The Color Ellipsoid Framework for Imaging in the Atmosphere

Permalink
https://escholarship.org/uc/item/4g75056r

Author
Gibson, Kristofor B.

Publication Date
2013-01-01

Peer reviewed|Thesis/dissertation
UNIVERSITY OF CALIFORNIA, SAN DIEGO

The Color Ellipsoid Framework for Imaging in the Atmosphere

A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy in

Electrical Engineering (Signal and Image Processing)

by

Kristofer B. Gibson

Committee in charge:

Professor Truong Nguyen, Chair
Professor Serge Belongie
Professor Pamela Cosman
Professor William Hodgkiss
Professor David Kriegman

2013
The dissertation of Kristofor B. Gibson is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

________________________

________________________

________________________

________________________

________________________

Chair

University of California, San Diego

2013
DEDICATION

To my Mamaw.
In his heart a man plans his course,  
but the Lord determines his steps.  
—Proverbs 16:9 (NIV)
# TABLE OF CONTENTS

Signature Page ........................................ iii
Dedication ........................................ iv
Epigraph ........................................ v
Table of Contents ................................... vi
List of Figures ....................................... ix
List of Tables ....................................... xiii
Acknowledgements .................................... xiv
Vita ................................................... xvi
Abstract of the Dissertation ........................ xvii

## Chapter 1
Introduction .......................................... 1

## Chapter 2
Pre/Post Contrast Enhancements ........................ 3
  2.1 Background in Contrast Enhancements ............. 4
  2.2 Fast Single Image Dehazing ....................... 6
  2.3 Dehazing Effects on Coding Artifacts ............ 11
    2.3.1 Ringing Artifacts .......................... 13
    2.3.2 Blocking Artifacts ........................ 15
    2.3.3 JPEG Pre/Post Experiment .................. 17
  2.4 Dehazing Effect on Motion Estimation ............. 21
    2.4.1 H.264 Pre/Post Experiments ............... 24
  2.5 Acknowledgment .................................. 29

## Chapter 3
Color Ellipsoid Framework ............................ 31
  3.1 Background ..................................... 32
  3.2 Atmospheric dichromatic model .................... 33
  3.3 Color ellipsoid framework without fog ........... 35
  3.4 Color ellipsoid framework with fog ............... 37
    3.4.1 General properties ........................ 37
    3.4.2 Color ellipsoid model with depth discontinuity 39
  3.5 Proposed ellipsoid prior method .................. 42
  3.6 Unification of single image defogging methods .......... 45
    3.6.1 Dark channel prior ........................ 45
    3.6.2 Fattal prior ................................ 48
LIST OF FIGURES

Figure 2.1: Dichromatic model. ........................................... 5

Figure 2.2: Examples of occlusion boundaries. (a) Original image. (b) De-
hazed using unrefined DCP method (2.9). (c) Dehazed using
our proposed MDCP method (2.12). (d) Dehazed using method
from [1]. Notice in (b) and (d) there are halo artifacts around
the leaf edges. The halo artifacts are not as dominant in (c). 11

Figure 2.3: Block diagram for JPEG compression. .................. 12

Figure 2.4: Comparison of dehazed image using Pre and Post methods with
similar file sizes. .................................................. 18

Figure 2.5: Real world example of Pre and Post using MDCP (2.12) and
JPEG compression. (a) Original image. (b) Post method with
MDCP. (c) Pre method with MDCP. (d) Detected block arti-
facts in Post image using [2]. (e) Detected block artifacts in Pre
image using [2]. .................................................. 18

Figure 2.6: Measure of blocking artifacts for each method at different image
sizes. ........................................................................ 19

Figure 2.7: A performance comparison of the deblocked Pre and Post meth-
ods using JPEG and WILD database. ............................ 19

Figure 2.8: A frame from the original video sequence is in (a). A snapshot
from the dehazed video is in (b). The red box indicates the
region used for closer comparisons. Subfigures (c)-(e) are the
comparisons of the Pre and Post methods with H.264 compres-
sion at 250 and 500 kbps bitrates. ............................. 25

Figure 2.9: Luminance PSNR vs. Bitrate using H.264. ............. 26

Figure 2.10: Measure of blocking artifacts for each method and bitrate. The
metric was applied to every frame in a video sequence and then
averaged. ................................................................. 27

Figure 2.11: Plot of ITU-R averaged marks with error bars vs. Rank. . . . 30

Figure 3.1: Color ellipsoids in clear natural scene. (a) Clear day scene with
a sample window over the tree trunk (white rectangle) and a
window over the road (black rectangle). (b) RGB histogram
plot of points from the two windows with their densities pro-
jected on each color plane. (c) The same RGB histogram plot
only with the ellipsoid approximations for each region of inter-
est (ROI). The green ellipsoid is from the tree trunk, and the
red ellipsoid is from the dirt road. ............................... 36

Figure 3.2: Color ellipsoids in foggy natural scene. (a to c) Images of the
tree branch in the fog at three different distances with sample
windows overlaid on the same branch in each image. (d) RGB
histogram of each sample set. ..................................... 39
Figure 3.3: Example of color ellipsoid with depth discontinuity. (a) Foggy image with ROI centered at a depth discontinuity. (b) RGB histogram of the ROI. Note the presence of two density mixtures.

Figure 3.4: Example of the halo effect. (a) Original image. (b) Enhanced image using centroid prior with simple averaging. (c) Enhancement of $\bar{\mathbf{x}}$ using centroid prior with median operator. Note the halo around the tree in the foreground in (b) is not present in (c).

Figure 3.5: Example of single image defogging methods. (a) Original house image from [3], $\bar{\mathbf{x}}$. (b to f) Defogged images using Lemma 2, He et al. [4], Fattal [3], Tarel and Hautière [1], and Gibson and Nguyen [5], respectively. (g to k) Transmission estimates used for the above defogged images.

Figure 3.6: DCP and color ellipsoid relationship. Graphical example of the relationship between the DCP and the minimum distance to three different minimum volume ellipsoids on the red-green plane.

Figure 3.7: Geometric interpretation of Equation 3.46. The figure contains two ellipsoids $\mathcal{E}_1$ and $\mathcal{E}_2$ with centroids $\mathbf{\bar{\mu}}_1$ and $\mathbf{\bar{\mu}}_2$, respectively. The projection of the centroids onto the airlight vector $\mathbf{a}$ are the same for both ellipsoids.

Figure 3.8: Tarel prior and color ellipsoid relationship. Graphical example of the relationship between the image whiteness and color ellipsoids on the red-green plane. Examples of ellipsoid positions and orientations that are well approximated with measuring the minimum color component of the respective centroid are the clusters with dark ellipses. The dashed blue ellipse is an example of a cluster orientation where the approximation is not valid.

Figure 3.9: Additional examples of single image defogging methods. (a) Original house image from [3], $\bar{\mathbf{x}}$. (b to f) Defogged images using Lemma 2, He et al. [4], Fattal [3], Tarel and Hautière [1], and Gibson and Nguyen [5], respectively. (g to k) Transmission estimates used for the above defogged images. Similarly, (l) is a original foggy image of pumpkins [3]. (m to q) Defog results. (r to v) Transmission estimates.

Figure 4.1: Comparison of images with conflicting $AGN$ and MOS values. An average gradient norm $AGN$ measure from Eq. (4.1) and MOS is associated for each image in (a)-(c). A high $AGN$ or MOS value means high contrast. Note that the MOS for image (b) is higher than the MOS for image (a) and likewise for their $AGN$ values. However the MOS for image (c) is higher than (b) but the $AGN$ value for image (b) is higher than (c).
Figure 4.2: Subjective Data. (a) Sorted MOS values with confidence intervals. (b) Histogram of all MOS values.

Figure 4.3: Example of DCP. Two $320 \times 240$ size images are measured with the DCP with window size $|\Omega| = 15 \times 15$, (a) and (c) are the original and enhanced images, respectively. The DCP applied to (a) and (c) are in (b) and (d), respectively. The cumulative distribution function (CDF) of (b) and (d) are in (e). Note that the CDF of the DCP of the enhanced image has a higher CDF at lower pixel intensities (below 20% of maximum intensity).

Figure 4.4: Example of sub-metric $M_5$ which uses the color ellipsoid centroids. (a) Geometric interpretation of the vector displacements illustrated on the Red-Green color plane without loss of generality. (b) $M_5$ applied to a foggy image without averaging. With averaging, $M_5 = 0.32$. (c) $M_5$ applied to enhanced image without averaging. With averaging, $M_5 = 0.90$.

Figure 4.5: ROC curves for $CEM_p$, $CEM_h$, $CEM_c$, and $CEM_{hc}$. (a) The test and training ROC curves for $CEM_p$ and $CEM_{hc}$. (b) A comparison of test curves with all methods. The Area Under the Curve (AUC) values are included.

Figure 4.6: Block size for $M_6$ (4.26) vs. Kendall rank correlation performance for $CEM_{hc}$.

Figure 4.7: The images (a)-(i) are ranked by the $CEM_{hc}$ from lowest to highest contrast, respectively. Images (a), (d) and (h) are of the same scene with decreasing levels of fog, respectively. Images (b) and (f) are from TID2008 database [6]. Images (c) and (g) are from [7]. Images (e) and (i) are from [3].

Figure 5.1: Simulation of object tracking in low contrast noisy images. (a) The first two images from left to right, $o_1$ and $o_2$, respectively, are the images used for measuring PSR with various $t$ and $\sigma_w$ values. The last two (from left to right) are an example of the images with $t = 0.4$, $\sigma_w = 0.1$, and $a = 0.3$. (b) PSR of translation estimation with cross correlation method with varying $t$ and $\sigma_w$ values.

Figure 5.2: Simulation of object tracking in low contrast noisy images and noise reduced images. The dashed curves are PSR values from object tracking with images containing noise and contrast reduction. The solid curves are PSR values from object tracking with images with low contrast but with noise reduced using a locally adaptive Wiener filter. Each curve pair is associated with the same noise variance (before noise reduction).
Figure 5.3: 32 × 32 grid control point examples used to simulate hot-air turbulence by image warping. (a) Original grid points. (b) Warping with $\sigma_{\Delta}^2 = 0.5$ and $L = 8$. (c) Warping with $\sigma_{\Delta}^2 = 1$ and $L = 8$. (d) Warping with $\sigma_{\Delta}^2 = 0.5$ and $L = 4$. (e) Warping with $\sigma_{\Delta}^2 = 0.5$ and $L = 2$. (f)-(j) Siemens star chart images associated with the grid warpings from the first row.

Figure 5.4: PSNR values from turbulence simulation experiment using (5.51) with two different methods: Defog Before (5.56) and Defog After (5.57) frame averaging. (a) With fixed $\sigma^2_w = 0.1$ and $L = 8$. (b) With $\sigma^2_\Delta = 1$ and $\sigma^2_w = 0.1$. (c) With $\sigma^2_\Delta = 0.5$ and $L = 8$.

Figure 5.5: Zoomed in view of images from simulation using (5.51) with parameters set to $\sigma^2_\Delta = 2$, $L = 64$, $\sigma^2_w = 0.05$, $a = 0.8$ and $K = 30$ frames. (a) Original image $o(x)$. (b) Single image from experiment with $t(x) = 0.2$. (c) “Defog After” method applied to image sequence with $t(x) = 0.2$. (d) ”Defog Before” method applied to image sequence with $t(x) = 0.2$. (e) Single image from experiment with $t(x) = 0.2o(x)$. (f) “Defog After” method applied to image sequence with $t(x) = 0.2o(x)$. (g) “Defog Before” method applied to image sequence with $t(x) = 0.2o(x)$.

Figure 5.6: Design of proposed method.

Figure 5.7: Example of Wiener Defogging method. (a) Foggy image $i(x)$. (b) Defogged image $o(x)$ using the 1-step Wiener Defogging method [8]. (c) The transmission map estimate $\hat{t}$ (normalized for display purposes). The sample window size $|\Omega|$ = 14 was estimated with (5.64).

