substack{V\\2\times 1}}{\begin{bmatrix}V_x\\V_y\end{bmatrix}} = -\underset{\substack{I_t\\N\times 1}}{\begin{bmatrix}I_t(P_1)\\ I_t(P_2)\\ \vdots\\ I_t(P_N)\end{bmatrix}} \tag{2.25}$$

This is in the form

$AV = I_t$

$or,\ (A^T A)V = A^T I_t$

Now we have more equations than unknowns. Minimization of least squares error is obtained by taking the summations over all pixels in the $k \times k$ window as

$\min \sum_i \left(I_{xi}V_x + I_{yi}V_y + I_t\right)^2$

Differentiating this we get

$$\begin{gathered}\sum\left(I_{xi}V_x + I_{yi}V_y + I_{ti}\right)I_{xi} = 0\\ \sum\left(I_{xi}V_x + I_{yi}V_y + I_{ti}\right)I_{yi} = 0\end{gathered} \tag{2.26}$$

or,

$$\begin{gathered}\sum I_{xi}{}^2 V_x + \sum I_{xi}I_{yi}V_y = -\sum I_{xi}I_{ti}\\ \sum I_{xi}I_{yi}V_x + \sum I_{yi}{}^2 V_y = -\sum I_{yi}I_{ti}\end{gathered} \tag{2.27}$$

In matrix form it can be written as

$$\begin{bmatrix}\sum I_{xi}{}^2 & \sum I_{xi}I_{yi}\\ \sum I_{xi}I_{yi} & \sum I_{yi}{}^2\end{bmatrix}\begin{bmatrix}V_x\\V_y\end{bmatrix} = \begin{bmatrix}-\sum I_{xi}I_{ti}\\ -\sum I_{yi}I_{ti}\end{bmatrix} \tag{2.28}$$

or,

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum {I_{xi}}^2 & \sum I_{xi}I_{yi} \\ \sum I_{xi}I_{yi} & \sum {I_{yi}}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum I_{xi}I_{ti} \\ -\sum I_{yi}I_{ti} \end{bmatrix} \tag{2.29}$$

or,

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \frac{1}{\sum {I_{xi}}^2 \sum {I_{yi}}^2 - \left(\sum I_{xi}I_{yi}\right)^2} \begin{bmatrix} \sum {I_{yi}}^2 & -\sum I_{xi}I_{yi} \\ -\sum I_{xi}I_{yi} & \sum {I_{xi}}^2 \end{bmatrix} \begin{bmatrix} -\sum I_{xi}I_{ti} \\ -\sum I_{yi}I_{ti} \end{bmatrix} \tag{2.30}$$

Thus,

$$\begin{aligned} V_x &= \frac{-\sum {I_{yi}}^2 \sum I_{xi}I_{ti} + \sum I_{xi}I_{yi} \sum I_{yi}I_{ti}}{\sum {I_{xi}}^2 \sum {I_{yi}}^2 - \left(\sum I_{xi}I_{yi}\right)^2} \\ V_y &= \frac{\sum I_{xi}I_{yi} \sum I_{xi}I_{ti} - \sum {I_{xi}}^2 \sum I_{yi}I_{ti}}{\sum {I_{xi}}^2 \sum {I_{yi}}^2 - \left(\sum I_{xi}I_{yi}\right)^2} \end{aligned} \tag{2.31}$$

For determining the flow of a particular image point Lucas-Kanade's method [25] uses a local window so it is called a local method [26].

2.14 Masking

The way of setting some of the pixel values in an image to zero, or some other “background” value is known as masking. A mask image is simply an image where some of the pixel intensity values are zero, and others are non-zero. In computer graphics and image processing when there is an intention to place an image over another image or background, a binary mask is used to specify the transparent area of the intended image. Two bitmaps are used in this purpose. In the actual image the pixel values of unused or transparent areas are set to 0’s and and in the background image pixel values of corresponding areas of actual image are set to 0’s. After that bitwise addition is applied between two masks. This preserves background pixels on transparent areas and set up the actual image over the background image. In this way, one image is placed over another image and this gives a perfect compound two image. This technique is widely used for pointing device cursors, image mixing applications, for creating characters in typical two dimensional videogames and in many other image processing works.

CHAPTER III

Proposed Method

3.1 Introduction

An improved method is proposed in this chapter for stabilizing turbulence degraded videos. Generally, in turbulence degraded videos, the pixels having moving objects have higher intensity differences than the other parts of a frame. In this work, all the masks are generated based on this basic concept.

A simplified block diagram of the proposed method is shown in Figure 3.1. In this method, at first, the input video is sampled into several frames. Three different techniques are used here to generate three different masks. To obtain a refined mask, these masks are combined together which is then applied to each frame for separating moving objects from background. The segmented moving objects are then fused with the background frame to generate the stabilized frames, thereby produced a stabilized video.

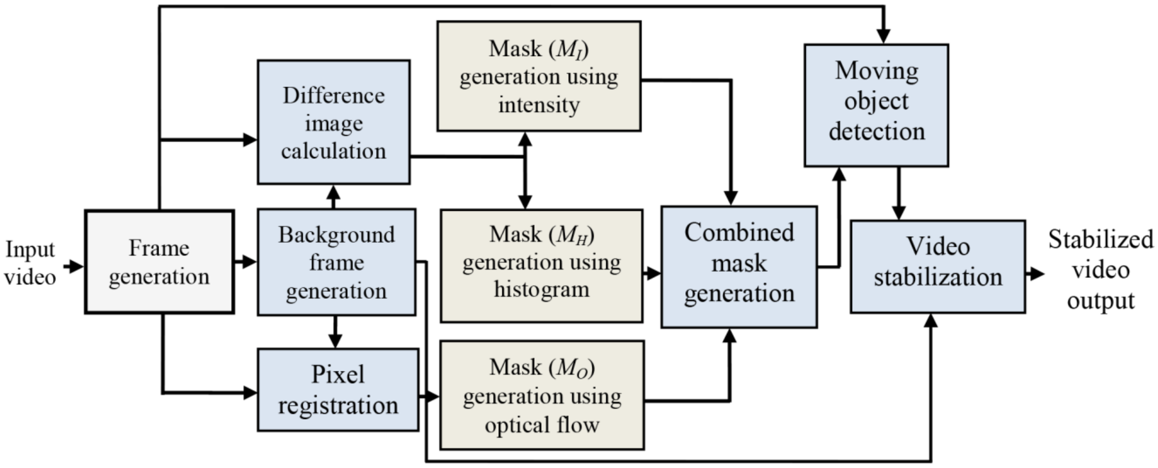

Input video
Frame generation
Difference image calculation
Background frame generation
Pixel registration
Mask (M_I) generation using intensity
Mask (M_H) generation using histogram
Mask (M_O) generation using optical flow
Combined mask generation
Moving object detection
Video stabilization
Stabilized video output

Figure 3.1: Block diagram of the proposed video stabilization method.

3.2 Difference Image Extraction

Since turbulence degraded video is taken as input, each of the video frame is randomly distorted by turbulence. The frames extracted from the input video are considered as input frames $I(x, y)$. Each frame has the salient part of moving object. A median operation is applied to determine the background $B(x, y)$. Subtracting the background from all input image frames gives the difference images.

The difference image is calculated as

$$D(x, y) = |I(x, y) - B(x, y)|, \tag{3.1}$$

where *D(x,y), I(x,y),* and *B(x,y)* represent the difference image, input image, and background image, respectively. In the difference image, higher intensity regions exhibit the moving object regions while lower intensity regions represent turbulence-induced motion. Therefore, removing low intensity regions is necessary and it can be done by using thresholding techniques.

3.3 Mask Generation Techniques

A mask image is simply an image, where some of the pixel intensity values are zero and others are non-zero. Masking is used to isolate features within an image above, below or equal to a specified pixel value and this value is known as the threshold level which determines how masking occurs. For determining three different masks, the threshold levels are calculated using three different techniques. These are:

(a). Mask generation using intensity

(b). Mask generation using histogram

(c). Mask generation using optical flow

3.3.1 Mask generation using intensity

Using the difference image of each frame the first mask is generated. A two-level threshold is calculated from the difference image, which is used to generate the first mask M_I. For adaptive thresholding, the threshold values are calculated using the statistics of the difference image. Assuming, T_{Il} and T_{Ih} as the low-level and high-level thresholds, respectively, their values are calculated as

$$\begin{aligned} T_{Il} &= g_I \bar{\iota} + K_l \\ T_{Ih} &= T_{Il} + K_h, \end{aligned} \tag{3.2}$$

where g_I is a gain, and K_l and K_h are two offsets. The values of gain and offsets are calculated by using an iterative algorithm for which lower MSE and higher PSNR of the stabilized video sequences are achieved. The values of gain and offsets are obtained 3 and 0.1, respectively, for the intensity range [0–1]. Also, i¯represents the mean intensity of the difference image, which is calculated as

$$\bar{\iota} = \frac{1}{X \times Y} \sum_{x=1}^{X} \sum_{y=1}^{Y} |D(x,y)|, \tag{3.3}$$

where size of the frame is represented by $X \times Y$.

Mask M_I is then generated using the two-level threshold calculated from equation (3.2) as

$$M_I(x,y) = \begin{cases} 0 & if\ D(x,y) \le T_{Il} \\ \frac{D(x,y)}{T_{Ih}} & if\ T_{Il} < D(x,y) < T_{Ih} \\ 1 & if\ D(x,y) \ge T_{Ih} \end{cases} \tag{3.4}$$

The accuracy of the mask depends greatly on the values of the gain and offsets. In the case when a low gain is chosen against a high strength of turbulence, the mask may contain too much undesired regions that the moving objects remain undetectable. On the other hand, selecting a high gain may lead the moving object regions being truncated or shows no object in the moving object regions.

3.3.2 Mask generation using histogram

A single mask may contain several falsely detected regions and therefore may not detect the moving object regions correctly. Thus, a second mask is generated using a threshold T_{hist} determined from the Otsu method [27]. This method calculates a single-valued threshold by minimizing intra-class intensity variance.

Otsu method is a global thresholding technique for selecting threshold between foreground and background. It is a region based segmentation method that follows bi-modal histogram technique. For separating two groups and achieving minimal intragroup variance the method calculates a threshold value [27].

Before starting the main process, computing a gray-level histogram is needed for calculating threshold by the Otsu method. Consider L gray level $[0, 2, \cdots, \mathrm{L}-1]$ to represent the pixels of an input image. The total number of pixels is N and n_i is used to denote each pixel. The probability distribution of gray-level histogram can be written as

$$P_i = \frac{n_i}{N}, \qquad P_i \geq 0, \qquad \sum_{i=o}^{L-1} P_i = 1 \tag{3.5}$$

The pixels of gray-level are divided into two groups as background and foreground. It is done by a threshold at level k, where two class C_1 and C_2 denote the pixels of level $[0,\cdots,k-1]$ and $[k,\cdots,L-1]$, respectively.

$$\begin{aligned} C_1 &= \{(x,y) | I(x,y) < K\} \\ C_2 &= \{(x,y) | I(x,y) \geq K\} \end{aligned} \tag{3.6}$$

The probability of class occurrence are given by

$$\begin{aligned} P_1 &= \sum_{i=0}^{K-1} (P_i) \\ P_2 &= \sum_{i=K}^{L-1} (P_i) = 1 - P_1 \end{aligned} \tag{3.7}$$

The class-mean levels are given by

$$\begin{aligned} M_1 &= \frac{\sum_{i=0}^{K-1} (iP_i)}{P_1} \\ M_2 &= \frac{\sum_{i=K}^{L-1} (iP_i)}{P_2} \end{aligned} \tag{3.8}$$

Hence the class variances are,

$$\begin{aligned} V_1 &= P_1(M_1 - M_g)^2 \\ V_2 &= P_2(M_2 - M_g)^2, \end{aligned} \tag{3.9}$$

where M_g is the mean average, and calculated as

$$M_g = \sum_{i=0}^{L-1} (iP_i) \tag{3.10}$$

According to Otsu’s criterion, minimizing the intra-class variance is equivalent to maximizing inter-class variance. Thus the most appropriate threshold value T_{hist} is the value of K for which V_t will be maximum. And V_t is calculated as

$$V_t = V_1 + V_2 \tag{3.11}$$

Using T_{hist} the second mask M_H is generated as

$$M_H(x,y) = \begin{cases} 0 & if\ D(x,y) < T_{hist} \\ 1 & if\ D(x,y) \geq T_{hist} \end{cases} \tag{3.12}$$

3.3.3 Mask generation using optical flow

From the shifts of the pixels of each frame with respect to the background another mask is generated for improving the accuracy of final mask. Each point in the reference background frame and input frame are related by a geometric relationship except the point of moving object. The general mapping function can be given in two forms: either relating the output coordinate system to that of the input, or vice versa. An optical flow estimation technique is employed for this purpose which estimates the x- and y- directional pixel shiftmaps [28]. The process of optical flow estimation is shown in Figure 3.2. In this figure, a pixel of input frame is shifted by 1 pixel in x direction and 2 pixels in negative y direction. For this case the values of S_x and S_y are 1 and -2, respectively. Figure 3.3 shows the pixel registration process for a sequence of input frames.