Figure 5.8: Comparison of averaging with global $K$ and spatially variant $K(x)$. (a) Single image from sequence with with fog, noise and turbulence. (b) Image result from averaging $K = 30$ frames. (c) Image result frame averaging with $K(x)$ frames in (5.70). (d) PSNR comparison with $\sigma^2 = 1e^{-3}$. (e) PSNR comparison with $\sigma^2_w = 0.01$.

Figure 5.9: Performance measurement of the CETM with simulated turbulent/foggy sequence in Fig. 5.8. (The turbulence variance $\sigma_{\Delta} = 0.5$ and scale $L = 64$. (a) The TMM measurements for various $K$ values. (b) The PSNR values for each $K$ values.) (c) The PSNR performance of the CETM at different stages of the algorithm.

Figure 5.10: CETM applied to real image sequences. Images (a) - (d) are frame captures from the image sequences where the original sequence is on top and the result from applying the CETM method is on the bottom. Images (e) and (f) are from Zhu and Milanfar’s proposed method in [9]. (a) Boat image sequence. (b) Star image sequence. (c) Water Tower. (d) Moon.
### LIST OF TABLES

Table 2.1: Comparison of average motion vector and residual bits per frame. 28  
Table 2.2: SCACJ Results .................................. 29  
Table 3.1: Summary of dark prior methods .......................... 57  
Table 4.1: Sub-Metric Errors .................................... 84  
Table 5.1: Performance metrics for CETM ............................ 128
ACKNOWLEDGEMENTS

I want to give thanks to God for sustaining me and guiding me through this arduous journey.

I want to thank my C.R.A.N.K crew (Can, Ramsin, Ankit, Natan) for the camaraderie, tips to survival, and encouragement especially during the times it felt like I had nothing but rejected papers and wanted to give up.

I thank SSC Pacific for providing the programs necessary for me to pursue a degree while solving difficult problems for the war fighter.

I also want to thank Prof. Nguyen for taking the risk of bringing me into the research group and for everything he has done to keep me going and to finish strong.

Many thanks to Professors Belongie and Kriegman for also taking a risk by writing letters in my support to be re-admitted into the graduate program. I would not have made this accomplishment without them.

I want to thank my brother for introducing me to higher level thinking at such a young age from theory of relativity to C programming. I want to thank my uncle Ralph for buying me my first of many electronics kits that fostered my curiosities. I thank my dad for getting that Commodore VIC-20 working for me so I can learn programming, for his hard work to allow me to get my degree at Purdue and for always encouraging me to think and be creative. I thank my mom for her love and support and for listening when the times were tough and I needed someone to talk to.

And finally, my thanks goes to my lovely and remarkable wife, Sarah. She has supported me steadfastly and unconditionally while I was tucked away in a lab writing software and papers while she wrangled 3 wild and rambunctious kids in the wee hours in the morning and late at night. I am forever in debt for her support and encouragement to pursue my dreams.

Chapter 2, in part, is a reprint of the material as it appears in IEEE Transactions on Image Processing 2012. Kristofor Gibson, Dũng Vô, and Truong Nguyen, IEEE, 2012. The dissertation author was the primary investigator and author of this paper.
Chapter 3, in part, is a reprint of the material as it appears in EURASIP Journal on Image and Video Processing 2013. Kristofor Gibson and Truong Nguyen; EURASIP, 2013. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in part, is a reprint of the material as it appears in IEEE Transactions on Image Processing 2013. Gibson, Kristofor Gibson and Truong Nguyen; IEEE, 2013. The dissertation author was the primary investigator and author of this paper.

Chapter 5, in part is currently being prepared for submission for publication of the material. Kristofor Gibson and Truong Nguyen. The dissertation author was the primary investigator and author of this material.
VITA

2001 B. S. in Electrical and Computer Engineering, Purdue University, West Lafayette, IN

2001-2013 Electrical Engineer at SSC Pacific

2008-2009 M. S. in Electrical Engineering (Signal and Image Processing), University of California, San Diego

2009-2013 Ph. D. in Electrical Engineering (Signal and Image Processing), University of California, San Diego

PUBLICATIONS


Within the past decade, there has been a growing interest in the development of surveillance systems deployed in the maritime domain. Surveillance in the maritime domain is confronted with many problems that are not easily solved. Most problems are caused by the weather, such as fog, haze, and turbulence that degrade the contrast and image quality of images and video. Most recently, researchers have proposed methods to remove fog in images fast enough for real-time processing. However, they are not unified in their approach. Additionally, there exists no metric that indicates the perceptual quality of an image based on the contrast. In this work, we discuss the problems encountered when including contrast enhancements for fog removal along with image and video compression.
We unify existing fog removal methods with our proposed Color Ellipsoid Framework and present a new fog removal method. We then utilize the Color Ellipsoid Framework to improve the performance of a no-reference perceptual based contrast enhancement metric. Over the years many researchers have provided insight into the physics of either the fog or turbulence but not both. In this work, we provide an analysis and method that incorporates both physics models: fog and turbulence. We observe how contrast enhancements (fog removal) can affect object tracking and frame averaging. We present in this work a new joint contrast enhancement and turbulence mitigation method (CETM) that utilizes estimations from the contrast enhancement algorithm to improve the turbulence removal algorithm. We provide two new turbulent mitigation metrics (based on the Color Ellipsoid Framework) that measures temporal consistency. And finally, we design the CETM to be efficient such that it can operate in fractions of a second for near real-time applications.
Chapter 1

Introduction

The official seal for the University of California includes a book and the words *Fiat Luxor*, “Let there be light,” to convey the goal of the university is to find knowledge and distribute it. This is an essay on gaining knowledge from light in a specific way. It is focused on addressing the problems that exist when viewing light that propagates over the ocean, parallel to the water surface—the maritime domain.

In the maritime domain, fog, haze and turbulence hinder the ability to observe objects of interest on the ocean surface. When fog is present, images presented to the sensor become low in contrast. Turbulence reduces the ability of an optical system to capture high resolution imagery and the structure of the images in time are perturbed severely. This work is an investigation of four problems not addressed specifically by the research community that exist in the development of an image or video surveillance system that is deployed in the maritime domain.

The first problem is best formed as a question. Is it better to enhance contrast before image (or video) compression or after compression? It is common to require a surveillance system to compress the image data it is collecting, either through JPEG compression for images or as complex as H.264 for video streaming. This paper addresses how the subjective and perceptual quality from image and video coding and decoding is affected when fog is present.

The next problem addressed is how fog is removed from a single image. There exist several methods that address this problem and are called *single image*
This work presents a new Color Ellipsoid Framework that describes how single image defogging methods gain information from fog. Based on properties from the framework, a new single image defogging method is proposed.

The results from the first problem—enhancement and compression—and the color ellipsoid framework are utilized in developing a solution to the third problem we address in this work. We add complexity to the surveillance system idea by requiring it to be autonomous and capable of enhancing contrast when needed. How will the system know when to enhance the contrast of the images? More specifically, given two images of the same scene but different contrast, can one develop a no-reference perceptual contrast enhancement metric?

For the final problem, we piece together our findings from the first, second, and third problems and develop a joint Contrast Enhancement Turbulence Mitigation (CETM) method. We provide an analysis for how object tracking and frame averaging are affected when contrast is degraded or when fog is present. Using our analysis we are able to develop an efficient CETM method that provides excellent results. In addition, a turbulence mitigation metric (TMM) is developed according to principles from the Color Ellipsoid Framework to provide a method of measuring temporal consistency.

This work is organized as follows. We discuss the effect of image enhancements and compression in Chapter 2. Chapter 3 introduces the proposed Color Ellipsoid Framework and is extended into Chapter 4 where a new perceptual contrast enhancement metric is proposed. The analysis and development of a joint contrast enhancement metric turbulence mitigation method is provided in Chapter 5. We provide the conclusion for Chapters 2 - 5 in Chapter 6.
Chapter 2

Pre/Post Contrast Enhancements

“The world is full of obvious things which nobody by any chance ever observes.” - Sherlock Holmes in The Hound of the Baskervilles

Suppose an engineer is tasked to design a remote video surveillance system. This remote portion will have a camera located in an environment where fog and haze is a common occurrence (e.g., near an ocean). The video from the camera is transmitted to a location (control station) where either the video is stored or is used for real-time surveillance and automatic target recognition (ATR). The video must be compressed in order to accommodate low bandwidths (e.g., H.264/AVC) or low complexity coders (e.g., MJPEG). The presence of fog and haze cause the video to be low in contrast [10]. Therefore the engineer must decide on a contrast enhancement and where it should occur. If the enhancement is applied before transmitting, then this is what we call Prefilter. If the enhancement is applied after transmitting, then we call it Postfilter. (We will refer to Pre and Post to simplify naming for Prefilter and Postfilter respectively.) Once the enhancement is chosen, the engineer is presented with another decision: Should a Pre or Post enhancement method be used?

The above scenario describes the crux of our investigation. Should one design a system that requires contrast enhancements and compression with a Pre or Post methodology? A Pre design may require more complex processing at the remote side which will require more power. However, the Post design (albeit simpler by requiring a less complex remote system) may suffer from compression
artifacts.

In Section 2.1 we provide a background in contrast enhancements and then Section 2.2 presents a new video dehazing algorithm that is based on a dichromatic model to account for the physical nature of fog and haze. This method is designed to be fast for near real-time video processing and is used for our simulated comparisons of Pre and Post methods.

In Section 2.3 we investigate what happens to a hazy image when it is compressed with JPEG and explore how compression artifacts differ when Pre and Post are applied. We then follow with Section 2.4 by looking at how dehazing or not dehazing can affect motion compensation and demonstrate the subjective and objective performance of our dehazing algorithm used as a Pre or Post filter with an H.264 video codec in Section 2.4.1.

2.1 Background in Contrast Enhancements

The contrast enhancements mentioned are commonly used for surveillance applications because the viewing environment is outside the control of the observer. Haze, fog, or non-ideal camera settings can cause the reduction of contrast of an image. The purpose is to improve either subjective performance for analysts (human receivers) and/or objective performance for automatic target recognition type systems.

There are several types of enhancement techniques which have been studied to increase the contrast of images. In [11] the enhancements are categorized as either global or adaptive techniques. In [12], two more classes of contrast enhancements are described: spatial and transform domain enhancements. Globally applied, spatial domain techniques that are statistically driven are studied in [13, 14, 15]. Their spatially adaptive counterparts are in [16, 14, 15]. There are also contrast measure driven enhancements that operate on an image based on locally measured contrasts [17, 11] which are spatially adaptive and operate in the spatial domain. In addition to spatially varying/spatial domain techniques, there are dehazing methods that account for images affected by fog and haze using a
physical model of the scene [18, 19, 3, 20, 4, 1]. In the transform domain, methods with spatially varying techniques are presented in [21, 22, 23, 12]. In [12], the authors take a unique approach based on the human visual system (HVS) model by loosely using a Retinex method [24, 25] while operating in the DCT domain. Most of the transform domain techniques may be applied either at the encoding end, \textit{Pre}, or decoding end, \textit{Post}, but the performance comparisons are not explored.

What is not commonly looked at is how the performance of a spatially adaptive/spatial domain method is affected when compression is applied before or after enhancing. Therefore in Chapter 2 we will investigate how the performance of a spatially adaptive/spatial domain method (dehazing) can be affected by transform domain manipulations, particularly DCT-based compressions.
2.2 Fast Single Image Dehazing

In Fig. 2.1 we have an illustration of the physics-based dichromatic model. The dehazing contrast enhancement filter we intend to use for this investigation is based on this dichromatic model,

\[ \tilde{x}(m, n) = x(m, n)t(m, n) + a(1 - t(m, n)). \]  

(2.1)

The RGB color hazy image \( \tilde{x} \) received at pixel location \((m, n)\) is

\[ \tilde{x}(m, n) = [\tilde{x}(m, n, r), \tilde{x}(m, n, g), \tilde{x}(m, n, b)]^T, \]

where \(r, g, b\) are the red, blue, and green channels respectively. Likewise, the image without haze \(x\) at pixel location \((m, n)\) is

\[ x(m, n) = [x(m, n, r), x(m, n, g), x(m, n, b)]^T. \]

The transmission, \(t(m, n)\), is an extinction term characterized by

\[ t(m, n) = e^{-\beta d(m, n)}. \]

The term \(d(m, n)\) represents the distance from the camera to the target in the scene at \((m, n)\) and \(\beta\) is the scattering coefficient which is dependent on the size of the scattering particle\(^1\).

We assume the scattering is homogeneous which restricts \(\beta\) to be spatially invariant. The transmission, \(0 \leq t(m, n) \leq 1\), attenuates the target color. The atmospheric veiling \([1]\), \(a(1 - t(m, n))\), is responsible for adding the airlight color \(a\). The airlight can be thought of as the color of a black target just above the horizon infinitely far away.

See [18, 19, 3, 4, 1] for a more detailed look at this model and its applications. What is most attractive with this type of method is in addition to enhancing the image, one may also recover the depth of the scene which is a beneficial side effect for surveillance applications.

\(^1\)The scattering coefficient is actually dependent on wavelength and size. Narasimhan et al. [10] showed that this scattering can be assumed to be the same for visible wavelengths in hazy and foggy scenes. For our case \(\beta\) is not a function of wavelength.
Our method is based on the work from [4] where the transmission is initialized using a Dark Channel Prior (DCP). He et al. found that clear day images exhibited a mostly dark image $\theta_D$ when the following morphological multi-scale operator was applied to clear day images:

$$\theta_D(m, n) = \min_{k,l \in \Omega(m, n)} \left( \min_{c \in \{r,g,b\}} \frac{\tilde{x}(k,l,c)}{a(c)} \right),$$

(2.2)

where $\Omega$ is a square shape of sizes 5x5 or 7x7. First, for each pixel located at $(m, n)$, the lowest value from all color channels is chosen. Then the lowest pixel value is chosen within the neighborhood $\Omega$ centered at pixel location $(m, n)$. [4] showed that hazy images exhibited "brighter" $\theta_D$ that reflected the depth of the scene. This $\theta_D$ effectively was a prior for the transmission $t(m, n)$.

There are assumptions when using (2.1) which we should address. First the scattering $\beta$ is assumed to be the same for all wavelengths. This assumption works well in fog and haze but breaks down when the scattering particles become smaller [26]. Second, the scattering needs to be homogeneous or for our spatially invariant scenario. Finally, the model in (2.1) is suitable for overcast lighting. Typically dehazing methods breakdown in shadow areas which are caused by direct lighting.

The $\theta_D$ image is then used to create a coarse estimate of transmission by

$$t_D(m, n) = 1 - w\theta_D(m, n)$$

(2.3)

where $w$ is set to 0.95 for most scenes.

Before we move on to the smoothing step that [4] uses to refine the transmission estimate, lets take a closer look at the meaning of (2.2). To simplify our exploration, suppose we only have one color channel such that the image $\theta_D$ becomes

$$\theta_D(m, n) = \min_{k,l \in \Omega(m, n)} \frac{\tilde{x}(k,l)}{a}$$

(2.4)

$$= \min_{k,l \in \Omega(m, n)} \left[ x(k,l)t(k,l)/a + 1 - t(k,l) \right].$$

(2.5)

If we assume that within the patch $\Omega(m, n)$ the radiance of the target is smooth...
\( x(k, l) = x \) then we have
\[
\theta_D(m, n) = \min_{k,l \in \Omega(m,n)} \left[ \left( \frac{x}{a} - 1 \right) t(k,l) + 1 \right] \\
= 1 + \left( \frac{x}{a} - 1 \right) \left( \min_{k,l \in \Omega(m,n)} t(k,l) \right).
\] (2.6)

In (2.6) the DCP \( \theta_D \) is a function of transmission \( t(k,l) \) in a patch with smooth radiance (no texture). Next, if we assume the transmission is smooth \( t(m, n) = t \) which means smooth depth within the patch \( \Omega(m, n) \) then we have
\[
\theta_D(m, n) = \min_{k,l \in \Omega(m,n)} \left[ \left( \frac{x(k,l)}{a} - 1 \right) t \right] \\
= \left( \frac{1}{a} \left( \min_{k,l \in \Omega(m,n)} x(k,l) \right) - 1 \right) t.
\] (2.7)

In (2.7) with smooth depth but high texture in patch \( \Omega(m, n) \), the DCP is strictly a function of the minimum value from \( x(k, l) \) but yet attenuated by \( t \). This also means that the DCP of a texture at a close distance \( \Omega(m_c, n_c) \) will be greater than the DCP of the same texture at far distance \( \Omega(m_f, n_f) \)
\[
\theta_D(m_c, n_c) > \theta_D(m_f, n_f).
\] (2.8)

Based on our simplifications, (2.6) and (2.7) show that there is a loose relationship between the DCP \( \theta_D \) and transmission \( t \). The relationship breaks down when the depth variation is not smooth and when the texture is high (occlusion boundaries).

To account for the DCP breaking down in textured and varying depth regions, He et al. use a spectral matting method to smooth or refine the transmission estimate \( t_D \) to \( t_{SD} \). The refinement method used by [4] works well but requires several seconds to process because of the generation of the Laplacian matrix that initializes the spectral matting method. This is not ideal for near real-time video processing. We will refer the reader to [4] for the details on this refinement step.

Although Fattal [3] uses a statistical measure to create an estimate for the transmission, they also require a refinement step where they use a statistical smoothing method which is a Gauss Markov Random Field method with the hazy image projected onto the airlight color used as an induced prior. This refinement also takes several seconds to process for one frame.
The recovered dehazed image using the non-smoothed DCP is
\[ x_D(m,n) = \tilde{x}(m,n) - a \max(t_D(m,n), \epsilon) + a, \] (2.9)
with \( \epsilon \) set to be small (0.01) to prevent a division by zero. One of the main visual drawbacks from not refining the DCP transmission estimate is a halo effect around occlusion boundaries. Our proposed method is a modification to the DCP method in (2.2) by replacing the second minimum operator with a median operator. Similar to [1], the median filter was chosen because of its ability to preserve edges in detail areas while still permitting dehazing in smooth areas. Instead of trying to estimate the atmospheric veiling by applying multiple median filters as in [1], we only use the filter once to produce a smoother transmission estimate while preserving edges. This reduces the complexity from using at minimum two median filter passes with large kernel sizes (41x41) to a single median filter with a smaller kernel size (11x11). Therefore this median operation is suitable for minimizing the halo effects early in the estimation in order to reduce the complexity (or need) of the smoothing step and operate faster than current methods.

The proposed Median DCP (MDCP) is constructed as
\[ \theta_M(m,n) = \text{med}_{k,l\in\Omega(m,n)} \left( \min_{c\in\{r,g,b\}} \frac{\tilde{x}(k,l,c)}{a(c)} \right). \] (2.10)
Likewise, the MDCP method initializes the transmission \( t_M(m,n) \) by
\[ t_M(m,n) = 1 - w\theta_M(m,n), \] (2.11)
with \( w \) set to 0.95 for our experiments. The recovered dehazed image using the MDCP is
\[ x_M(m,n) = \frac{\tilde{x}(m,n) - a \max(t_M(m,n), \epsilon)}{\max(t_M(m,n), \epsilon)} + a. \] (2.12)

Similar to the previous consideration for (2.2), (2.10) is simplified by assuming there is only one color channel,
\[ \theta_M(m,n) = \text{med}_{k,l\in\Omega(m,n)} \left( \frac{\tilde{x}(k,l)}{a} \right) \]
\[ = \text{med}_{k,l\in\Omega(m,n)} x(k,l)t(k,l)/a + 1 - t(k,l). \] (2.13)
We will get the same conclusions in (2.13) with assuming smooth radiance and smooth depth as shown in (2.6) and (2.7), respectively by replacing the operator \( \min \) with \( \text{med} \).

However, let’s take a look at how the MDCP method compares to the DCP method at occlusion boundaries. First, let the vector \( \mathbf{w}_{m,n} \) contains the sorted arrangement of the pixels \( \tilde{x} \in \Omega(m, n) \) such that the lowest value is at \( \mathbf{w}_{m,n}(0) \) and highest value at \( \mathbf{w}_{m,n}(N^2 - 1) \) for a window \( \Omega \) of size NxN

\[
\mathbf{w}_{m,n} = \begin{bmatrix}
\min_{k,l \in \Omega(m,n)} \tilde{x}(k, l) \\
\vdots \\
\max_{k,l \in \Omega(m,n)} \tilde{x}(k, l)
\end{bmatrix}.
\tag{2.14}
\]

We now have the non-smoothed DCP and MDCP as

\[
\theta_D(m, n) = \mathbf{w}_{m,n}(0)/a,
\tag{2.15}
\]

and

\[
\theta_M(m, n) = \mathbf{w}_{m,n} \left( \frac{N^2 - 1}{2} \right)/a
\tag{2.16}
\]

for \( N \) odd.

Suppose at an occlusion boundary there are two different types of pixels \( \tilde{x} \) in patch \( \Omega(m, n) \) that are either foreground pixels \( \tilde{x}_f \) or background pixels \( \tilde{x}_b \) in the sorted vector \( \mathbf{w}_{m,n} \)

\[
\mathbf{w}_{m,n} = \begin{bmatrix}
\min_{k,l \in \Omega(m,n)} \tilde{x}_f(k, l) \\
\vdots \\
\max_{k,l \in \Omega(m,n)} \tilde{x}_f(k, l) \\
\min_{k,l \in \Omega(m,n)} \tilde{x}_b(k, l) \\
\vdots \\
\max_{k,l \in \Omega(m,n)} \tilde{x}_b(k, l)
\end{bmatrix}.
\tag{2.17}
\]

As shown in Fig. 2(a), the closer pixels (less atmospheric veiling [1]) are darker than hazier background pixels that are farther away. Notice that comparing (2.15) with (2.16), even if there was one foreground pixel at the extreme edge of the neighborhood \( \Omega(m, n) \), the DCP will choose the foreground pixel value \( \tilde{x}_f \) at \( (m, n) \)
2.3 Dehazing Effects on Coding Artifacts

We will now explore how compression artifacts are affected when dehazing is employed before or after compression. The model used in this section is JPEG compression with uniform quantization. See Fig. 2.3. The input image is first converted into a YUV (or YCrCb) colorspace using a 3x3 projection matrix $R$.

\[(y, c_r, c_b)^T = Rx. \quad (2.18)\]

If we are using a Pre method with a perfect dehazing algorithm, then we simply have $y$ as our luminance. If the chosen method is Post then the resulting luminance is still hazy $\tilde{y}$.

Typically the U and V channels are sub-sampled spatially in JPEG compression [27, 28]. For simplicity we will focus on the luminance, $y$, which is not
sub-sampled and is obtained by multiplying the first row of the matrix, \( r_1^T \), by the input image. Therefore using our dichromatic model, the hazy luminance is

\[
\tilde{y}(m,n) = r_1^T \tilde{x}(m,n) = r_1^T x(m,n) t(m,n) + r_1^T a(1 - t(m,n)) = y(m,n) t(m,n) + ay(1 - t(m,n))
\]

(2.19)

where \( y \) is the luminance of the non-hazy image pixel \( r_1^T x(m,n) \) and the airlight projected onto the luminance channel is \( a_y = r_1^T a \).

Once the luminance is obtained, the values are offset to be within the range of \([-2^P, 2^P - 1]\) for \((P + 1)\) bit data. For clarity purposes we will not make a notational difference and from this point on assume the luminance values for \( \tilde{y} \) and \( y \) are subtracted by \( 2^P \). This brings us to the junction between the “Level and Offset” and “\( N \times N \) Block DCT” blocks in Fig. 2.3.