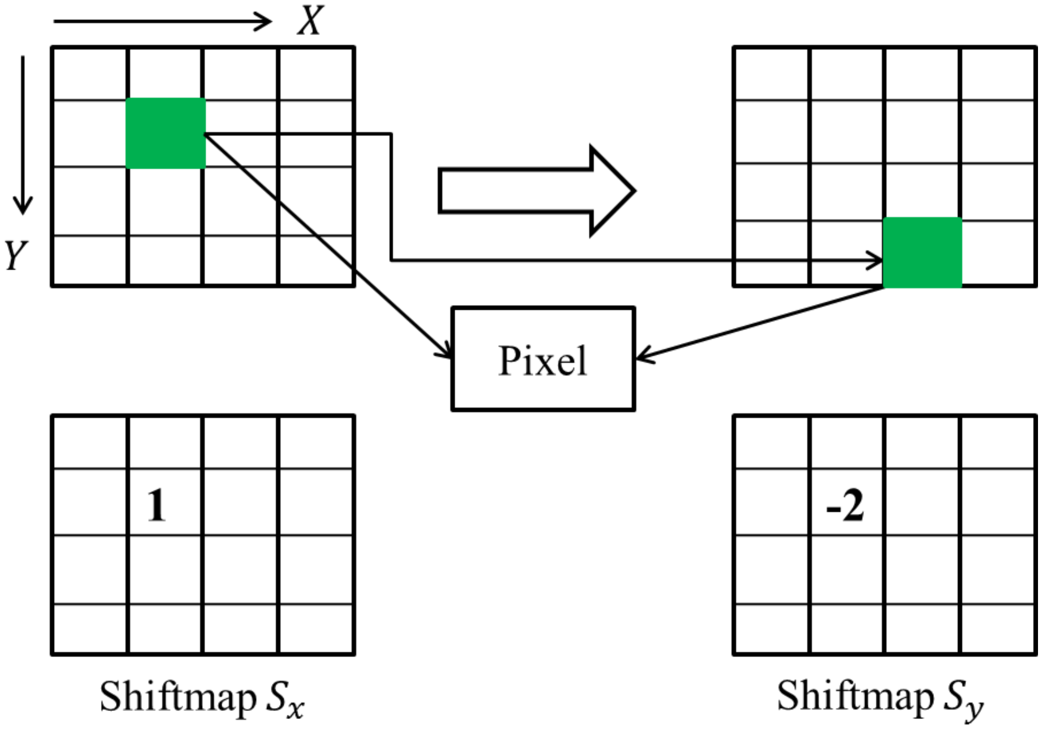


Figure 3.2: Simple illustration of image registration to obtain the shiftmaps.

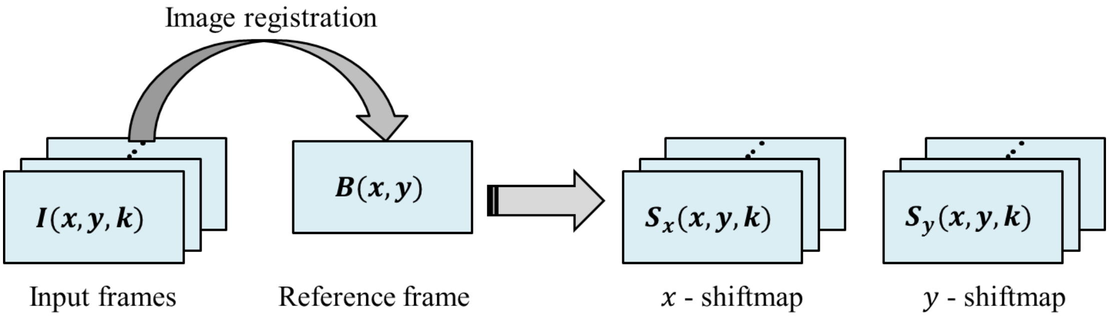


Image registration
$\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{k})$
$\boldsymbol{B}(\boldsymbol{x},\boldsymbol{y})$
$\boldsymbol{S}_{\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{k})$
$\boldsymbol{S}_{\boldsymbol{y}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{k})$
Input frames
Reference frame
x - shiftmap
y - shiftmap

Figure 3.3: Image registration process for input sequences [4].

Using the x- and y- shiftmaps, the resultant shiftmap S is calculated as

$$S(x,y) = \sqrt{{S_x}^2(x,y) + {S_y}^2(x,y)}, \tag{3.13}$$

where S_x and S_y are the x-directional and y-directional pixel shiftmaps, respectively.

The resultant shiftmap shows lower values of shift in the turbulence-induced regions whereas higher values of shift in the moving object region. Considering this aspect, a threshold T_o is determined as

$$T_o = g_o\sigma_o + K_o\ , \tag{3.14}$$

where g_o is a gain, σ_o is the standard deviation of S, and K_o is an offset. The gain and offset values are chosen as 2 and 3, respectively, in this work.

By applying the threshold T_o from equation (3.13), the third mask M_O is generated as

$$M_O = \begin{cases} 0 & if\ S(x,y) < T_o \\ 1 & if\ S(x,y) \geq T_o \end{cases} \tag{3.15}$$

3.4 Final Mask Generation

Three different masks generated using three different methods are now fused together to generate a more accurate mask that can minimize the distortion of moving object region. Averaging all the masks from equations (3.4), (3.11), and (3.14), the final mask is produced as

$$M_F(x,y) = \frac{M_I(x,y)+M_H(x,y)+M_O(x,y)}{3} \tag{3.16}$$

To soften the edges of the objects and removing unwanted small regions a median filter is applied on the combined mask. The final mask extracts the moving object regions more precisely from the input frames.

3.5 Video Stabilization

The frame-wise objects are then fused with the background to get stable frames and thereby a stabilized video output. This is done in the following way:

1. Each frame and its corresponding final mask are multiplied pixel-wise to produce a frame having only moving object.
2. Complement of the final mask of each frame is multiplied with the background to produce background frames having empty region at the moving object position.
3. For each frame, the object frames generated from Step-1 are now added up with its respective background frames from Step-2 to produce stabilized frame and thus a stabilized video output.

CHAPTER IV

Results and Discussion

4.1 Introduction

The method proposed in this thesis was tested for four different real-world video sequences [29]. These videos contain remote moving objects like car or truck, and are distorted by atmospheric turbulence. Each of the videos contains different number of frames and of different dimensions. Since pixel shiftmap estimation is the most computationally expensive part, it was implemented in MATLAB Executable (MEX) code which reduces the runtime significantly. The results of this proposed approach is compared with that of the approach proposed in [16].

4.2 Mask Generation

In order to generate the mask for obtaining the moving objects, at first, the background frame is estimated from first few input frames through a median operation. Then difference images and pixel shiftmaps are calculated from the input frames and the background frame. After that three different masks are generated by applying three different thresholding techniques. The next step involves combining the above described three masks to achieve minimal distortion.

Figure 4.1 represents the outputs of different stages of mask generation for the proposed video stabilization method. Figures 4.1(i) and 4.1(ii) show a sample turbulence degraded frame and the estimated background frame. It can be noted that the background frame does not contain any moving object. Figures 4.1(iii) and 4.1(iv) show the difference frame and the calculated shiftmap obtained through a pixel registration algorithm. The darker regions in both Figures 4.1(iii) and 4.1(iv) represent lower turbulence-induced motion, whereas the lighter regions indicate either moving object regions or strong turbulence-induced motion. Three masks generated by three different techniques are shown in Figures 4.1(v), 4.1(vi) and 4.1(vii), respectively. In the masks, white pixels indicate the moving object region. Each individual mask contains either few unwanted regions or has some parts missing from the moving object.

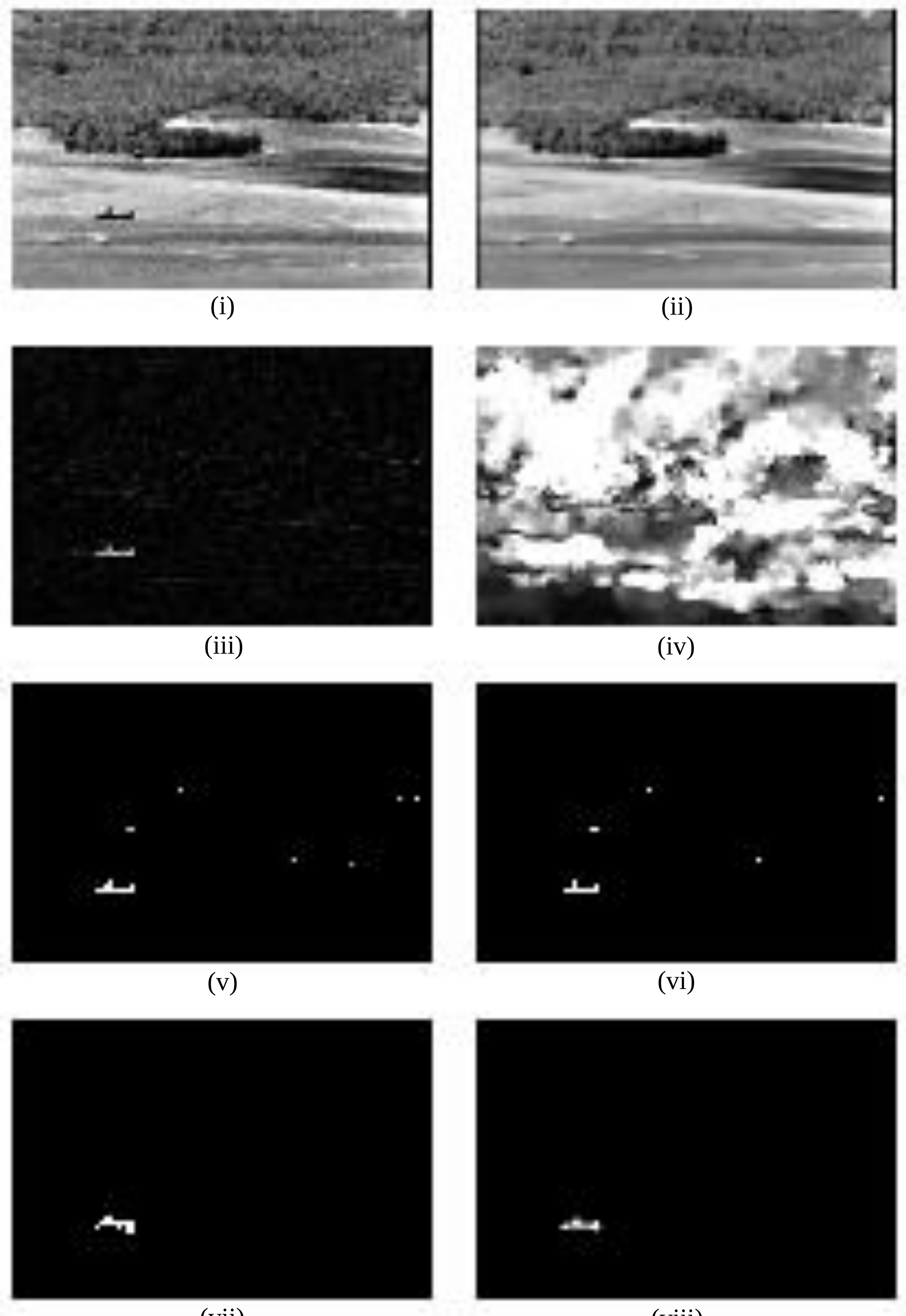


Figure 4.1: (i) A sample input frame, (ii) background frame, (iii) difference image, (iv) pixel shiftmap, (v) mask using intensity (M_I), (vi) mask using histogram (M_H), (vii) mask using optical flow (M_O), and (viii) final mask (M_F).