The next step is to apply an \( N \times N \) forward Type-II DCT. In direct form, the \((i,j)\)\(^{th}\) \( N \times N \) block of the transformed luminance is

\[
f_{i,j}(u,v) = K(u,v) \sum_{0 \leq m,n \leq N-1} y(Ni + m, Nj + n) C_u(m) C_v(n)
\]

(2.20)

for \( u, v = 0, ..., N - 1 \) and

\[
C_s(k) = \cos \left[ \frac{(2k + 1)s\pi}{2N} \right],
\]

(2.21)

\[
K(u,v) = \alpha(u) \alpha(v),
\]

(2.22)

\[
\begin{align*}
x(m,n) &\in \mathbb{R}^3 \\
y_{i,j}(m,n) &\in \mathbb{Z} \\
f_{i,j}(u,v) &
\end{align*}
\]

\[
\begin{align*}
f^q_{i,j}(u,v) &
\end{align*}
\]

**Figure 2.3:** Block diagram for JPEG compression.
and

\[
\alpha(s) = \begin{cases} 
\sqrt{1/N}, & \text{if } s = 0 \\
\sqrt{2/N}, & \text{if } s \neq 0.
\end{cases}
\]  

(2.23)

(2.24)

In [20], the dichromatic model is used to explain the contrast at iso-depths (smooth depth regions) due to haze or fog is reduced. If we measure the gradient of the iso-depth and threshold the result and sum them, \( \|\nabla \cdot \| \), we get a contrast measure. Therefore we have a relationship

\[
\|\nabla \cdot Y_{i,j}\| > t_{i,j}\|\nabla \cdot \tilde{Y}_{i,j}\| = \|\nabla \cdot \tilde{Y}_{i,j}\|
\]

(2.25)

for any iso-depth luminance region \( Y \) at \((i, j)\) and \(0 \leq t_{i,j} \leq 1\).

Similarly, we will use the above framework to explore how compression artifacts differ with hazy and non-hazy images.

### 2.3.1 Ringing Artifacts

When an image is decompressed, ringing artifacts will occur when frequency components are lost at the compression side. This loss is caused by the quantization step (block “Uniform Quant.” in Fig. 2.3).

Using (2.19) and (2.20), the DCT of the hazy luminance at block \((i, j)\) looks like

\[
\tilde{f}_{i,j}(u, v) = K(u, v) \sum_{0 \leq m, n \leq N-1} \left[ t_{i,j}y(Ni + m, Nj + n) + a_y(1 - t_{i,j}) \right] C_u(m)C_v(n).
\]

(2.26)

If we assume the depth is the same at every pixel within block \((i, j)\) \((t_{i,j}(m, n) = t_{i,j})\) then (2.26) becomes

\[
\tilde{f}_{i,j}(u, v) = K(u, v) \sum_{0 \leq m, n \leq N-1} \left[ t_{i,j}y(Ni + m, Nj + n) + a_y(1 - t_{i,j}) \right] C_u(m)C_v(n).
\]

(2.27)
Next we associate the zigzag frequency $\nu$ with the horizontal and vertical frequencies, $u$, and $v$ respectively. The operators $Z_u(\nu)$, $Z_v(\nu)$, and $Z_\nu(u,v)$ are given as

$$u = Z_u(\nu), \quad v = Z_v(\nu), \quad \nu = Z_\nu(u,v),$$

which allows us to parameterize (2.27) with $\nu$. Therefore we parameterize with $\nu$ as

$$\tilde{f}_{i,j}^z(\nu) = \tilde{f}_{i,j}(Z_u(\nu), Z_v(\nu)). \quad (2.28)$$

Thus the DC component of $\tilde{f}_{i,j}^z(\nu)$ is $\tilde{f}_{i,j}^z(0)$. Ignoring the DC component and hence the effect of the addition of $(1 - t_{i,j})a_y$, the AC components with $1 \leq \nu \leq N^2 - 1$ are

$$\tilde{f}_{i,j}^z(\nu) = K(Z_\nu(u), Z_\nu(v)) \sum_{0 \leq m,n \leq N-1} [t_{i,j}y(Ni + m, Nj + n)]
C_u(\nu_m)C_v(\nu_n)
= t_{i,j}f_{i,j}^z(\nu), \text{ for } 1 \leq \nu \leq N^2 - 1. \quad (2.29)$$

As shown in (2.29) the relationship between the AC components of the hazy block $\tilde{f}_{i,j}^z$ and its non-hazy counterpart $f_{i,j}^z$ is an attenuation by $t_{i,j}$ (assuming $t_{i,j}$ is the same throughout the block at $(i,j)$).

If we apply quantization $q(\nu)$ for each DCT coefficient we get

$$\tilde{f}_{i,j}^q(\nu) = \left\lfloor \frac{\tilde{f}_{i,j}^z(\nu)}{q(\nu)} + 1/2 \right\rfloor, \quad (2.30)$$

where the operator $[x + 1/2]$ is simply a rounding to the nearest integer operation on $x$. After the quantization process (2.30), an AC component $h^z(\nu)$ is annihilated when

$$h^z(\nu)/q(\nu) < 1/2. \quad (2.31)$$

We can represent the probability a hazy AC coefficient is annihilated as

$$P \left[ |\tilde{f}_{i,j}^z(\nu)| < \frac{q(\nu)}{2} \right], \quad (2.32)$$
and the probability a non-hazy AC coefficient is annihilated as
\[ P \left[ |f_{i,j}(\nu)| < \frac{q(\nu)}{2} \right]. \quad (2.33) \]

With (2.29) and (2.32) we can relate the probability using the non-hazy AC coefficient,
\[
P \left[ |\tilde{f}_{i,j}(\nu)| < \frac{q(\nu)}{2} \right] = P \left[ |t_{i,j}f_{i,j}(\nu)| < \frac{q(\nu)}{2} \right] = P \left[ |f_{i,j}(\nu)| < \frac{q(\nu)}{2t_{i,j}} \right].
\]
(2.34)

(since \( t_{i,j} \) is always positive). For this particular analysis, let’s put another restriction on \( t_{i,j} \) such that \( 0 \leq t_{i,j} < 1 \) which is reasonable because at \( t_{i,j} = 1 \) would mean the distance to the camera is zero. Using this additional restriction, shifting the threshold from \( \frac{q(\nu)}{2} \) to \( \frac{q(\nu)}{2t_{i,j}} \) increases the probability (2.33) to (2.34) respectively because of the monotonic increasing property of the cumulative distribution function which gives us the inequality
\[ P \left[ |\tilde{f}_{i,j}(\nu)| < \frac{q(\nu)}{2} \right] < P \left[ |f_{i,j}(\nu)| < \frac{q(\nu)}{2t_{i,j}} \right]. \quad (2.35) \]

With (2.34) and (2.35), we can say the probability that a hazy AC coefficient is annihilated is greater than the probability a non-hazy AC coefficient is annihilated,
\[ P \left[ |\tilde{f}_{i,j}(\nu)| < \frac{q(\nu)}{2} \right] > P \left[ |f_{i,j}(\nu)| < \frac{q(\nu)}{2} \right]. \quad (2.36) \]

The relationship in (2.36) also means, depending on the quantization, the probability of ringing artifacts with the Post method is greater than the probability of ringing using a Pre method.

### 2.3.2 Blocking Artifacts

The cause of blocking artifacts in lossy compression methods is due to the artificial boundaries induced by the block based DCT between neighboring blocks [29]. To compare the severity of blocking we will compare the signal to noise ratios (SNR) on the reconstructed (or decompressed) end of the system. To do this,
we will analytically compare the \((i, j)^{th}\) reconstructed block \(y_{i,j}^r\) which is simply a dequantized and inverse block DCT of \(f_{i,j}^q\),

\[
y_{i,j}^r(m, n) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} K(u, v) f_{i,j}^q(u,v) C_m(u) C_n(v) q(u,v), \tag{2.37}
\]

where \(q(u,v)\) is the uniform quantization at frequency \((u,v)\).

It has been shown in [30] that the reconstructed block \(y_{i,j}^r\) can be characterized as the original signal plus reconstruction noise,

\[
y_{i,j}^r(m, n) = y_{i,j}(m, n) + \epsilon_r \tag{2.38}
\]

where \(\epsilon_r\) is the reconstructed noise. Likewise, the hazy reconstructed block is

\[
\bar{y}_{i,j}^r(m, n) = t(m,n) y_{i,j}(m, n) + a y(1 - t(m,n)) + \epsilon_r. \tag{2.39}
\]

If we take the next step in this analysis by also adding camera noise \(\epsilon_n\) to (2.19), then the image used for dehazing in the Pre method (no compression) is

\[
\bar{y}_{i,j}^{Pre}(m, n) = t_{i,j}(m,n) y_{i,j}(m, n) + a_y (1 - t_{i,j}(m,n)) + \epsilon_n \tag{2.40}
\]

and the hazy reconstructed image used for the Post method is

\[
\bar{y}_{i,j}^{Post}(m, n) = t_{i,j}(m,n) y_{i,j}(m, n) + a_y (1 - t_{i,j}(m,n)) + \epsilon_n + \epsilon_r \tag{2.41}
\]

Now consider if we have a perfect dehazing function \(D(.)\) defined as

\[
D(\tilde{h}) = h = \frac{1}{t(m,n)} (\tilde{h} - a_y) + a_y \tag{2.42}
\]

where the transmission \(t(m,n)\) and airlight \(a_y\) are exactly known. Using (2.40), (2.41), and (2.42), we have the dehazing function to be

\[
D_{i,j}^{Pre} = D(\bar{y}_{i,j}^{Pre}(m, n)) = y_{i,j}(m, n) + \frac{\epsilon_n}{t_{i,j}(m,n)} \tag{2.43}
\]
and

\[ D_{i,j}^{Post} = D(\tilde{y}_{i,j}^{Post}(m,n)) = y_{i,j}(m,n) + \frac{\epsilon_n + \epsilon_r}{t_{i,j}(m,n)}. \]  

(2.44)

Note that in (2.43) the reconstruction error is not present because it hasn’t been compressed. Similar to (2.38) we have the compressed then reconstructed version of \( D_{i,j}^{Pre} \) to be

\[ D_{i,j}^{Pre,r} = D_{i,j}^{Pre} + \epsilon_r. \]  

(2.45)

With \( E[\epsilon_n] = E[\epsilon_r] = 0 \), the expected value for both \( D_{i,j}^{Pre,r} \) and \( D_{i,j}^{Post} \) methods are both equal to \( y_{i,j}(m,n) \). But more interestingly, the variances are

\[ \text{var}[D_{i,j}^{Pre}] = \sigma^2_{y_{i,j}} + \frac{\sigma^2_n}{t^2_{i,j}(m,n)} + \sigma^2_r \]  

(2.46)

and

\[ \text{var}[D_{i,j}^{Post}] = \sigma^2_{y_{i,j}} + \frac{\sigma^2_n + \sigma^2_r}{t^2_{i,j}(m,n)}. \]  

(2.47)

with \( \text{var}[y_{i,j}(m,n)] = \sigma^2_{y_{i,j}} \). Finally, we can show the Pre and Post SNR relationship at block \((i,j)\) for the perfect dehazing function \( D \), \( SNR_{i,j}^{Pre} \) and \( SNR_{i,j}^{Post} \) respectively, is

\[ SNR_{i,j}^{Pre} = \frac{\sigma^2_{y_{i,j}}}{\frac{\sigma^2_n}{t^2_{i,j}(m,n)} + \sigma^2_r} \geq \frac{\sigma^2_{y_{i,j}}}{\frac{\sigma^2_n + \sigma^2_r}{t^2_{i,j}(m,n)}} = SNR_{i,j}^{Post}. \]  

(2.48)

### 2.3.3 JPEG Pre/Post Experiment

We see in (2.48) that for a neighboring block pair \((i,j)\) and \((i,j+1)\) with the same luminance and depth, the Post method (with lower SNR) will have higher noise variance causing more blocking artifacts compared to the Pre method only with equivalence at \( t_{i,j}(m,n) = 1 \). It is also shown in (2.48) that the noise contribution is spatially varying and dependent on the depth of the scene where again \( t_{i,j}(m,n) = e^{-\beta d_{i,j}(m,n)} \) with distance \( d_{i,j}(m,n) \). Therefore the blocking artifacts will become more severe as distance and/or scattering \( \beta \) is increased. Also, if one must estimate the noise variance or try to reduce noise, they must account for spatial dependence which is also addressed in [31].
(a) Synthesized Haze

(b) Dehazed (no compression). Red box is region of interest used for visual comparison.

(c) Uncompressed Dehazed image.

(d) Pre method.

(e) Post method.

**Figure 2.4:** Comparison of dehazed image using Pre and Post methods with similar file sizes.

(a) Original image.

(b) Post method with MDCP.

(c) Pre method with MDCP.

(d) Detected block artifacts in Post image using [2].

(e) Detected block artifacts in Pre image using [2].

**Figure 2.5:** Real world example of Pre and Post using MDCP (2.12) and JPEG compression. (a) Original image. (b) Post method with MDCP. (c) Pre method with MDCP. (d) Detected block artifacts in Post image using [2]. (e) Detected block artifacts in Pre image using [2].
Figure 2.6: Measure of blocking artifacts for each method at different image sizes.

Figure 2.7: A performance comparison of the deblocked Pre and Post methods using JPEG and WILD database.

To demonstrate our analysis we used the Weather and Illumination Database (WILD) [32]. This database provides the depth of a scene along with high resolution images of the same scene in all weather conditions. Using WILD, a synthesized hazy image was created using an arbitrarily chosen airlight value. We then used
(2.42) to reconstruct a haze-free image before and after JPEG compression (Pre and Post respectively). First we modified the JPEG quality parameters for the Pre and Post methods so that the file sizes were similar. For this particular case we used a quality factor of 48 to generate an 83.1 kilobyte image for the Pre method and a quality factor of 70 to generate an 84.7 kilobyte image for the Post method. A cropped region of the results is displayed in Fig. 2.4. Even though we set the quality low enough such that the Pre method resulted in a smaller image size, the Post method suffered more from ringing artifacts as evident around the building edges in Fig. 4(e). We can also see more blocking artifacts in the sky region above the buildings.

Next we compared the Pre and Post methods with JPEG quality values set from 5 to 100 to the uncompressed dehazed image (Fig. 4(b)) and measured the amount of blocking artifacts using a non-reference measure proposed in [2]. We stored the image sizes that resulted from each method and plotted them in Fig. 2.6. From these results we see the Pre method produced significantly fewer blocking artifacts than the Post method, especially at smaller image sizes.

To reduce the effect of blocking in our experiment in order to expose other decoding artifacts (e.g., ringing) we applied a deblocking filter [33] after decompression in the Pre method and after dehazing in the Post method. These images were then compared to the non-compressed dehazed image for PSNR calculations and is shown in Fig. 2.7. Again we see the Pre method has close to 5dB higher performance over the Post method.

A real world example (non-synthesized) of a house image is also presented in Fig. 2.5. Since the depth mapping was unknown, the MDCP method (2.12) was used for dehazing. Both the Pre and Post images were compressed to approximately the same size, 17 kilobytes, where JPEG quality 20 was selected for Pre and 45 for Post. Compression artifacts are more prevalent in the Post image in Fig. 5(b). The blocking artifacts in the luma channels are emphasized using [2] in Fig. 5(d) and 5(e). It is evident the Pre image has fewer blocking artifacts in the center and upper right regions of the image which is where the original has more haze (lower transmission). The blocking artifact measurement for Pre was
0.07 and for Post 0.08.

This concludes the JPEG analysis by showing that it is best to enhance before JPEG compression, Pre, in order to minimize blocking and ringing artifacts.

### 2.4 Dehazing Effect on Motion Estimation

We will now extend our analysis in the still image (JPEG) environment to compressed video. The largest gain in compression is motion estimation for video sequences [34] and plays an important role in commonly used codecs, e.g. MPEG-4 and H.264.

To simplify our analysis without over generalization, we will restrict our focus to inter prediction mode because the investigation of intra prediction accuracy is very complex with its dependency on scene content and multiple modes for 4x4 (9 modes) or 16x16 (4 modes) luma blocks and 4x4 chroma blocks (4 modes). Therefore we will show how dehazing contrast enhancements applied before or after video compression affects the performance of motion estimation.

A block matching algorithm (BMA) is a popular choice for reducing the temporal redundancy between frames in video compression. In a BMA, one motion vector (MV) is generated for each block. A MV represents the displacement between a block in one frame to the best-matched block in the next [35]. Video codecs such as MPEG up to H.264 use BMAs to estimate the MVs [36]. Not every block can be assigned a MV therefore thresholds are used to indicate whether or not a MV is used for a coded block [37].

We begin by using the Sum of Absolute Difference (SAD) method which is commonly used in BMAs to measure the similarity (or dissimilarity) between candidate blocks for motion compensation [38] with and without the presence of haze. First we define the perfectly dehazed SAD measure between the \((i,j)\) reference block (subscript 0) and the \((k,l)\) block in another frame (subscript 1) as

\[
SAD_{i,j}(k,l) = \sum_{0 \leq m, n \leq N-1} |y_0(Ni + m, Nj + n) - y_1(Nk + m, Nl + n)|. \tag{2.49}
\]
Similarly, the SAD of a hazy sequence is

\[
S \tilde{A}D_{i,j}(k,l) = \sum_{0 \leq m,n \leq N-1} |\tilde{y}_0(Ni + m, Nj + n) - \tilde{y}_1(Nk + m, Nl + n)|. \tag{2.50}
\]

It can be shown that if the transmission is constant surrounding block \((i, j)\) such that \(t_{i,j}(m,n) = t_{i,j} = t_{k,l}\) and is consistent in each frame, then we have a relationship between both SAD measures

\[
S \tilde{A}D_{i,j}(k,l) = \sum_{0 \leq m,n \leq N-1} |t_{i,j}(m,n)y(Ni + m, Nj + n) + ay(1 - t_{i,j}(m,n)) - t_{k,l}(m,n)y(Nk + m, Nl + n) - ay(1 - t_{k,l}(m,n))| = t_{i,j}SAD_{i,j}(k,l). \tag{2.51}
\]

Let the set \(S_\tau\) be the set of all SAD values measured within the block search region \(\Omega_B\) that measured lower than a threshold \(\tau\),

\[
S_\tau = \{SAD_{i,j}(k,l) < \tau; \forall(k,l) \in \Omega_B\}, \tag{2.52}
\]

and similarly (with the assumption of constant \(t_{i,j}\)) the set \(\tilde{S}_{i,j}\) be

\[
\tilde{S}_\tau = \{S \tilde{A}D_{i,j}(k,l) < \tau; \forall(k,l) \in \Omega_B\} = \{t_{i,j}SAD_{i,j}(k,l) < \tau; \forall(k,l) \in \Omega_B\}. \tag{2.53}
\]

For completeness, their compliments \(S^c_\tau\) and \(\tilde{S}^c_\tau\) are defined as

\[
S^c_\tau = \{SAD_{i,j}(k,l) \geq \tau; \forall(k,l) \in \Omega_B\} \tag{2.54}
\]

and

\[
\tilde{S}^c_\tau = \{S \tilde{A}D_{i,j}(k,l) \geq \tau; \forall(k,l) \in \Omega_B\} \tag{2.55}
\]

such that

\[
S_\tau \cup S^c_\tau = \Omega_B, \quad \tilde{S}_\tau \cup \tilde{S}^c_\tau = \Omega_B, \quad S_\tau \cap S^c_\tau = \emptyset, \quad \tilde{S}_\tau \cap \tilde{S}^c_\tau = \emptyset.
\]
In practice, a typical search region $\Omega_B$ is $32 \times 32$ pixels. We are assuming within this region that the transmission is constant. In foggy scenes $\beta$ ranges are typically between 0.2 and 0.4 km$^{-1}$ [39]. The ratio of two transmission values is simply

$$\frac{t_2}{t_1} = \exp \{-\beta(d_2 - d_1)\},$$

$$\Rightarrow d_1 - d_2 = \log\left(\frac{t_2}{t_1}\right) \beta.$$  (2.56)

With constant $\beta = 0.3$ km$^{-1}$ and transmission ratio not exceeding 1%, $t_2/t_1 = 1.01$, the distance variation $d_2 - d_1$ using (2.56) is approximately 33 meters. Thus our assumption is invalid where the $\Omega_B$ is observing a scene patch that exceeds 33 meters in variation. For our analysis we are making a reasonable assumption that the search region is over objects that don’t exhibit a large depth variation such that $t$ is constant.

Now, the relationship in (2.53) can further be expressed using (2.51) as

$$\tilde{S}_\tau = \left\{ \text{SAD}_{i,j}(k,l) < \frac{\tau}{t_{i,j}} = \gamma_{i,j}; \forall (k,l) \in \Omega_B \right\}$$

$$= S_{\gamma_{i,j}}$$  (2.57)

where the new threshold, $\gamma_{i,j} = \frac{\tau}{t_{i,j}} \geq \tau$, is a scale of the original chosen threshold $\tau$ (using $0 \leq t_{i,j} \leq 1$).

Therefore the number of SAD values lower than $\tau$, $|S_\tau|$, compared to SAD values less than $\gamma_{i,j}$, $|S_{\gamma_{i,j}}|$, with the number of search blocks $|\Omega_B|$ staying the same is

$$\frac{|S_\tau|}{|\Omega_B|} \leq \frac{|S_{\gamma_{i,j}}|}{|\Omega_B|} = \frac{|\tilde{S}_\tau|}{|\Omega_B|}$$  (2.58)

which leads to the probability relationship

$$P\{S_\tau\} \leq P\{\tilde{S}_\tau\}.$$  (2.59)

In (2.59), we simply state that there are more hazy MV block candidates in the search region $\Omega_B$ than in a dehazed block search region. By dehazing a block region, the number of motion vector candidates are reduced when using a threshold $\tau$ for SAD minimum criteria and when the transmission is the same ($t_{i,j} = t_{k,l}; \forall (k,l) \in \Omega_B$) within the search region $\Omega_B$. 
In (2.59) one can see that if there truly isn’t motion within $\Omega_B$, the number of candidates with the dehazed version, zero, is higher with the hazy version which can lead to false MV estimation. This will cause blocking artifacts and flickering because of spatial-temporal inconsistencies.

### 2.4.1 H.264 Pre/Post Experiments

We performed an experiment by applying our proposed MDCP dehazing method on a video sequence before and after compressing with an H.264 compressor with varying bitrates. The video sequence was of a real scene with haze present that 728 frames long with resolution $720 \times 480$. The compressor was set to High Profile, Level 3.0 [40] with the exception of bitrates 8000 and 10000 kbps. For 8000 kbps, Level 3.1 was chosen whereas Level 3.2 was chosen for 10000 kbps. For all the sequences, The GOP was set to be IPBPB..., with only 4 I frames total a video sequence.

A visual sample of the results are in Fig. 2.8. Within the video sequence, a Coast Guard ship is in view which is indicated with a red box in Fig. 8(b). We see that the Pre method gives a better enhanced view of the ship compared to the Post method.

We then measured the PSNR for each bitrate by comparing the uncompressed dehazed video with compressed video with dehazing applied before (Pre) or after (Post) compression. The plot of the results are in Fig. 2.9. Over all bitrates, the Pre method has a higher PSNR than the Post method.

The blocking artifacts were measured for each video sequence and the results are in Fig. 2.10. Again, the no-reference blocking artifact metric [2] was used for these measurements. For all bitrates, the Pre method exhibited considerably fewer blocking artifacts than the Post method.

What is perplexing at first is why we can have a higher PSNR and a lower bitrate with the Pre method? (Note that the PSNR we mention is always with the uncompressed dehazed video as a reference and not simply the encoder calculated PSNR.) In order to demonstrate why, we analyzed the encoder’s motion estimation performance and recorded the number of MV bits and the residual er-
ror bits for each bitrate and filter method in Table 2.1. The bit ratio for each method, Post/Pre, is added to the table to make it easy to see the bit allocation relationships.