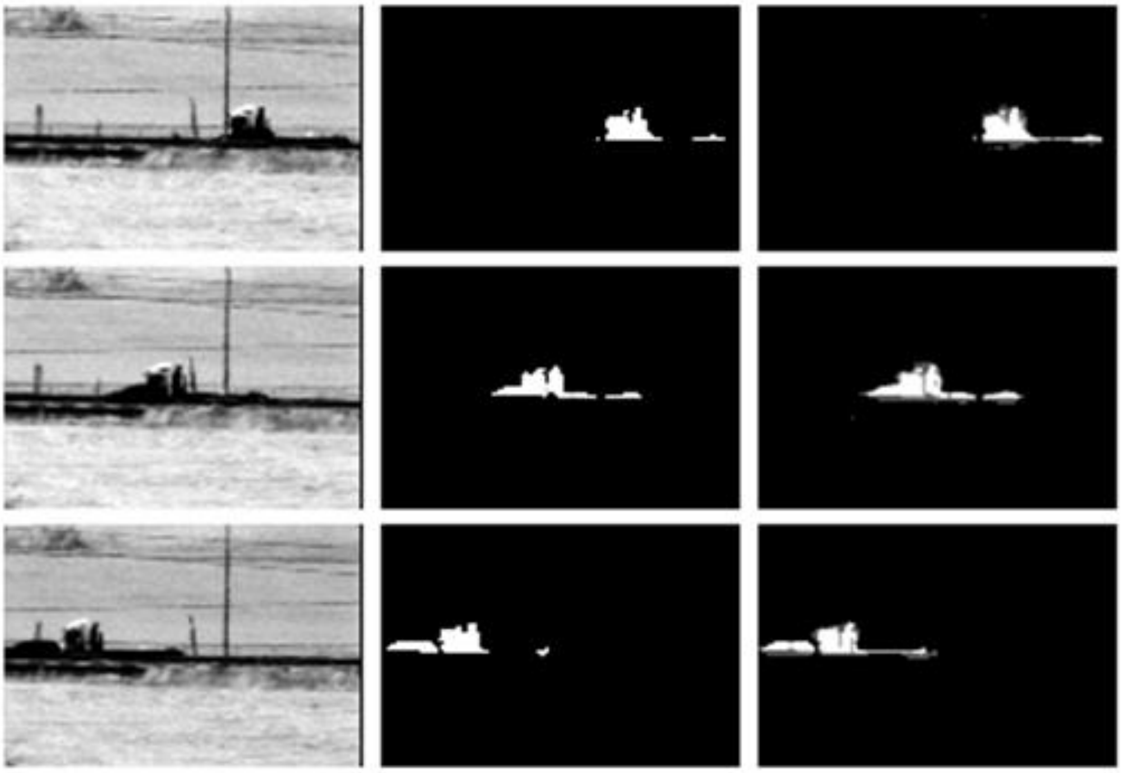
More specifically, the masks M_I and M_H contain several erroneous regions, whereas the mask M_O exhibits truncation of the moving object. For minimizing the distortion effect, these three masks are combined together and it is shown in Figure 4.1(viii). It can be visualized that the final mask gives a better quality mask, where the unwanted regions are filtered out and the lost area is reformed. This is why three different thresholding techniques are used here to generate a binary mask. It helps to reduce the effect of turbulence from the degraded videos, which is a bottleneck when analyzing the distorted video for information gathering.

4.3 Comparison of Proposed Method with Existing Method

The real world videos collected from [29] are degraded due to the turbulence in the medium. These input videos are real world benchmark videos, where the degradation is added naturally. The videos contain different moving cars which are detected in the proposed method with minimal loss of information. The generated final masks of different video sequences are compared with the results presented in an existing method [16]. Besides, the stabilized videos obtained from both the proposed method and the compared method are qualitatively analyzed. A quantitative comparison using three different quality metrics for the four datasets is also presented.

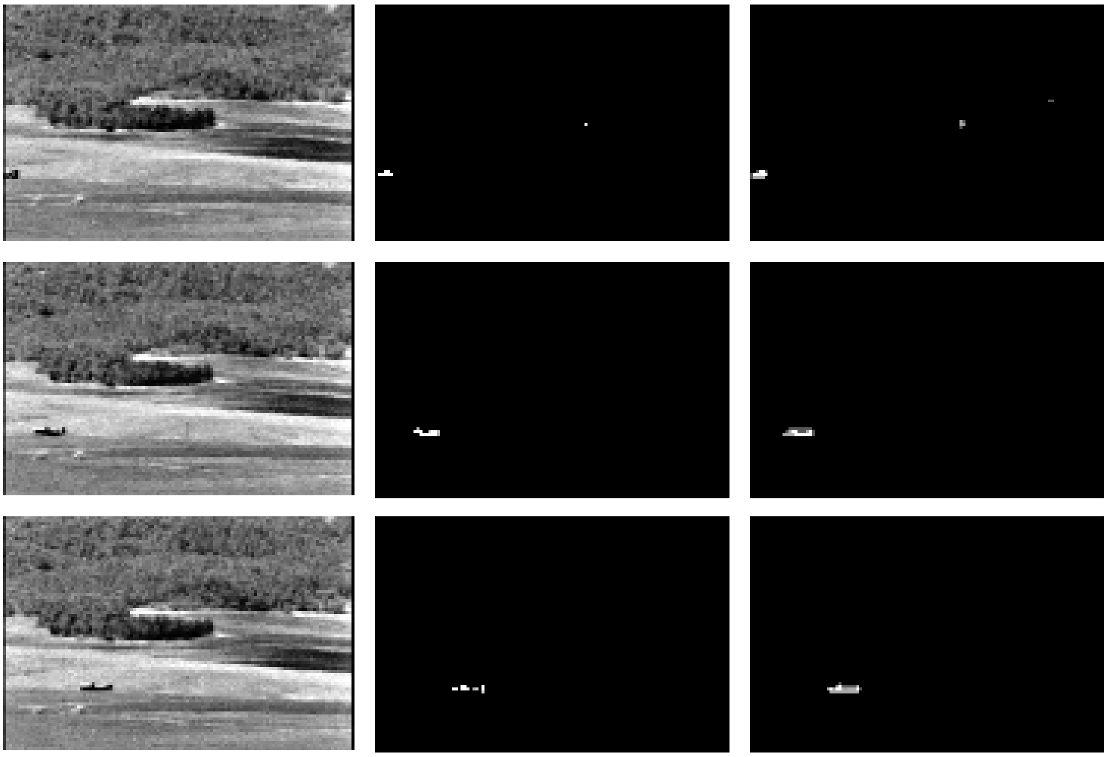
4.3.1 Comparison by generated masks

The final masks generated from the combination of three different masks are compared with those of an existing approach [16]. Three frames from each real world dataset are randomly selected for comparing the results of the proposed approach with the compared approach. Figures 4.2, 4.3, 4.4, and 4.5 show the comparison of final masks for the four datasets. A judicious visual analysis elicits that the proposed method is more accurate in determining the moving objects in almost all cases, except some minor errors.



(i) (ii) (iii)

Figure 4.2: Mask generation result for dataset I: (i) Sample input frames, (ii) generated masks using [16], and (iii) generated masks using the proposed method.



(i) (ii) (iii)

Figure 4.3: Mask generation result for dataset II: (i) Sample input frames, (ii) generated masks using [16], and (iii) generated masks using the proposed method.

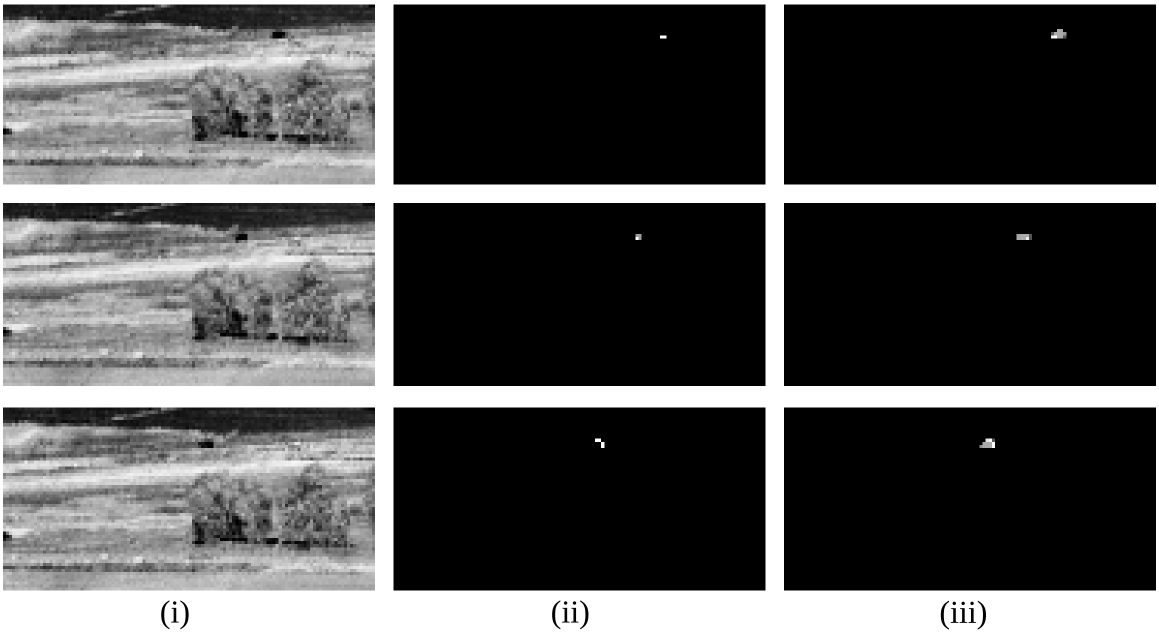


Figure 4.4: Mask generation result for dataset III: (i) Sample input frames, (ii) generated masks using [16], and (iii) generated masks using the proposed method.

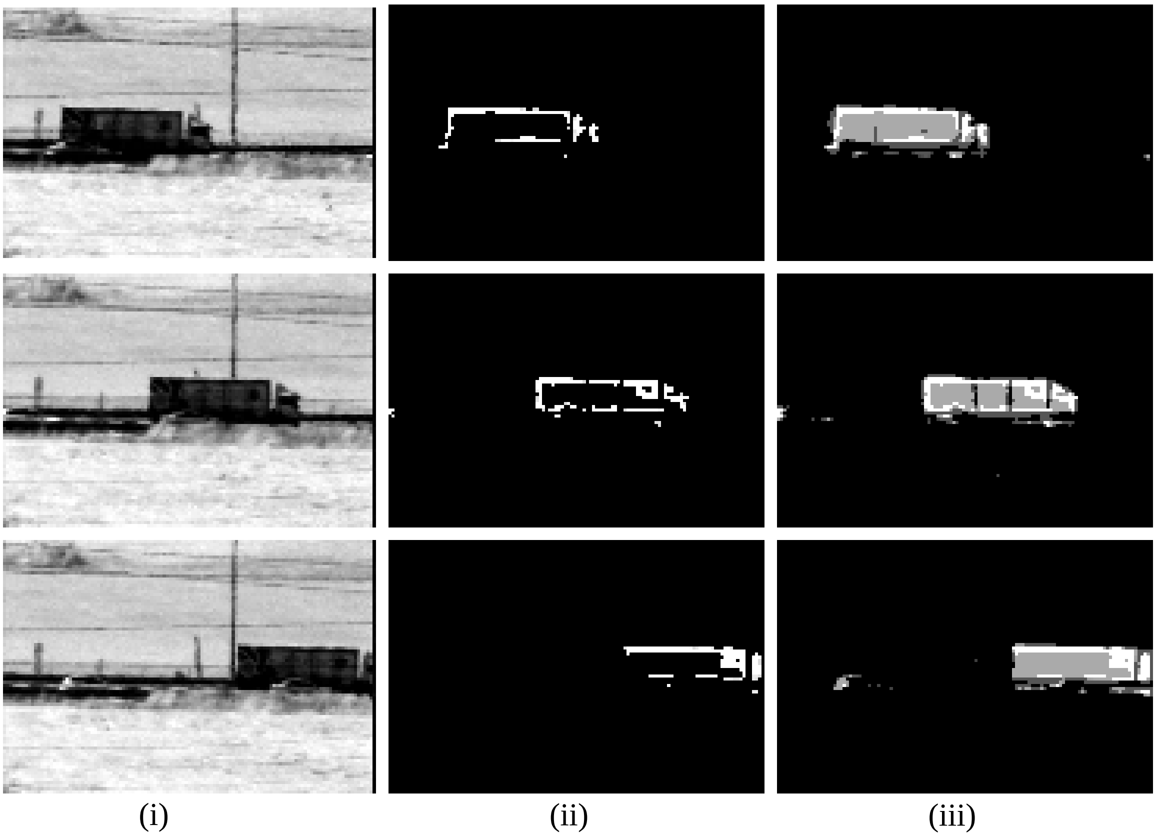


Figure 4.5: Mask generation result for dataset IV: (i) Sample input frames, (ii) generated masks using [16], and (iii) generated masks using the proposed method.

4.3.2 Comparison by stabilized video sequences

The stabilized video sequences are obtained by fusing the background frame with the moving objects extracted from the input frames using the generated masks. Figures 4.6, 4.7, 4.8, and 4.9 show the comparison of three randomly selected frames of the stabilized videos obtained from the proposed method and the compared method. The output frames obtained from the proposed method contain lesser artifacts than the compared method, confirming the effectiveness of the proposed method.

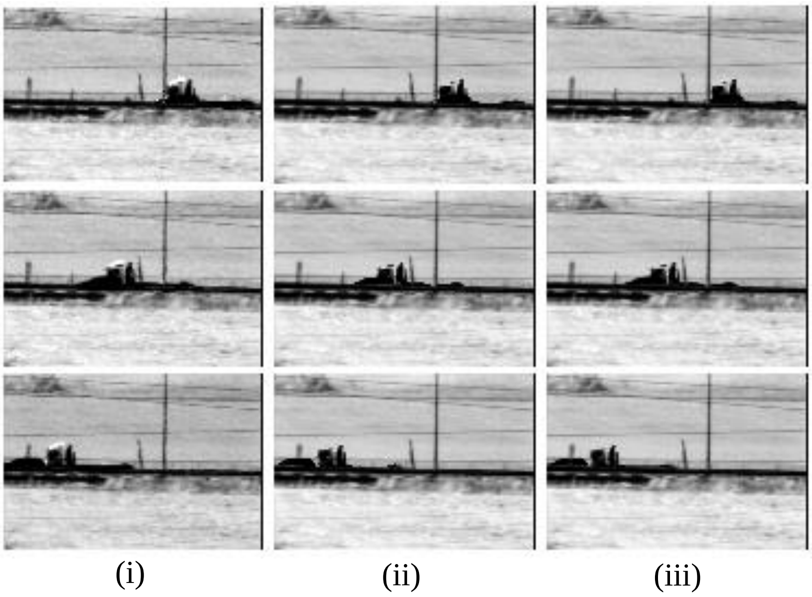


Figure 4.6: Stabilized output for dataset I: (i) Sample input frames, (ii) output frames using [16], and (iii) output frames using the proposed method.

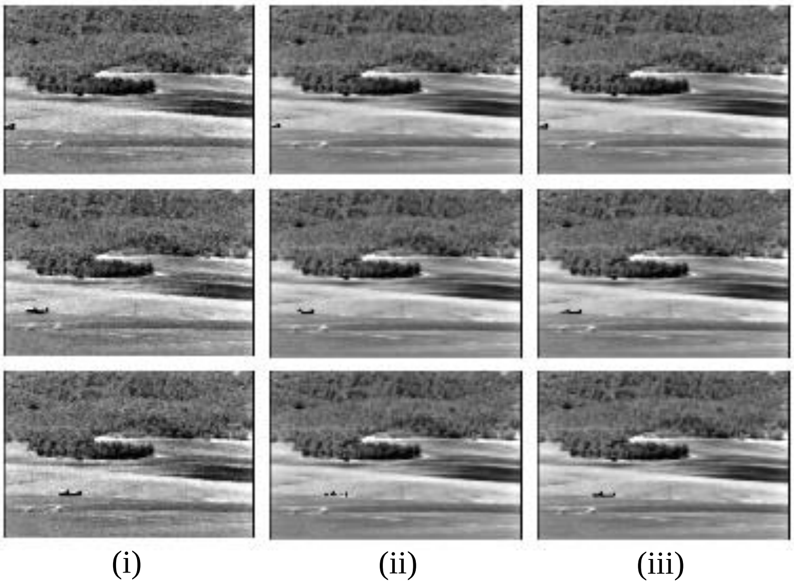


Figure 4.7: Stabilized output for dataset II: (i) Sample input frames, (ii) output frames using [16], and (iii) output frames using the proposed method.

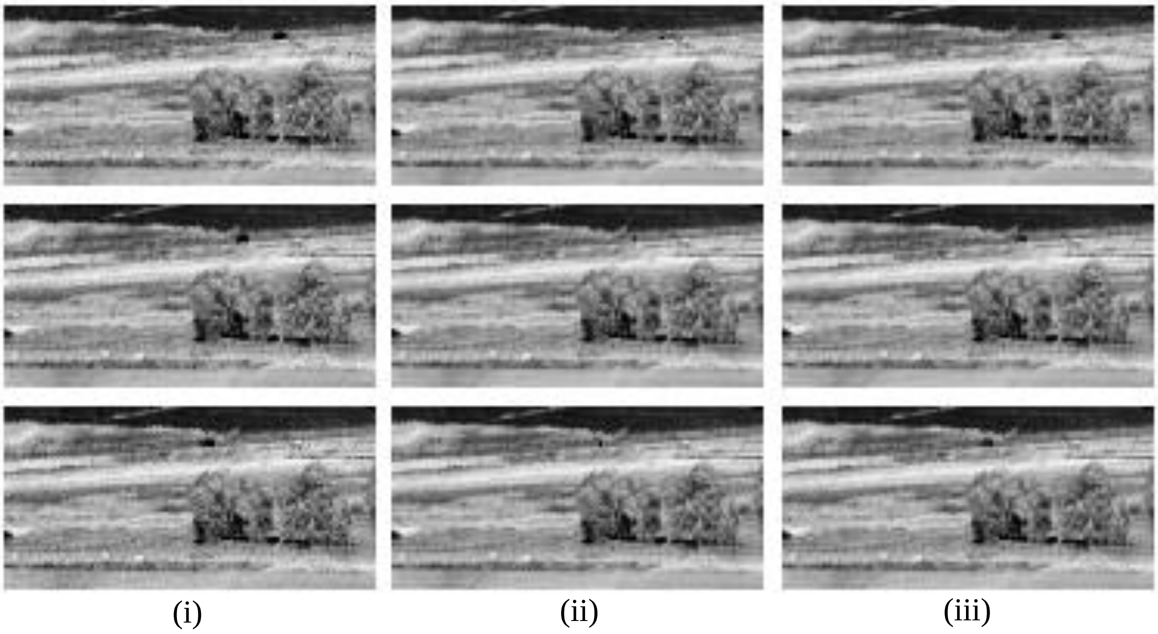


Figure 4.8: Stabilized output for dataset III: (i) Sample input frames, (ii) output frames using [16], and (iii) output frames using the proposed method.

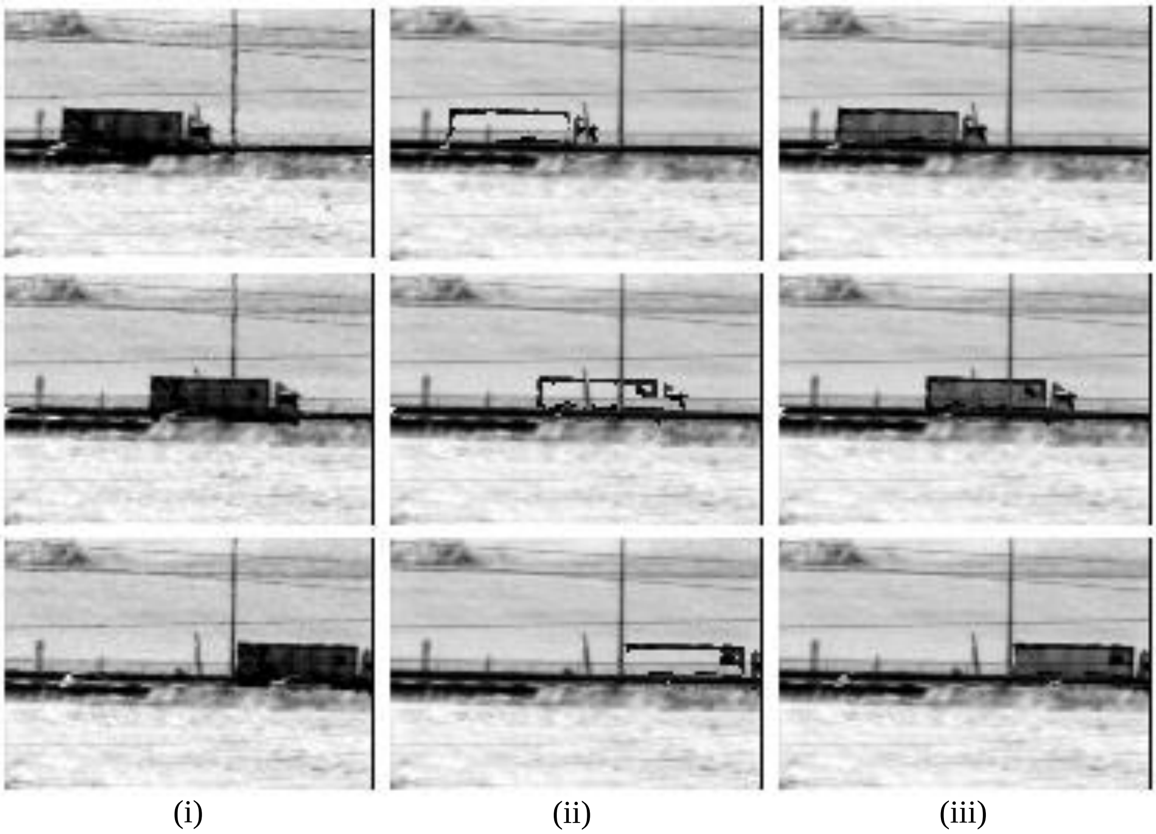


Figure 4.9: Stabilized output for dataset IV: (i) Sample input frames, (ii) output frames using [16], and (iii) output frames using the proposed method.

4.3.3 Comparison by quality metrics

Quality metrics are the measurements of the value and performance of a process. In video stabilization application the moving objects are of smaller size compared to background. Since the quality metrics are dominated by larger background area, foreground correction has very little impact on these measures.

4.3.3.1. Mean-squared error (MSE)

MSE measures the average of squared of the errors. The error is the amount by which the values of the original image differ from the degraded image. The MSE is a measure of the quality of an estimator. It is always non-negative, and values closer to zero are better. It can be defined as

$$\mathrm{MSE} = \frac{1}{\mathrm{M}\times\mathrm{N}} \sum_{\mathrm{x}=1}^{\mathrm{M}} \sum_{\mathrm{y}=1}^{\mathrm{N}} (\mathrm{F_r}(\mathrm{x},\mathrm{y}) - \mathrm{F_t}(\mathrm{x},\mathrm{y}))^2 \,, \qquad (4.1)$$

where $M \times N$ represents the size of frame. F_r and F_t represent the restored frame and the ground-truth frame, respectively.

Since the real world dataset does not have any ground-truth frame, in this work, the MSE is calculated between two successive stabilized frames. Therefore, the MSE plots indicate how well the video sequences are stabilized by the proposed and the compared method. Figures 4.10, 4.11, 4.12, and 4.13 show the comparison of MSE between the proposed method and the compared method for the four datasets.

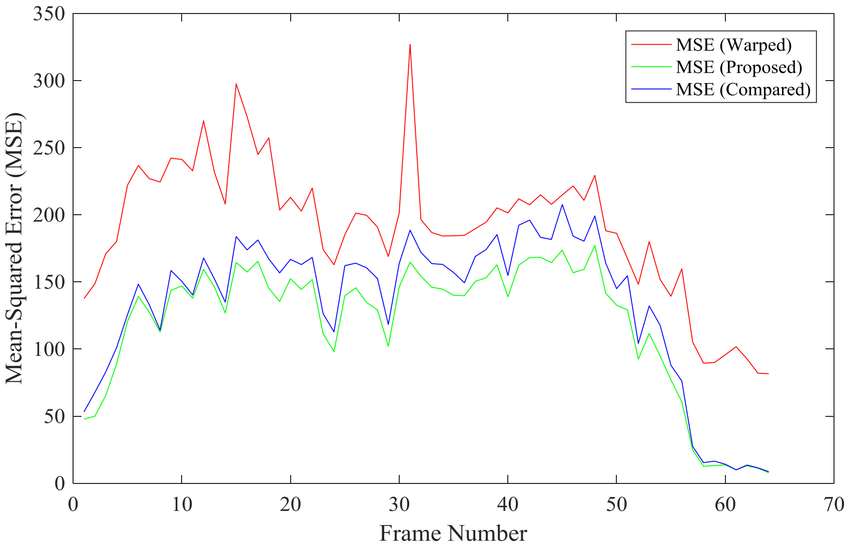


Figure 4.10: MSE plots between two successive frames of the warped sequence and the stabilized sequences using proposed and compared methods for dataset I.

The MSE of the input warped sequence is high as the frames are nonuniformly distorted, i.e., the background is unstable throughout the sequence. A careful observation shows that the MSE curve obtained for the proposed method is consistently lower than that of the compared method for all the datasets. It gives upto 26% improvement in MSE.