Figure 2.8: A frame from the original video sequence is in (a). A snapshot from the dehazed video is in (b). The red box indicates the region used for closer comparisons. Subfigures (c)-(e) are the comparisons of the Pre and Post methods with H.264 compression at 250 and 500 kbps bitrates.
With the exception of bitrates 750 and 1000 kb/s, the Pre method resulted in fewer MV bits compared to the Post method. This shows fewer MVs were selected in the Pre method which coincides with (2.59). The results show that the encoder was able to find more MV candidates in the Post method where the Pre method prevented the encoder from finding MVs that were below a threshold cost.

The Pre method, however, had more residual error bits (except at bitrates 750 and 1000) which shows the encoder put more effort into preserving details (DC, AC coefficients) compared to the Post method. As mentioned in Section 2.3.1, the block DCT AC coefficients of the residual signals are more likely to be annihilated (2.48) when haze is present. Important details are lost after quantization and can not be recovered at the decoder end which gives the Post method no chance in being able to enhance important details of a video frame. Consequently, the Pre method has a higher PSNR because the encoder’s efforts were driven slightly more into encoding the residual error bits. The preserved details using Pre method contributed to the noticeable improvement in visual quality.

We performed a subjective test with 20 different observers. These observers were presented with a test that displayed two videos side-by-side with all combinations such that the pool of randomly selected videos were different with respect to their bitrates and method (Pre or Post). After each playback, the observer
was asked to compare the left video to the right. The choices the observers were allowed to select for comparing the left video sequence to the right were adjectival categorical judgments ranging from Much worse, Worse, Slightly worse, The same, Slightly better, Better, and Much better. This type of test is called Stimulus Comparison Adjectival Categorical Judgment (SCACJ) [41]. After processing the data, only the results from one observer was rejected due to rejection criteria in [41]. The remaining 19 observer results were used to compute the ITU-R statistics. For the pool of video sequences, we choose a range of bitrates that were close enough together to give us a good resolution in our results but limited the max bitrate to 1000 kbps because it was difficult beyond that to distinguish from the uncompressed sequence.

The SCACJ results are assembled in Table 2.2. The highest ranking video sequence was the sequence that wasn’t compressed which isn’t a surprise. Following that we see the observers preferred the sequences generated by the Pre method over the Post method for any bitrate. The errorbar plot (using the rankings from Table 2.2) of the ITU-R results in Fig. 2.11 shows two significant overlaps

Figure 2.10: Measure of blocking artifacts for each method and bitrate. The metric was applied to every frame in a video sequence and then averaged.
Table 2.1: Comparison of average motion vector and residual bits per frame.

<table>
<thead>
<tr>
<th>kb/s</th>
<th>MV (kbits)</th>
<th>Residual (kbits)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Post</td>
<td>Pre</td>
</tr>
<tr>
<td>250</td>
<td>3.5</td>
<td>3.3</td>
</tr>
<tr>
<td>500</td>
<td>5.6</td>
<td>5.6</td>
</tr>
<tr>
<td>750</td>
<td>7.4</td>
<td>7.5</td>
</tr>
<tr>
<td>1000</td>
<td>8.8</td>
<td>8.9</td>
</tr>
<tr>
<td>1250</td>
<td>10.4</td>
<td>10.0</td>
</tr>
<tr>
<td>1500</td>
<td>11.8</td>
<td>10.8</td>
</tr>
<tr>
<td>2000</td>
<td>13.8</td>
<td>12.0</td>
</tr>
<tr>
<td>3000</td>
<td>16.8</td>
<td>13.9</td>
</tr>
<tr>
<td>4000</td>
<td>19.2</td>
<td>15.2</td>
</tr>
<tr>
<td>5000</td>
<td>20.0</td>
<td>15.9</td>
</tr>
<tr>
<td>8000</td>
<td>20.5</td>
<td>17.2</td>
</tr>
<tr>
<td>10000</td>
<td>20.6</td>
<td>18.0</td>
</tr>
</tbody>
</table>

between the uncompressed sequence (ranked 1st) and 1000 kbps Pre sequence (ranked 2nd), and 250 kbps Pre (ranked 5th) and 1000 kbps Post (ranked 6th) sequences. The observers had difficulty distinguishing between the uncompressed enhanced sequence and the pre-enhanced and compressed sequence. In addition the observers had difficulty distinguishing between the 250 kbps Pre sequence and 1000 kbps Post sequence. This is a significant compression difference by a factor of 4. If one is only concerned with a dehazed video sequence and can decide whether to enhance before or after compression, our results show choosing the Pre method can operate just as well as the traditionally chosen Post method while operating in significantly lower bitrates.

For more results, please visit the following website: http://videoprocessing.ucsd.edu/~kgibson.
Table 2.2: SCACJ Results

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No Compression</td>
<td>7.06</td>
<td>6.82</td>
<td>7.29</td>
<td>0.52</td>
</tr>
<tr>
<td>2</td>
<td>1000 kbps Pre</td>
<td>6.51</td>
<td>6.21</td>
<td>6.81</td>
<td>0.67</td>
</tr>
<tr>
<td>3</td>
<td>750 kbps Pre</td>
<td>6.31</td>
<td>6.06</td>
<td>6.57</td>
<td>0.57</td>
</tr>
<tr>
<td>4</td>
<td>500 kbps Pre</td>
<td>6.09</td>
<td>5.81</td>
<td>6.38</td>
<td>0.63</td>
</tr>
<tr>
<td>5</td>
<td>250 kbps Pre</td>
<td>4.78</td>
<td>4.37</td>
<td>5.18</td>
<td>0.89</td>
</tr>
<tr>
<td>6</td>
<td>1000 kbps Post</td>
<td>4.21</td>
<td>3.81</td>
<td>4.61</td>
<td>0.89</td>
</tr>
<tr>
<td>7</td>
<td>750 kbps Post</td>
<td>4.17</td>
<td>3.85</td>
<td>4.50</td>
<td>0.72</td>
</tr>
<tr>
<td>8</td>
<td>500 kbps Post</td>
<td>3.80</td>
<td>3.53</td>
<td>4.08</td>
<td>0.60</td>
</tr>
<tr>
<td>9</td>
<td>250 kbps Post</td>
<td>2.04</td>
<td>1.62</td>
<td>2.47</td>
<td>0.94</td>
</tr>
</tbody>
</table>

2.5 Acknowledgment

The text of Chapter 2 is adapted from An investigation of dehazing effects on image and video coding, Kristofor Gibson, Dung Võ, and Truong Nguyen, published in IEEE Transactions on Image Processing in February of 2012. The dissertation author is the primary author of this publication.
Figure 2.11: Plot of ITU-R averaged marks with error bars vs. Rank.
Chapter 3

Color Ellipsoid Framework

“All the effects of Nature are only the mathematical consequences of a small number of immutable laws,” - Pierre-Simon Laplace

The phrase single image defogging is used to describe any method that removes atmospheric scattering (e.g., fog) from a single image. In general, the act of removing fog from an image increases the contrast. Thus, single image defogging is a special subset of contrast restoration techniques.

In this article, we refer to fog as the homogeneous scattering medium made up of molecules large enough to equally scatter all wavelengths as described in [42]. Thus, the fog we are referring to is evenly distributed and colorless.

The process of removing fog from an image (defogging) requires the knowledge on physical characteristics of the scene. One of these characteristics is the depth of the scene. This depth is measured from the camera sensor to the objects in the scene. If scene depth is known, then the problem of removing fog becomes much easier. Ideally, given a single image, two images are obtained: a scene depth image and a contrast restored image.

The essential problem that must be solved in most single image defogging methods is scene depth estimation. This is equivalent to converting a two-dimensional image to a three-dimensional image with only one image as the input. The approach to estimating the scene depth for the purpose of defogging is not trivial and requires prior knowledge such as depth cues from fog or atmospheric scattering.
There are eight sections in this article including this section. We provide a background in single image defogging methods in Section 3.1. Section 3.2 presents a detailed description of the atmospheric dichromatic model. Section 3.3 introduces the color ellipsoid framework. The framework is analyzed when fog is present, and our new defogging method is introduced in Section 3.4. We then unify four different single image defogging methods using the color ellipsoid model in Section 3.6. The discussion is provided in Section 3.7.

3.1 Background

The concept of depth from scattering is not new. It has been used by artists to convey depth to a viewer in their paintings as early as the renaissance [43]. The mathematical model of light propagating through a scattering medium and dependence on distance can be traced back to Beer-Lambert-Bouguer, then Koschmieder [44], Middleton [45], Duntley [46] and then McCartney [26]. The light attenuation is characterized as an exponential decaying term,

\[ t_i(\lambda) = e^{-\beta_i(\lambda)d_i}, \]  

(3.1)

where at pixel location \( i \), the transmission \( t_i \) is a function of the scattering \( \beta_i(\lambda) \) and distance \( d_i \). The term \( \lambda \) is the specific wavelength.

Even though depth from scattering is a well-known phenomenon, single image defogging is relatively new, and a growing number of methods exist. The first methods trying to achieve single image defogging were presented by Tan [19] and Fattal [3]. Both authors introduced unique methods that remove fog from a single image by inferring the transmission image or map. Soon afterwards, another unique method called the dark channel prior (DCP) by He et al. [4] supported the ability to infer a raw estimate of \( t \) using a single image with fog present. The DCP method has also influenced many more single image defogging methods (see [47, 48, 49, 50, 51, 52, 53]). Within the same time frame, Tarel and Hautière [1] introduced a fast single image defogging method that also estimates the transmission map.

In this work we address the question: Can existing single image defogging methods be unified with a common model? One key message is the existing meth-
ods estimate the transmission with a common prototype
\[ \hat{t} = 1 - w\theta, \]  \hspace{1cm} (3.2)

where \( w \) is a scaling term, and \( \theta \) is a ‘dark prior’. The DCP method by He et al. [4] was the first to explicitly use (3.2); however, we demonstrate that this is the prototype used also by other methods regardless of their approach. We find that the dark prior is dependent on properties from the proposed color ellipsoid framework. The following single image defogging methods are analyzed within the framework: Fattal [3], He et al. [4], Tarel and Hautière [1], and Gibson et al. [53].

The second key message in this chapter is that a new single image defogging method is proposed. This method is developed using a lemma from the color ellipsoid framework and also estimates the transmission with the same prototype in (3.2).

### 3.2 Atmospheric dichromatic model

For each color \( \lambda \) at pixel location \( i \), the dichromatic atmospheric scattering model [10],
\[ \tilde{x}_i(\lambda) = t_i(\lambda)x_i(\lambda) + (1 - t_i(\lambda))a(\lambda), \]  \hspace{1cm} (3.3)

is commonly used in single image defogging methods for characterizing the intensity of a foggy pixel.

In comparison to the dichromatic reflectance model [54], the diffuse and specular surface reflections are analogous to the direct transmission, \( t_i(\lambda)x_i(\lambda) \), and atmospheric veiling, \( (1 - t_i(\lambda))a(\lambda) \), respectively. The atmospheric scattering causes the apparent radiance to have two chromatic artifacts caused by particles in the air that both attenuate direct transmission and add light induced by a diffuse light source.

For obtaining a defogged image, the goal is to estimate the \( p \)-channel color image \( (x(0), x(1), \ldots, x(p-1))^T = \mathbf{x} \in \mathbb{R}^p \) using the dichromatic model (3.3). (For most cases, \( p = 3 \) for color images.) However, the problem with (3.3) is that it
is under-constrained with one equation and four unknowns for each color channel. Note that there are two unknowns contained within the transmission, $t(\lambda)$, in (3.1).

The first unknown is the desired defogged image $x$. The second unknown variable is the airlight color, $(a(0), \ldots, a(p - 1))^T = a \in \mathbb{R}^p$. This is the color and intensity observed from a target when the distance is infinite. A good example is the color of the horizon on a foggy or hazy day.