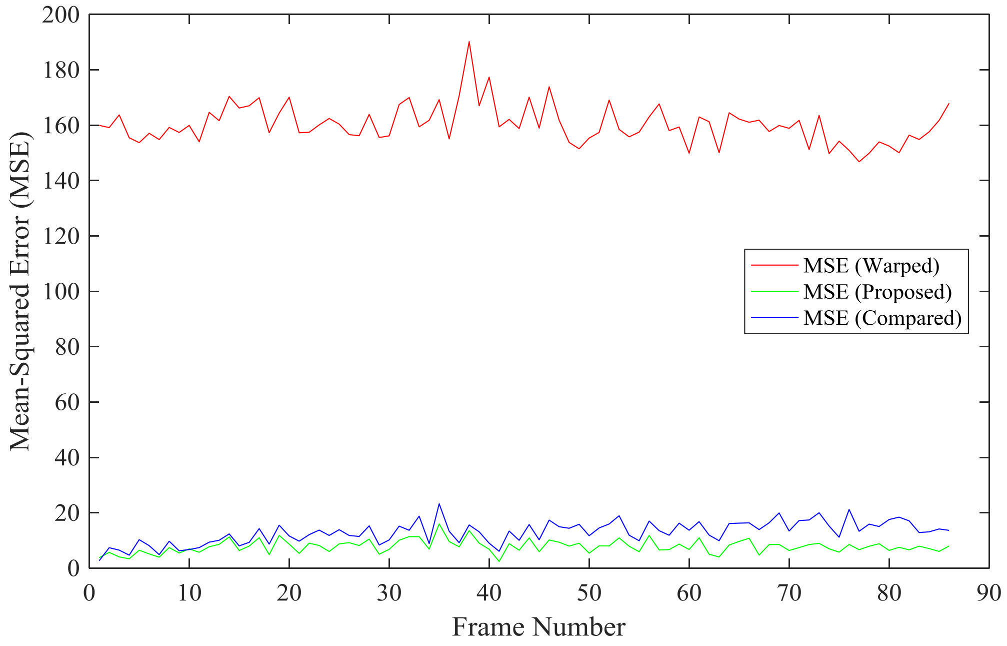


Figure 4.11: MSE plots between two successive frames of the warped sequence and the stabilized sequences using proposed and compared methods for dataset II.

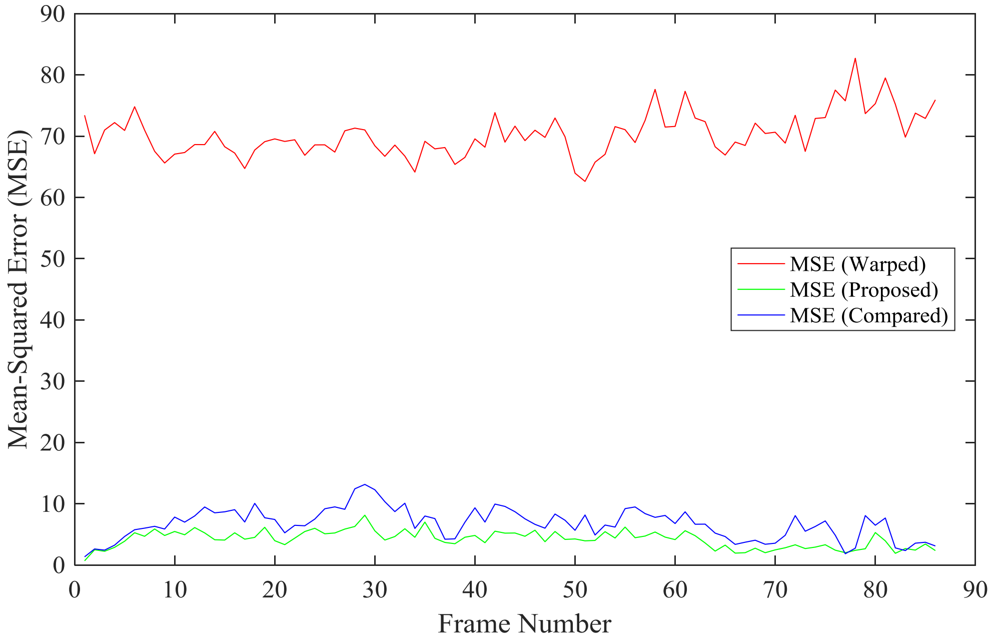


Figure 4.12: MSE plots between two successive frames of the warped sequence and the stabilized sequences using proposed and compared methods for dataset III.

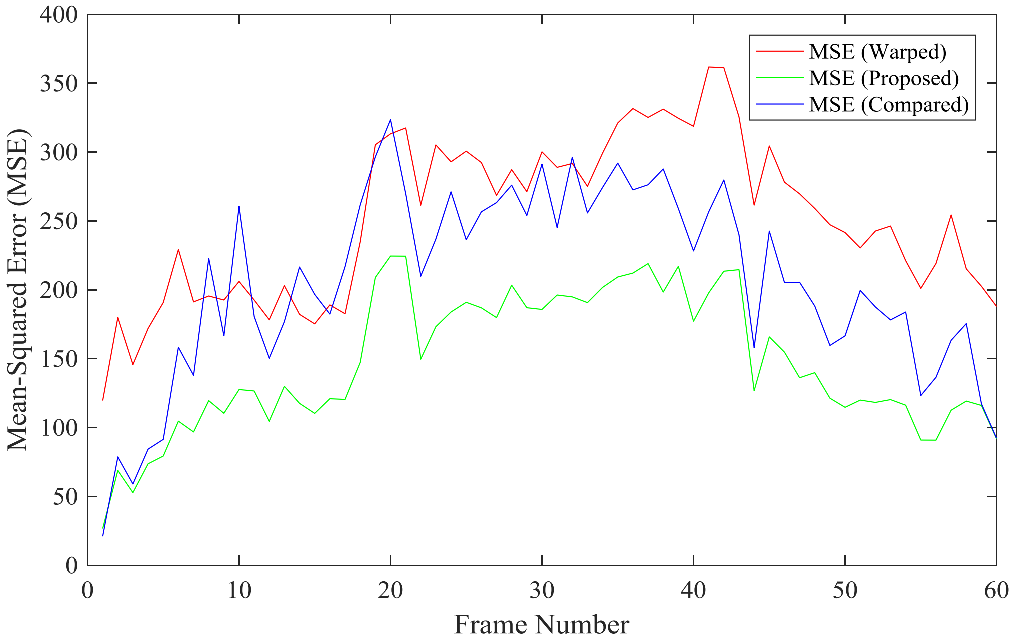


Figure 4.13: MSE plots between two successive frames of the warped sequence and the stabilized sequences using proposed and compared methods for dataset IV.

4.3.3.2. Structural similarity (SSIM)

The SSIM is used for measuring the similarity between two images. The SSIM metric combines local image structure, luminance, and contrast into a single local quality score. In this metric, structures are patterns of pixel intensities, especially among neighboring pixels, after normalizing for luminance and contrast. For two images F and G of common size $M \times N$, SSIM is defined as

$$\text{SSIM}(F,G) = \frac{(2\mu_F\mu_G+c_1)(2\sigma_{FG}+c_2)}{(\mu_F{}^2+\mu_G{}^2+c_1)(\sigma_F{}^2+\sigma_G{}^2+c_2)}, \tag{4.2}$$

where μ_F and μ_G are the average of F and G, respectively; $\sigma_F{}^2$ and $\sigma_G{}^2$ are the variance of F and G, respectively; σ_{FG} is the covariance of F and G; $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$, where L is the dynamic range of the pixel-values (for gray-scale image it is $(2^8 - 1)$), k_1 and k_2 are two offsets.

Figures 4.14, 4.15, 4.16, and 4.17 show the comparison of SSIM between the proposed method and the compared method for the four datasets. Since the input videos are geometrically distorted, the SSIM of the input videos are lower compared to the stabilized ones. For all the tested datasets, the SSIM of the stabilized videos obtained from the proposed method is higher than that of the compared method.

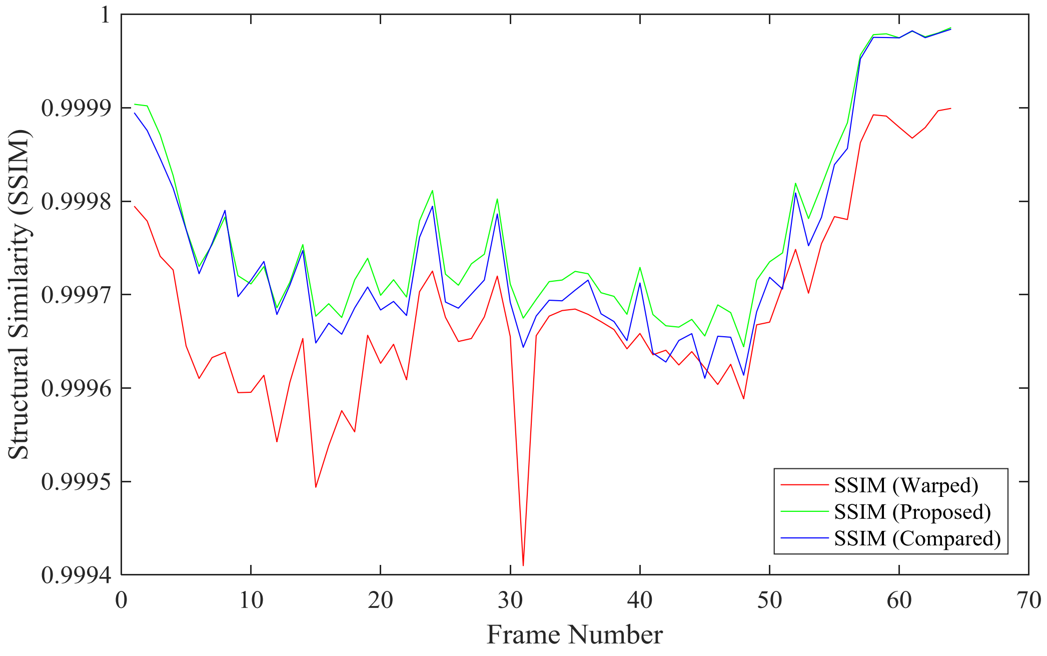


Figure 4.14: SSIM plots between two successive frames of the warped sequence and the stabilized sequences using proposed and compared methods for dataset I.

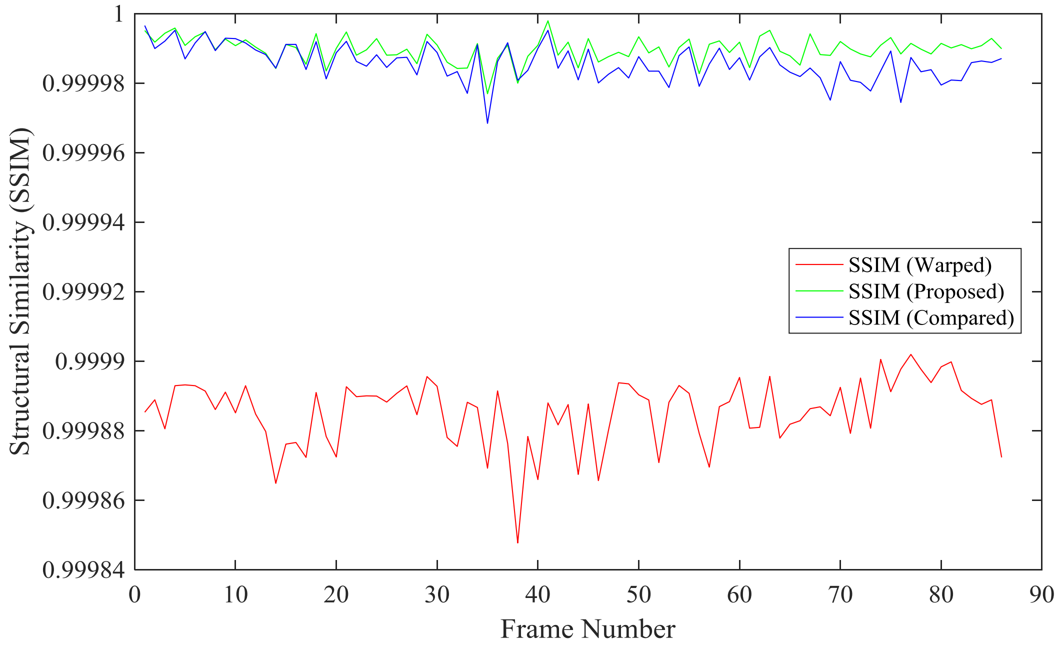


Figure 4.15: SSIM plots between two successive frames of the warped sequence and the stabilized sequences using proposed and compared methods for dataset II.

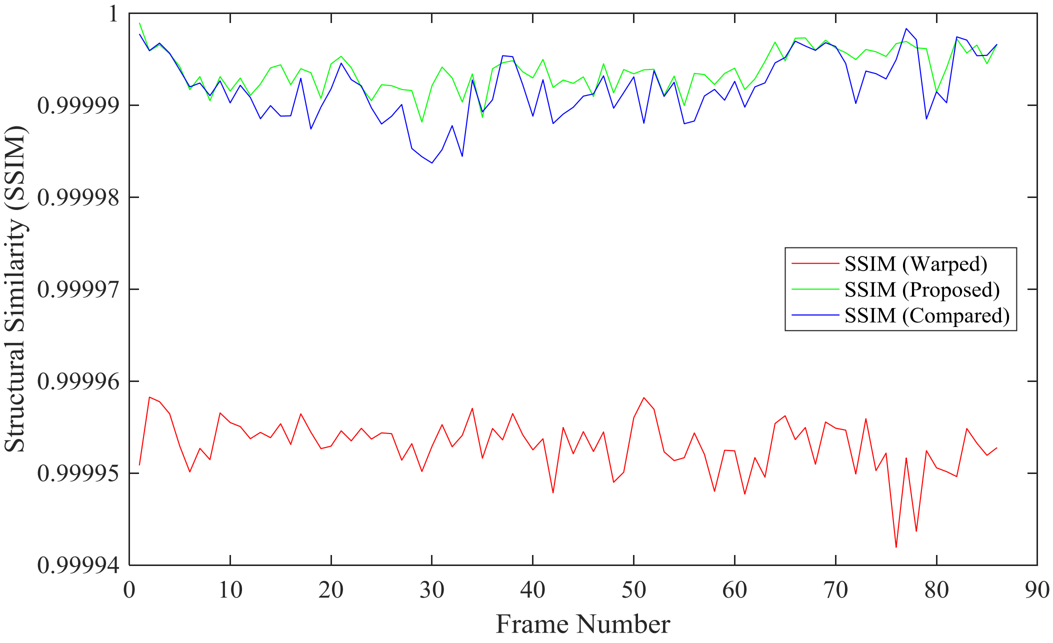


Figure 4.16: SSIM plots between two successive frames of the warped sequence and the stabilized sequences using proposed and compared methods for dataset III.

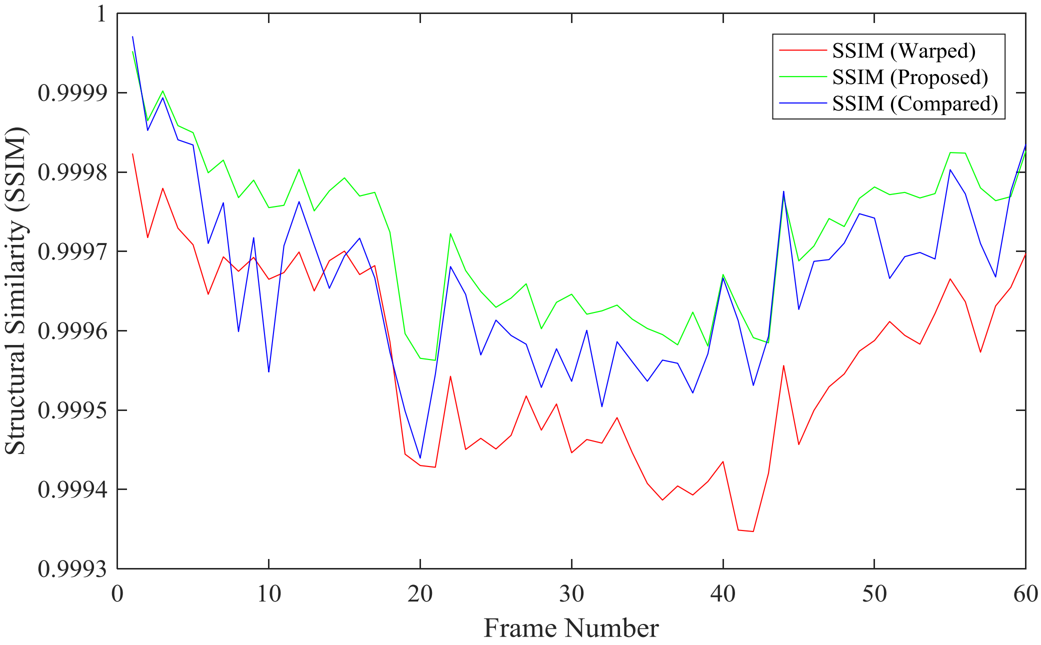


Figure 4.17: SSIM plots between two successive frames of the warped sequence and the stabilized sequences using proposed and compared methods for dataset IV.

4.3.3.3 Peak signal-to-noise ratio (PSNR)

The ratio between the maximum possible power of a signal and the power of corrupting noise is known as PSNR. Since many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale. The higher value of PSNR indicates that the degraded image has been reconstructed better to match the original image and also shows the better efficiency of reconstructive algorithm. This would occur because it is desired to minimize the MSE between images with respect the maximum signal value of the image. PSNR is most easily defined as

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\mathrm{MAX}\big(F_r(x,y)\big)^2}{\frac{1}{M\times N}\sum_{x=1}^{M}\sum_{y=1}^{N}\big(F_r(x,y)-F_t(x,y)\big)^2}\right) \tag{4.3}$$

Figures 4.18, 4.19, 4.20, and 4.21 show the comparison of PSNR between the proposed method and the compared method for the four datasets. Similar to SSIM metric, the input video sequences exhibit lower value of PSNR compared to the stabilized ones. For all datasets, the proposed method provides higher values of PSNR compared to the existing method.

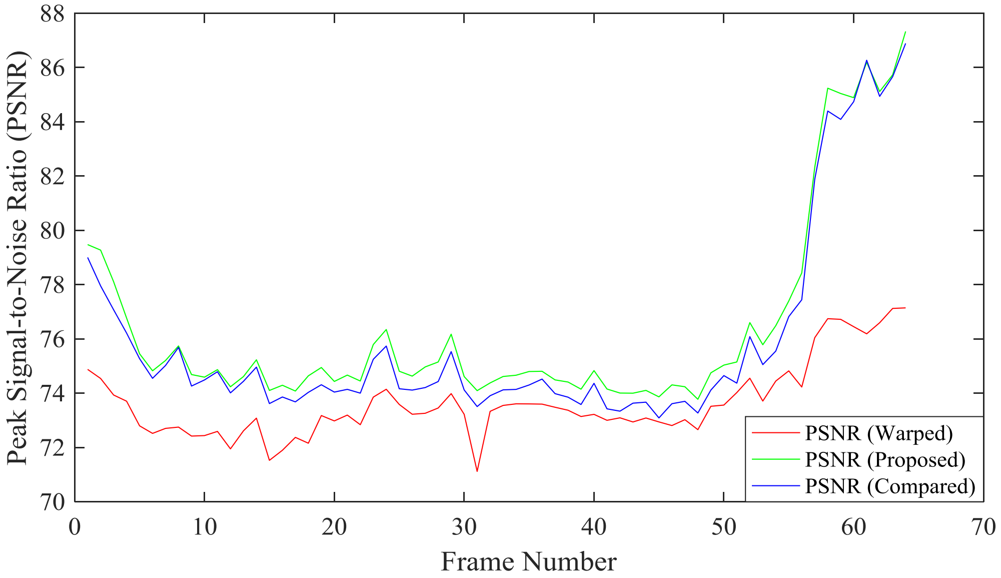


Figure 4.18: PSNR plots between two successive frames of the warped sequence and the stabilized sequences using proposed and compared methods for dataset I.

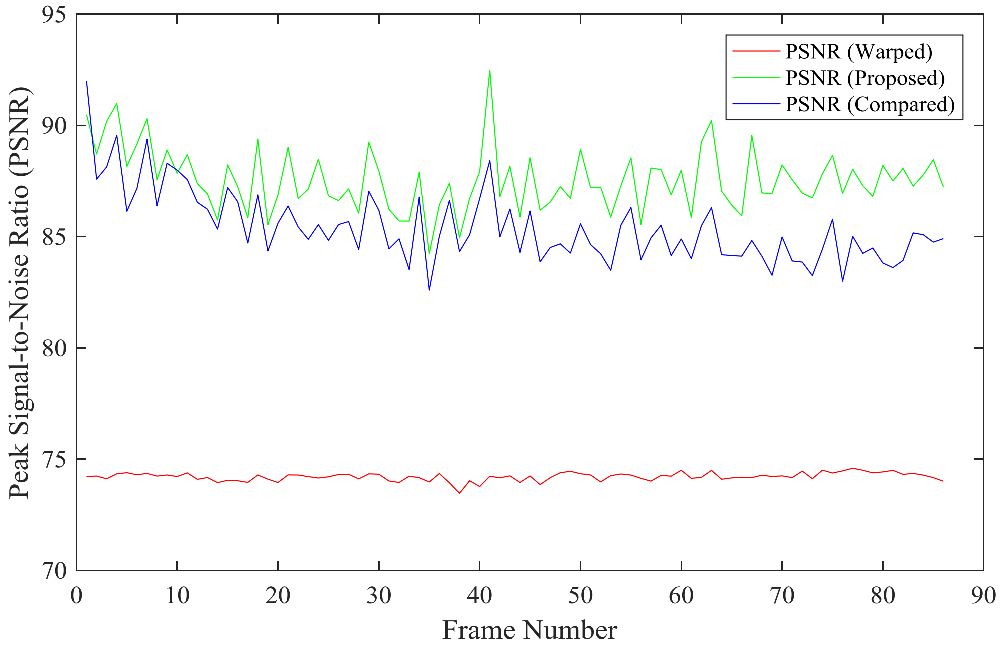


Figure 4.19: PSNR plots between two successive frames of the warped sequence and the stabilized sequences using proposed and compared methods for dataset II.

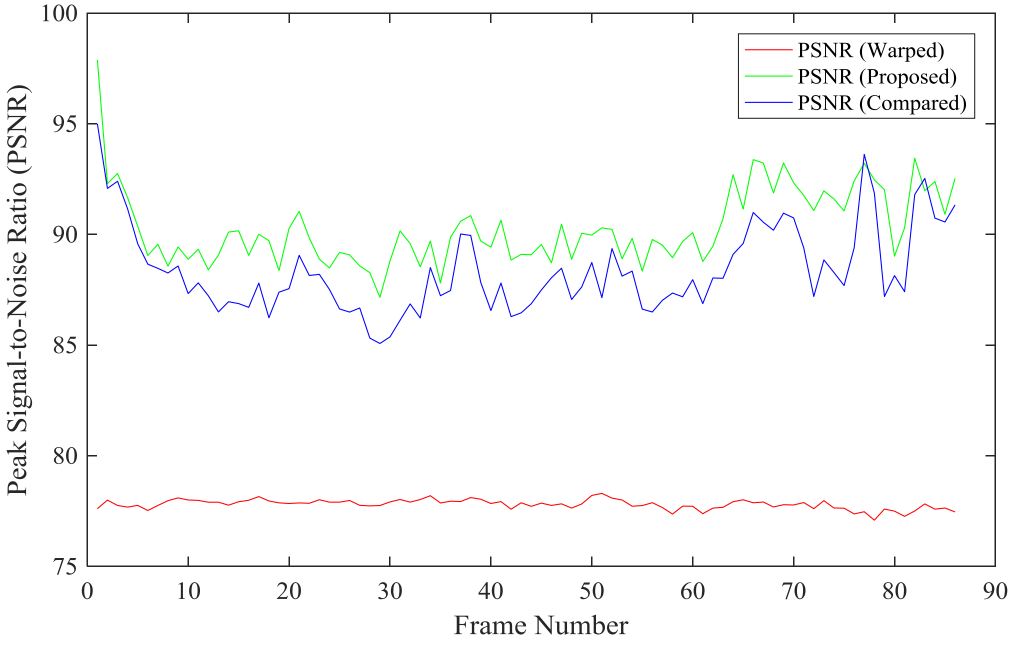


Figure 4.20: PSNR plots between two successive frames of the warped sequence and the stabilized sequences using proposed and compared methods for dataset III.