The third and fourth unknowns are from the transmission introduced in (3.1). The transmission, $t_i(\lambda) \in \mathbb{R}$, is the exponentially decaying function based on scattering, $\beta_i(\lambda)$, and distance $d_i$.

The scattering $\beta_i(\lambda)$ is itself a function of particle size and wavelength. For foggy days, the scattering is color independent. On clear days with very little fog, the scattering coefficient becomes more dependent on wavelength. In [10], the scattering is assumed to be the same for all wavelengths and also homogeneous for scenes with thick fog down to dense haze [45]. In this article, we make the same assumption that $\beta_i(\lambda) = \beta$ for scenes with at least dense haze present, therefore $t_i(\lambda) = t_i \forall \lambda$. The atmospheric dichromatic model is simplified to:

$$\tilde{x}_i = t_i x_i + (1 - t_i)a,$$

(3.4)

bringing the unknown count down to a total of two for gray-scale or four for red-green-blue (RGB) color excluding estimating $x$. The transmission $t$ is the first unknown and airlight $a$ is the second unknown for gray-scale. For color ($p = 3$), transmission $t$ is one unknown and airlight $a$ has three unknowns.

The single image defogging problem is composed of two estimations using only the input image $\tilde{x}$: the first is to estimate the airlight $a$ and the second to estimate the transmission $t$.

There exists several methods for estimating $a$ [10, 19, 4]. In this article, we will assume that the airlight has been estimated accurately in order to focus the analysis on how transmission is estimated (with possible need for refinement). Therefore, the key problem in single image defogging is estimating transmission given a foggy image.
3.3 Color ellipsoid framework without fog

The general color ellipsoid model and its application to single image defogging was introduced by Gibson and Nguyen in [55] and [5]. This work will be reproduced here to facilitate the development of additional properties of the model in this article.

The motivation for approximating a color cluster with an ellipsoid is attributed to the color line model in [56] which is heavily dependent on the work from [57]. The color line model exploits the complex structure of RGB histograms in natural images. This line is actually an approximation of an elongated cluster where Omer and Werman [56] model the cluster with a skeleton and a 2D Gaussian neighborhood. Likewise, truncated cylinders are used in [57].

We continue the thought presented by Omer and Werman [56] that subsets of these clusters are ellipsoidal in shape. We accomplish this by instead generating an RGB histogram using color pixels sampled from a small window within the image.

Let us begin with modeling the observed apparent radiance at window $\Omega_i$ with pixel location $i$ as a three-dimensional random variable $X_i$,

$$X_i = \{ x \mid x \in \Omega_i \}.$$  \hspace{1cm} (3.5)

Assume that the observed data within the sample window exhibits a normal density,

$$p_{X_i}(x) \sim N(\mu_i, \Sigma_i),$$  \hspace{1cm} (3.6)

with $\mu_i \in \mathbb{R}^3$ and $\Sigma_i \in S_{+}^3$. The covariance matrix is decomposed as

$$\Sigma_i = U_i^T D_i U_i$$  \hspace{1cm} (3.7)

with the eigenvalues in $D_i = \text{diag}(\sigma_{i,1}^2, \ldots, \sigma_{i,3}^2)$ are sorted in decreasing order.

Given color pixels within window $\Omega_i$, we will define the color ellipsoid as

$$E_c(\mu_i, \Sigma_i) = \{ x \mid (x - \mu_i)^T \Sigma_i^{-1}(x - \mu_i) \leq 1 \},$$  \hspace{1cm} (3.8)
parameterized by the sample mean $\mu_i$ and sample covariance $\Sigma_i$. We will drop the parameters for clarity so that $E_c(\mu_i, \Sigma_i) = E_c$.

It is common to assume that the distribution of the color values sampled within $\Omega_i$ is normally distributed or can be modeled with an ellipsoid. The distribution for the tristimulus values of color textures was assumed to be normally distributed by Tan [58]. Even though Devaux et al. [59] do not state that the sample points are normally distributed, they model the color textures with a three-dimensional ellipsoid using the Karhunen-Loeve transform. Kuo and Chang [60] sample the entire image and characterize the distribution as a mixture of Gaussians with $K$ clusters.

In Figure 3.1, we illustrate the concept of approximating the cluster of points from a sample window $\Omega_i$. We have a clear day image with two sample windows located on a tree trunk and dirt road. The color points are plotted in Figure 3.1b. The densities from the data points are then estimated and plotted using two-dimensional histograms for each color plane: red-green, green-blue, and red-blue. The higher the frequency, the darker red the density points become.

**Figure 3.1**: Color ellipsoids in clear natural scene. (a) Clear day scene with a sample window over the tree trunk (white rectangle) and a window over the road (black rectangle). (b) RGB histogram plot of points from the two windows with their densities projected on each color plane. (c) The same RGB histogram plot only with the ellipsoid approximations for each region of interest (ROI). The green ellipsoid is from the tree trunk, and the red ellipsoid is from the dirt road.

In Figure 3.1c, we approximated color ellipsoids to each cluster using principal component analysis, where the sample mean and sample covariances were
used. In Figure 3.1b,c, the upper cluster is from the road and the lower cluster is from the tree trunk. Approximating the RGB clusters with an ellipsoidal shape does well in characterizing the three-dimensional density of the cluster of points.

### 3.4 Color ellipsoid framework with fog

#### 3.4.1 General properties

We derive in this section the constraints for color ellipsoids when fog is present. We first simplify the derivation by assuming that the surface of the radiant object within the sample window is flat with respect to the observation angle so that the transmission $t_i$ is the same within $\Omega_i$ ($t_i = t$).

If the apparent radiance of the same surface is subjected to fog, then it can be shown using (3.4) and (3.6) that the observed foggy patch is also normally distributed with attenuated variance and translated mean,

$$p_{\tilde{X}_i}(\tilde{x}) \sim N(\tilde{\mu}_i, \tilde{\Sigma}_i),$$

with

$$\tilde{\mu}_i = t\mu_i + (1 - t)a,$$

and

$$\tilde{\Sigma}_i = t^2\Sigma_i.$$ 

Note that the transmission is the same within the patch because it is assumed that the depth is flat.

The RGB histogram of the surface and a foggy version of the surface should exhibit two main differences. The first is that the RGB cluster will translate along the convex set between $\mu_i$ and $a$ according to (3.10). Second, with $0 \leq t_i \leq 1$, the size of the cluster will become smaller when fog is present according to (3.11). In this article, we present the following new lemmas.

**Lemma 1.** The transmission $t$ of any scene with fog in the atmosphere $\beta > 0$ has the inequality

$$0 \leq t < 1.$$
Proof. Let $\beta > 0$ since the scene is viewed within the fog. Then, $t = e^{-\beta d} = 1$ holds if and only if $d = 0$. In real world images, however, the distance to the camera is never zero ($d > 0$), therefore $0 \leq t < 1$.

**Lemma 2.** Define the clear day color ellipsoid as

$$\mathcal{E}_c = \{ x \mid (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \leq 1 \},$$

and the foggy day color ellipsoid as

$$\tilde{\mathcal{E}}_c = \{ x \mid (x - \tilde{\mu}_i)^T \tilde{\Sigma}_i^{-1} (x - \tilde{\mu}_i) \leq 1 \}.$$

If the parameters $\mu$ and $\tilde{\mu}$ are formed according to (3.10), and $||\mu||^2, ||\tilde{\mu}||^2, ||a||^2 \in \mathbb{R}_{>0}$, then the centroid of $\mathcal{E}_c$ is closer to the origin than the centroid of $\tilde{\mathcal{E}}_c$.

**Proof.** Let us begin with a reasonable assumption that the airlight is the brightest color in the image,

$$||a||^2 \geq ||\mu||^2. \tag{3.13}$$

The centroid of the foggy day color ellipsoid $\tilde{\mu}$ is within the convex set in (3.10) such that when $t = 0$, $\mu = a$, and when $t = 1$, $\mu = x$. Similarly,

$$||\mu||^2 \leq ||\tilde{\mu}||^2 \leq ||a||^2. \tag{3.14}$$

Lemma 1, however, strictly excludes the point $\tilde{\mu} = \mu$; therefore

$$||\mu||^2 < ||\tilde{\mu}||^2. \tag{3.15}$$

**Lemma 3.** The volume of the color ellipsoid $\mathcal{E}_c$ is larger than the foggy color ellipsoid $\tilde{\mathcal{E}}_c$.

**Proof.** Using (3.11) and denoting det as the determinant, the ellipsoid volumes are

$$\det \tilde{\Sigma} = \det t^2 \Sigma. \tag{3.16}$$

Given Lemma 1, $0 \leq t < 1$, we then have the relationship

$$\det \tilde{\Sigma} < \det \Sigma. \tag{3.17}$$

$\Box$
We demonstrate Lemmas 2 and 3 with a real-world foggy day image set. In Figure 3.2, there are three images of the same tree on a foggy day at three different distances. A sample window $\Omega_i$ is located on the same tree branch in each image. For each $\Omega_i$, the densities are plotted in Figure 3.2d. Note that the densities are ellipsoidal in shape. Also, for the tree branch positioned closer to the camera, the ellipsoid is larger in size and positioned closer to the RGB cube origin ($\Omega_3$). For the tree branch positioned farthest away ($\Omega_1$), the ellipsoid is smaller in size and positioned farther away from the RGB origin.

Figure 3.2: Color ellipsoids in foggy natural scene. (a) to (c) Images of the tree branch in the fog at three different distances with sample windows overlaid on the same branch in each image. (d) RGB histogram of each sample set.

3.4.2 Color ellipsoid model with depth discontinuity

We have assumed in the previous section that the transmission within a sample window is constant. However, this is not always true. For example, the sample window may be centered on a depth discontinuity (e.g., edge of a building).

If depth discontinuities are not accounted for in transmission estimation, then undesired artifacts will be present in the contrast restored image. These artifacts are discussed in more detail in [4, 1, 61]. In summary, these artifacts appear to look like a halo at a depth edge.

To account for the possibility that the sample window is over a depth discontinuity, we characterize the pixels observed within $\Omega$ as a Gaussian mixture
model [62]. The sample window may cover \( K \) different types of objects. This yields \( K \) clusters in the RGB histogram.

Let the \( g^{th} \) random mixture variable at pixel location \( i \) be the summation of disjoint sub-windows of \( \Omega_i \),

\[
\tilde{X}_{i,g} = \{ \tilde{x} \mid \tilde{x} \in \Omega_{i,g} \} \text{ with } \Omega_i = \bigcap_{g=1}^K \Omega_{i,g},
\]

and \( \Omega_{i,h} \cap \Omega_{i,j} = \emptyset, \forall h \neq j \),

and the total mixture distribution become

\[
p_{\tilde{X}_i}(\tilde{x} \mid \Theta_K) = \sum_{g=1}^K \pi_{i,g} p_{\tilde{X}_{i,g}}(\tilde{x} \mid \tilde{\mu}_{i,g}, \tilde{\Sigma}_{i,g}). \tag{3.18}
\]

The parameter vector \( \Theta_K = (\tilde{\mu}_1, \ldots, \tilde{\mu}_K, \tilde{\Sigma}_1, \ldots, \tilde{\Sigma}_K) \) is a culmination of the \( K \) Gaussian mean and covariance parameters defined by Equations 3.10 and 3.11, respectively. The mixture weight \( \pi_{i,g} \) is \(|\Omega_{i,g}| / |\Omega_i| \) with \( \sum_{g=1}^K \pi_{i,g} = 1 \).

An example of the presence of multiple mixtures within \( \Omega \) is shown in Figure 3.3. The sample window is centered on a region with leaves close to the camera and leaves on a tree branch farther away. Even though the plot of the color pixels appear to be one elongated cluster, the existence of two mixtures is evident in the density plots with two distinct dark red regions on each color plane in Figure 3.3b.