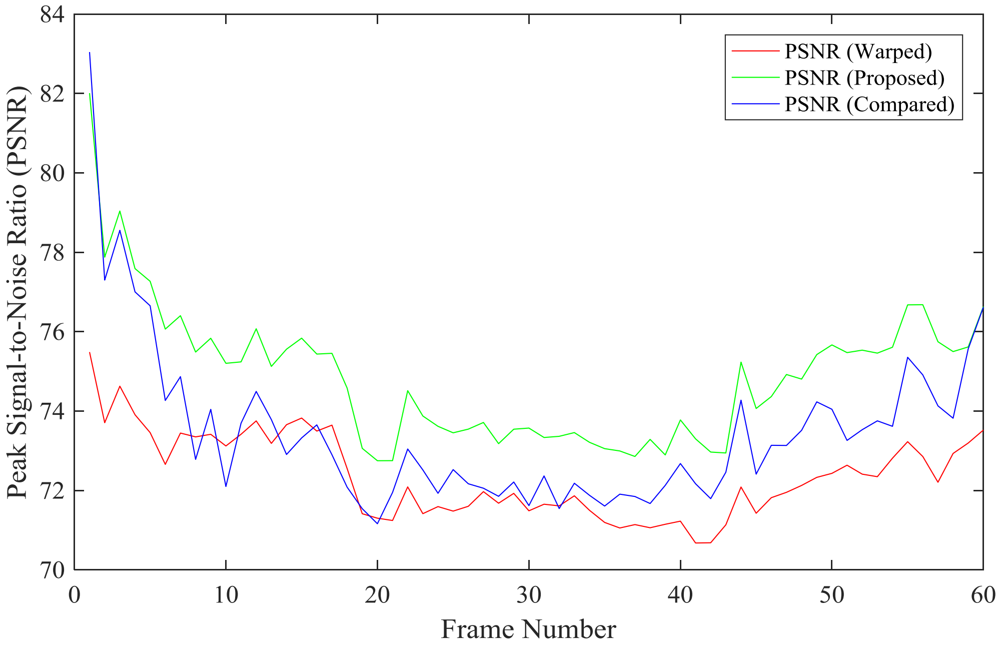


Figure 4.21: PSNR plots between two successive frames of the warped sequence and the stabilized sequences using proposed and compared methods for dataset IV.

4.3.3.4 Overall Comparison

The overall comparison of the quality metrics is shown in Table 4.1. The proposed method shows better result for every tested real-world dataset.

Table 4.1: Overall comparison of the proposed method with the compared method [16].

Parameter / Dataset	MSE		PSNR		SSIM	
	Proposed	Compared	Proposed	Compared	Proposed	Compared
I	110.8952	124.5267	77.35908	74.12666	0.999782	0.999765
II	7.772483	12.89212	87.56701	85.41107	0.99999	0.999986
III	4.206228	6.712453	90.30489	88.3615	0.999994	0.999992
IV	113.652	153.1284	76.63569	75.91616	0.999785	0.99975

The proposed method is capable of minimizing the loss of information from the input videos and can recover the missing regions of moving objects which is the main challenge in the detection of moving objects from turbulence degraded videos. From both qualitative and quantitative analyses, it is noted that the proposed method outperforms an existing method, therefore, it is a greater contribution to the moving object detection and turbulence degraded video stabilization problems.

CHAPTER V

Conclusion and Recommendations

5.1 Conclusion

In this thesis work, an efficient technique has been presented for identifying moving objects and stabilizing turbulence degraded videos. Distinguishing turbulence and moving objects was the main challenge in this work. The proposed method can be subdivided into four parts: (1) generation of a geometrically stable background frame, (2) generation of three masks using difference image, histogram, and optical flow of the individual frames, (3) obtaining a refined mask from the three different masks, and (4) stabilizing the geometrically distorted frames. The intuition behind the use of three different masks to generate a final one is that it could provide minimal distortion and smoothen the area of the moving object. For different real-world video sequences, the result of comparison between the proposed method and an existing method ensures that the proposed method is more accurate in moving object detection and video stabilization which adds up an attainment in the field of image and video processing in computer vision.

The proposed method is intended to be used in surveillance, where the objects, such as cars, tanks and people are moving and the background scene is stationary. This method is simpler to implement and has less computational complexity. If this method is used for online video stabilization then the background may be updated after every five frames. The real-world videos generally have 30 frames/second. Thus the delay is about (1/6)th of a second which is acceptable and no visual difference occurs. Research studies on human perception of images indicate that this delay is not of adverse impact for routine tasks.

5.2 Recommendations for Future Works

The presented work in this thesis can be extended to improve the performance of similar problems in underwater imaging. In this thesis work, the most computationally complex part is image registration. Thus the future work may concentrate on this to reduce both the complexity and registration time.

Object tracking in surveillance system is an interesting research area. Developing a moving object tracking system using Artificial Neural Network (ANN) could also be a future research topic.

REFERENCES

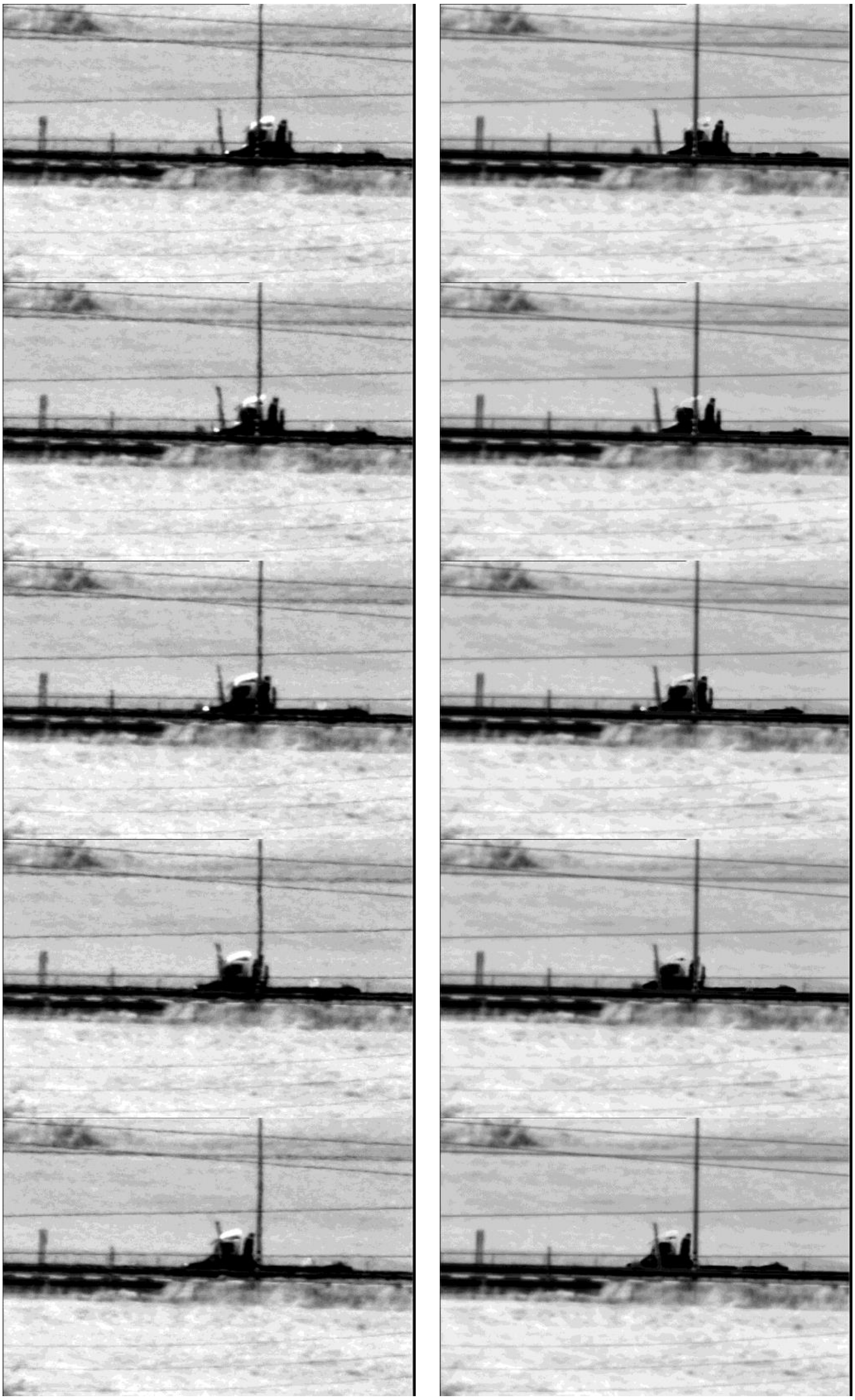
1. M. A. Alharbi, “Fast video stabilization algorithm,” M.Sc. dissertation, Air Force Institute of Technology, Ohio, 2006.
2. I. Overington, “Vision and acquisition: Fundamentals of human visual performance, environmental influences, and applications in instrumental optics,” London: Pentech Press, New York: Crane, Russak & Company, Inc., 1976.
3. M. Tahtali, “Imaging Techniques through the atmosphere,” Ph.D. dissertation, The University of New South Wales, Australia, 2007.
4. K. K. Halder, “High precision techniques for imaging through turbulence,” Ph.D. dissertation, The University of New South Wales, Australia, 2016.
5. N. Anantrasirichai, A. Achim, N. G. Kingsbury, and D. R. Bull, “Atmospheric turbulence mitigation using complex wavelet-based fusion,” IEEE Transactions on Image Processing, vol. 22, no. 6, pp. 2398-2408, 2013.
6. K. K. Halder, M. Tahtali, and S. G. Anavatti, “Simple and efficient approach for restoration of non-uniformly warped images,” Applied Optics, vol. 53, no. 25, pp. 5576–5584, 2014.
7. J. Guo, C. Hsia, Y. Liu, M. Shih, C. Chang, and J. Wu, “Fast background subtraction based on a multilayer codebook model for moving object detection,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 10, pp. 1809-1821, 2013.
8. B. Fishbain, L. P. Yaroslavsky, and I. A. Ideses, “Real-time stabilization of long range observation system turbulent video,” J. Real-Time Image Proc., vol. 2, no. 1, pp. 11–22, 2007.
9. O. Oreifej, X. Li, and M. Shah, “Simultaneous video stabilization and moving object detection in turbulence,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 2, pp. 450-462, 2013.
10. J. Gilles, F. Alvarez, N. Ferrante, M. Fortman, L. Tahir, A. Tarter, and A. von Seeger, “Detection of moving objects through turbulent media. Decomposition of oscillatory vs non-oscillatory spatio-temporal vector fields,” Image and Vision Computing, vol. 73, pp. 40–55, 2018

11. C. Zhang, F. Zhou, B. Xue, and W. Xue, “Stabilization of atmospheric turbulence-distorted video containing moving objects using the monogenic signal,” Signal Processing: Image Communication, vol. 63, pp. 19–29, 2018.
12. X. Zhu and P. Milanfar, “Stabilizing and deblurring atmospheric turbulence,” IEEE International Conference on Computational Photography (ICCP), Pittsburgh, PA, pp. 1-8, 2011.
13. Y. Mao and J. Gilles, “Non rigid geometric distortions correction - Application to atmospheric turbulence stabilization,” Inverse Problems & Imaging, vol. 6, no. 3, pp. 531-546, 2012.
14. Y. Lou, S. H. Kang, S. Soatto, and A. L. Bertozzi, “Video stabilization of atmospheric turbulence distortion,” Inverse Problems and Imaging, vol. 7, no. 3, pp. 839-861, 2013.
15. R. P. J. Nieuwenhuizen, A. W. M. van Eekeren, J. Dijk, and K. Schutte, “Dynamic turbulence mitigation with large moving objects,” Electro-Optical and Infrared Systems: Technology and Applications XIV, vol. 10433, 2017.
16. K. K. Halder, M. Tahtali, and S. G. Anavatti, “Geometric correction of atmospheric turbulence-degraded video containing moving objects,” Optics Express, vol. 23, no. 4, pp. 5091–5101, 2015.
17. I. B. Islam, M. T. E. Elahi, and K. K. Halder, “An efficient mask generation method for moving object detection in atmospheric imaging,” International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, pp. 1-5, 2019.
18. M. T. E. Elahi, and K. K. Halder, “Detecting moving objects from long-range atmospheric turbulence degraded videos,” International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), Dhaka, Bangladesh, pp. 260-263, 2018.
19. K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: Principles and practice of background maintenance,” IEEE International Conference on Computer Vision (ICCV), Corfu, Greece, vol. 1, 1999.
20. M. Sezgin and B. Sankur, “Survey over image thresholding techniques and quantitative performance evaluation,” Journal of Electronic Imaging, vol. 13, pp. 146-165, 2004.
21. D. P. Reddy and K. Rao, “An efficient approach for fingerprint identification through finger images,” International Journal of Research Studies in Science, Engineering and Technology, vol. 1, no. 8, pp. 71-84, 2014.

22. N. A. Pham, A. Morrison, J. Schwock, S. A. Ronen, V. Iakovlev, M. S. Tsao, and D. W. Hedley, “Quantitative image analysis of immunohistochemical stains using a CMYK color model,” Diagnostic Pathology, vol. 2, pp. 1-10, 2007.
23. Y. Wu, “Optical flow and motion analysis,” Advanced Computer Vision Notes Series, Northwestern University, Evanston.
24. B. K. P. Horn and B. G. Schunck, “Determining optical flow,” Artificial intelligence laboratory, Massachusetts Institute of Technology, Cambridge, pp. 185-203, USA, 1980.
25. B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” Proceedings of Imaging Understanding Workshop, pp. 121-130, 1981.
26. D. Li, “Restoration of atmospheric turbulence degraded video using kurtosis minimization and motion compensation,” Ph.D. dissertation, Georgia Institute of Technology, Georgia, 2007.
27. N. Otsu, “A threshold selection method from gray-level histograms,” IEEE Transactions on System, Man, and Cybernetics, vol. 9, no. 1, pp. 62-66, 1979.
28. C. Liu, “Beyond pixels: Exploring new representations and applications for motion analysis,” Ph.D. dissertation, Massachusetts Institute of Technology, USA, 2009.
29. N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, “Changedetection.net: A new change detection benchmark dataset,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, pp. 1-8. 2012.

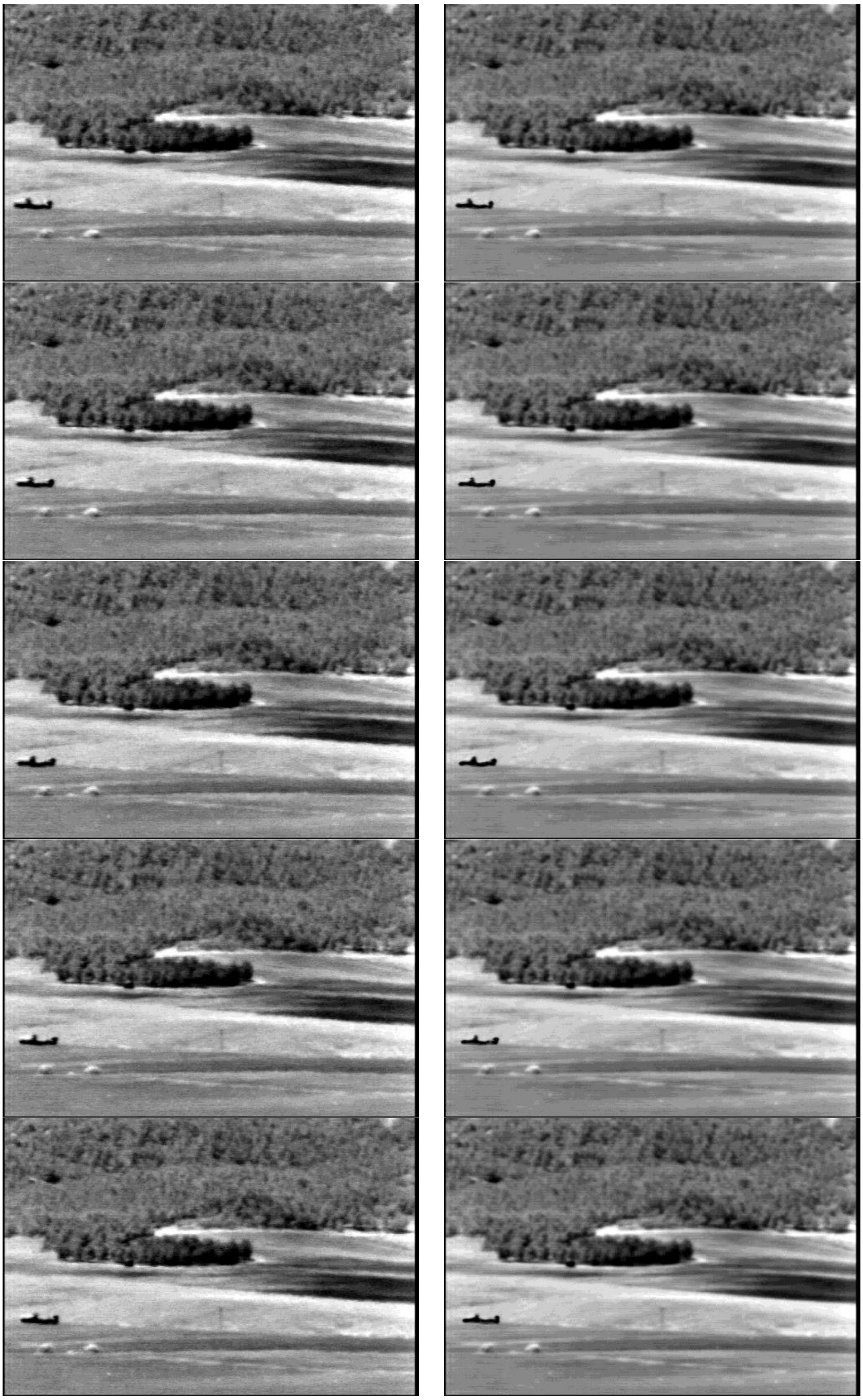
APPENDIX

1. Sample input and output video frames for dataset I:

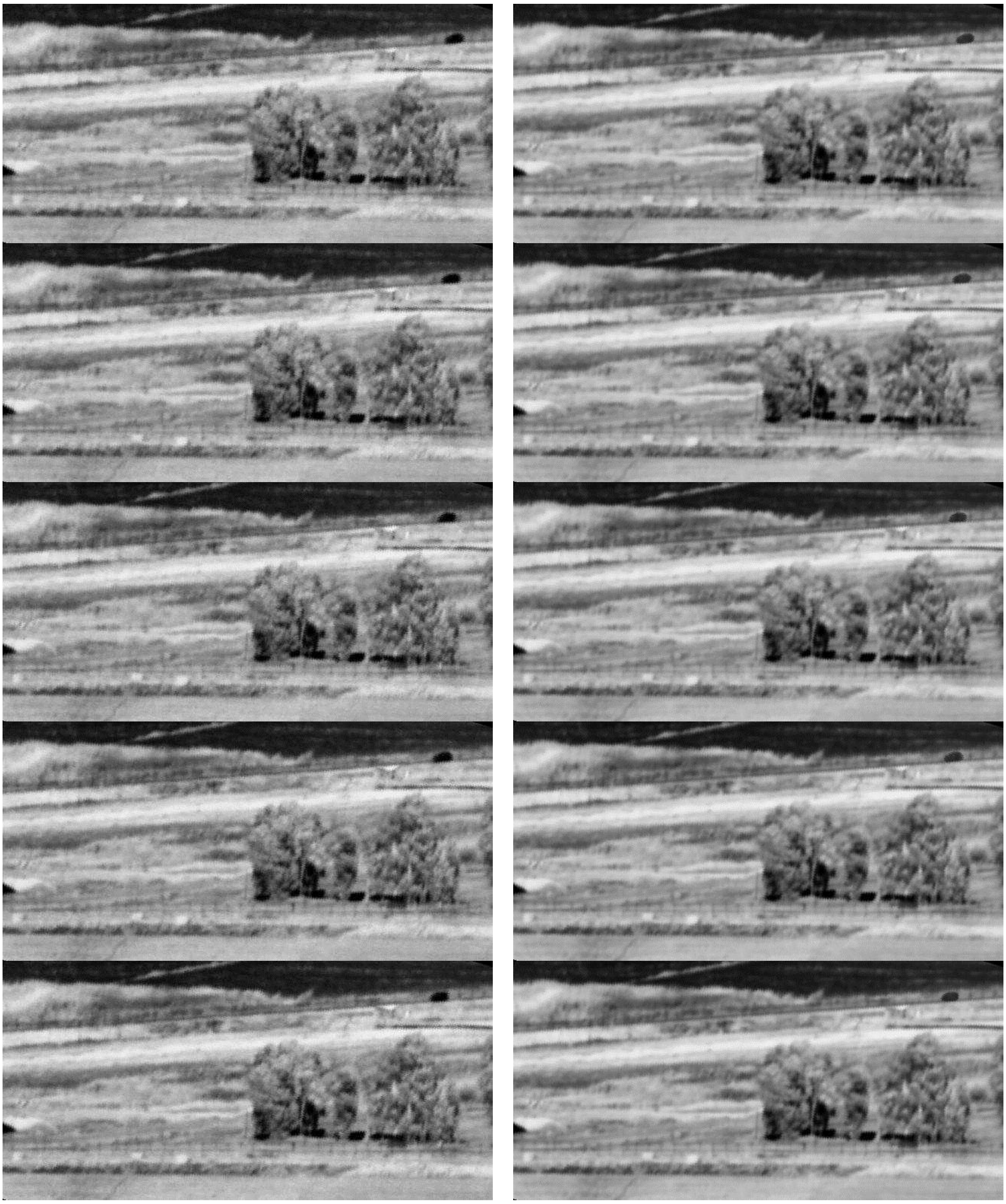


Input Output

2. Sample input and output video frames for dataset II:



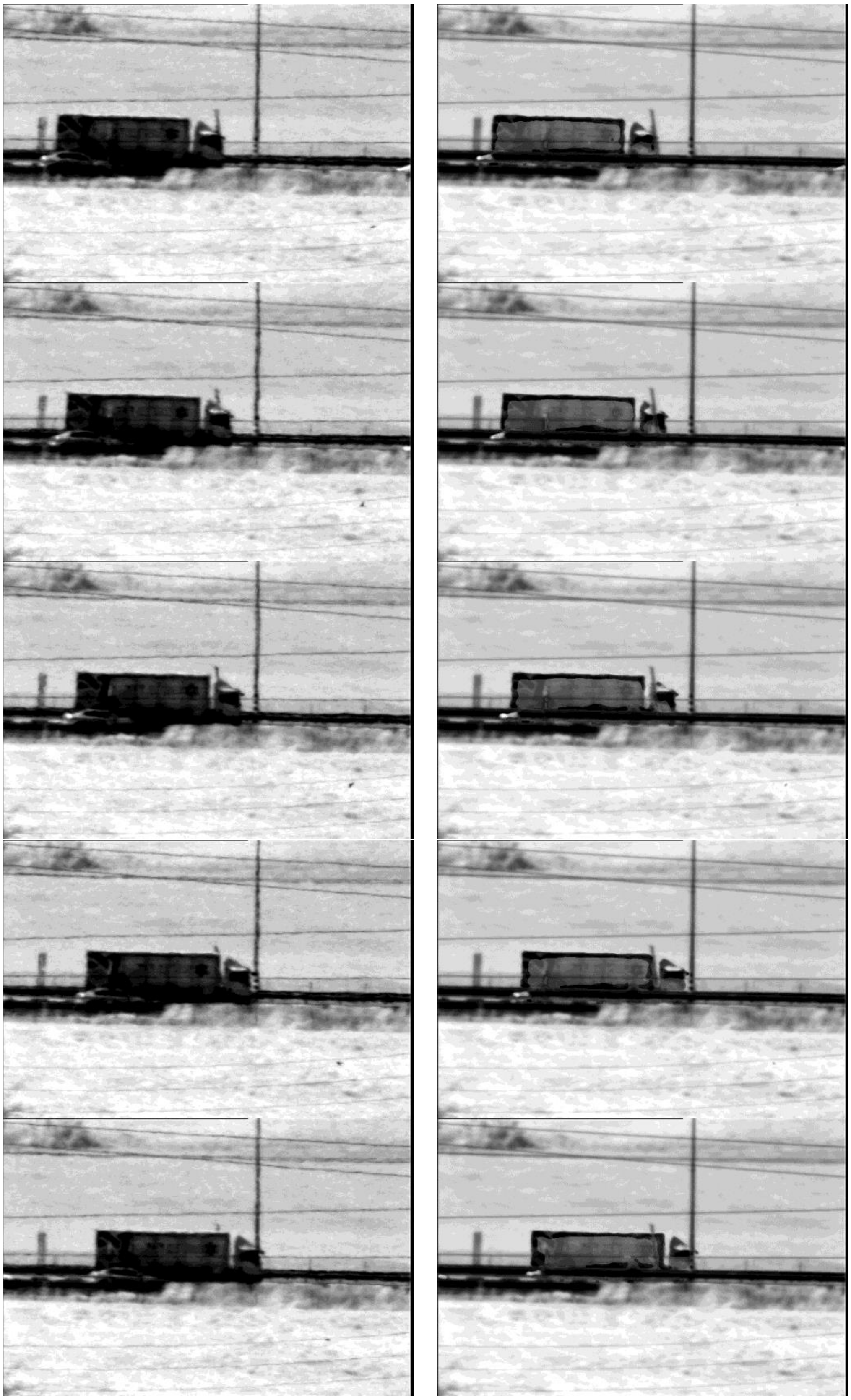
3. Sample input and output video frames for dataset III:



Input

Output

4. Sample input and output video frames for dataset IV:



Input

Output