Similar to the example in Figure 3.3, let the sample window be small enough to only contain two mixtures \((K = 2)\). Denoting the mixtures with subscripts 1 and 2, and using (3.10) and (3.11), the overall sample mean is

\[
\tilde{\mu} = \pi_1 t_1 (\mu_1 - a) + \pi_2 t_2 (\mu_2 - a) + a, \tag{3.19}
\]

which is the weighted average between the two mixtures. The sample covariance,

\[
\tilde{\Sigma} = \pi_1 t_1^2 \tilde{\Sigma}_1 + \pi_2 t_2^2 \tilde{\Sigma}_2 + \pi_1 \pi_2 (\tilde{\mu}_1 - \tilde{\mu}_2)(\tilde{\mu}_1 - \tilde{\mu}_2)^T \tag{3.20}
\]

\[
= \pi_1 t_1^2 \Sigma_1 + \pi_2 t_2^2 \Sigma_2
\]

\[
+ \pi_1 \pi_2 \left\{ t_1^2 (\mu_1 - a)(\mu_1 - a)^T + t_2^2 (\mu_2 - a)(\mu_2 - a)^T
\right. \\
- t_1 t_2 (\mu_1 - a)(\mu_2 - a)^T - t_1 t_2 (\mu_2 - a)(\mu_1 - a)^T \right\}, \tag{3.21}
\]
Figure 3.3: Example of color ellipsoid with depth discontinuity. (a) Foggy image with ROI centered at a depth discontinuity. (b) RGB histogram of the ROI. Note the presence of two density mixtures.

has a shape influenced by the mixture weights.

Let us simplify even more by assuming that \( \Omega \) is at an extreme depth discontinuity where one of the depths is at infinity or \( t = 0 \). With \( t_1 > t_2 = 0 \), the sample mean and covariance become

\[
\tilde{\mu} = \pi_1 t_1 (\mu_1 - a) + a \quad (3.22)
\]

and

\[
\tilde{\Sigma} = \pi_1 t_1^2 \Sigma_1 + \pi_1 (1 - \pi_1) t_1^2 (\mu_1 - a)(\mu_1 - a)^T, \quad (3.23)
\]

respectively. Instead of the transmission influencing the position \( \tilde{\mu} \) of the ellipsoid, the mixture weight also has influence on the sample mean. Therefore, the problem of ambiguity exists because of the combination of the mixture weight and transmission \( \pi_1 t_1 \). In order to use the sample mean to estimate the transmission value, the mixture weight must be considered.
3.5 Proposed ellipsoid prior method

Part of our key message in unifying existing defogging methods is that the transmission can be estimated using parameters from $\tilde{E}_c$. As an introduction to this unification, we will use Lemma 2 to derive a new unique dark prior.

The principal question to address is how can we infer transmission given the observed color ellipsoid $\tilde{E}_c$. Suppose we use only Lemma 2 to create a cost function such that $||\tilde{\mu}||^2 > ||\mu||^2$. A cost function $J(\hat{t})$ can be created in order to minimize the defogged centroid magnitude $\hat{\mu}$,

$$J(\hat{t}) = ||\hat{\mu}||^2 = \left||\frac{\tilde{\mu} - a}{\hat{t}} + a\right||^2,$$

where the defogged estimate is

$$\hat{\mu} = \frac{\tilde{\mu} - a}{\hat{t}} + a.$$ (3.25)

Minimizing $J(\hat{t})$ is simply trying to make the image darker on average. The cost function $J(\hat{t})$ is minimized when

$$\hat{t}_C = 1 - \frac{a^T \tilde{\mu} - ||\tilde{\mu}||^2}{||a||^2 - a^T \tilde{\mu}} = 1 - \theta_C.$$ (3.26)

Similar to the nomenclature in [4], let the centroid prior, $\theta_C$, be the dark prior using Lemma 2.

The transmission estimate must account for depth discontinuities. One method to acquire the sample mean with respect to the mixture weights is to use the median operator. The median operator is used for this purpose by Gibson et al. [61] and Tarel and Hautiere [1]. In the same fashion, the centroid prior, $\theta_C$ (3.26), can be modified to use the median operator so that depth discontinuities can be considered. We include the median operator when acquiring the centroid of the ellipsoid with

$$\theta_{C,m,i} = \frac{a^T \tilde{\mu}_{m,i} - ||\tilde{\mu}_{m,i}||^2}{||a||^2 - a^T \tilde{\mu}_{m,i}},$$

$$\tilde{\mu}_{m,i}(c) = \text{med} \tilde{x}_j(c), \forall c.$$ (3.28)
where \( c \) is the color channel.

An example of the improvement when using the median operator is in Figure 3.4. The tree in the foreground poses a dramatic depth discontinuity and is evident with the halo around the edge in Figure 3.4b. The halo is diminished using the median operator in Figure 3.4c. From this point on, we drop the subscripts \( m \) and \( i \) \((\theta_{C,m,i} = \theta_C)\) for clarity but still imply the median operator is used.

The defogged image, \( \hat{x} \), is then estimated with

\[
\hat{x} = \frac{\hat{x} - a}{\max(\hat{t}, t_0)} + a,
\]

with \( t_0 \) set to a low value for numerical conditioning \((t_0 = 0.001)\) (see the work by [4] for the recovery method and [1] for additional gamma corrections). For generating the defogged image using the centroid prior, \( \hat{x}_C \), a gamma value of 1/2 was used for the examples in this article, e.g., \( \hat{x}_C^{1/2} \). The complete algorithm for the ellipsoid prior defogging method is in Algorithm 1.

In Figure 3.5, we compare existing single image defogging methods with the centroid prior using a house image provided by Fattal [3]. The defogged image
Algorithm 1 The ellipsoid prior defogging algorithm.

1: procedure ELLIPSOIDPRIORDEFOG(\(\tilde{x}, \Omega, t_0\))

2: a is estimated from methods by [3, 4]

3: for all \(i\) do

4: \(\tilde{\mu}_i(c) \leftarrow \text{med}_{j \in \Omega} \tilde{x}_j(c)\) for all \(c\)

5: end for

6: \(\hat{t}_{C,i} \leftarrow 1 - \frac{a^T \tilde{\mu}_i - ||\tilde{\mu}_i||^2_2}{||a||^2_2 - a^T \tilde{\mu}_i}\) for each pixel \(i\).

7: \(\hat{x} \leftarrow \text{from (3.29) using } \hat{t}_C\text{ and } t_0\).

8: \(\hat{x} \leftarrow \sqrt{\hat{x}}\) (Gamma correction)

9: return \(\hat{x}, \hat{t}_C\)

10: end procedure

Given a foggy image \(\tilde{x}\), a sample window \(\Omega\), and numerical conditioning value \(t_0\), compute the transmission \(\hat{t}_C\) and defogged image \(\hat{x}\).

using \(\hat{t}_C\) has richer color because the cost function, \(J(\hat{t})\), tries to minimize the magnitude of \(\hat{\mu}\) while being constrained to the atmospheric dichromatic model.

![Figure 3.5](image.png)

**Figure 3.5:** Example of single image defogging methods. (a) Original house image from [3], \(\tilde{x}\). (b to f) Defogged images using Lemma 2, He et al. [4], Fattal [3], Tarel and Hautière [1], and Gibson and Nguyen [5], respectively. (g to k) Transmission estimates used for the above defogged images.

The transmission estimate in (3.26) is of the same prototype form in (3.2). Deriving a transmission estimate based on Lemma 2 results in creating a centroid
prior that is a function of the ellipsoid parameters. In Section 3.6, we will show that other single image defogging methods also use the prototype in (3.2) where a dark prior is used. We will also show that the dark prior is a function of the color ellipsoid properties.

### 3.6 Unification of single image defogging methods

The color ellipsoid framework will now be used to analyze how four single image defogging methods (Fattal [3], He [4], Gibson et al. [53], and Tarel [1]) estimate the transmission using properties of the color ellipsoids.

#### 3.6.1 Dark channel prior

In [55], the dark channel prior (DCP) method [4] was explained using a minimum volume ellipsoid which we will reproduce here for completeness.

In order to estimate the transmission, the DCP was used which is a statistical operator,

\[
\theta_{D,i} = \min_{j \in \Omega_i} \left( \min_{c \in \{r,g,b\}} \frac{\bar{x}_j(c)}{a(c)} \right).
\]

(3.30)

The transmission \(\hat{t}_{D,i}\) was then estimated by a linear operation on the prior,

\[
\hat{t}_{D,i} = 1 - w\theta_{D,i},
\]

(3.31)

with \(w = 0.95\) for most scenes. This DCP transmission estimate in (3.31) is of the same form as (3.2).

It was observed by He et al. [4] through an experiment that the DCP of non-foggy outdoor natural scenes had 90% of the pixels below a tenth of the maximum possible value, hence the dark nomenclature in DCP. The \(\hat{t}_D\) is constructed in such a way that it assumes there is a pixel within the sample region centered at \(i\) that originally was black. This is a strong assumption, and there must be more to why this initial estimate works.
He et al. [4] stated that ‘the intensity of the dark channel is a rough approximation of the thickness of the fog.’ This can be understood when the DCP is considered as an approximation of a minimum distance measure to the Löwner-John ellipsoid [55] either from the R-G, G-B, or R-B planes,

\[
\theta_{D,i} \simeq \begin{cases} 
\arg\min_{c,z} & ||z_c - y||_2, \\
\text{subject to} & z_c^T e_c = 0, \\
\text{and} & y^T A^{-1} y = 1,
\end{cases}
\] (3.32)

with equivalence when \( z_c \in \Omega_i \) since a point from the set \( \Omega_i \) is selected instead of the estimated shell of the ellipsoid.

The unit vector \( e_c \) represents the normal to one of the three color planes within the RGB cube. The matrix parameter \( A \) is from the Löwner-John ellipsoid, or minimum volume ellipsoid, that encapsulates the cluster from \( \Omega_i \),

\[
\text{minimize} \quad \log \det A_i^{-1} \\
\text{subject to} \quad \sup_{x \in \Omega_i} ||A_i x + b_i||_2 \leq 1. 
\] (3.33)

An illustration of the relationship between the DCP and the minimum volume ellipsoid is in Figure 3.6. The example is on the R-G plane and demonstrates how the DCP measures the minimum distance from either the red or green axis which is dependent on the position, size, and orientation of the ellipsoid. What was not addressed in [55] was that the DCP is able to estimate transmission (with need for refinements) because it utilizes Lemma 2 and Lemma 3.

The DCP, however, is not a function of the mixture weights such that depth discontinuities are accounted for. This results in halo artifacts when trying to recover a defogged image as discussed in Section 3.4.2. In [4], a soft matting algorithm by Levin et al. [63] was applied to \( \hat{t}_D \) to refine the transmission image, \( \hat{t}_{DS} \). The alpha matting of an image at pixel \( i \) is a function of foreground and background mixtures,

\[
x_i = \alpha_i x_F + (1 - \alpha_i) x_B.
\] (3.34)

Being similar with the atmospheric dichromatic model (3.4) and the alpha matting (3.34), the transmission can be treated as an alpha matting [4],

\[
\hat{x}_i = t_i \hat{x}_F + (1 - t_i) \hat{x}_B.
\] (3.35)
Figure 3.6: DCP and color ellipsoid relationship. Graphical example of the relationship between the DCP and the minimum distance to three different minimum volume ellipsoids on the red-green plane.

The transmission vector $\hat{t}_D$ (canonically stacked by columns) is smoothed into $\hat{t}_{DS}$ by minimizing the cost function

$$J(\hat{t}_{DS}) = \hat{t}_{DS}^T L_{DS} \hat{t}_{DS} + \lambda (\hat{t}_{DS} - t_D)^T (\hat{t}_{DS} - \hat{t}_D).$$

(3.36)

The right hand side of (3.36) was chosen by He et al. [4] to regularize the matting based on the DCP and to enforce smoothing weighted by $\lambda$.

The derivation of the Laplacian matrix, $L$, by Levin et al. [63] is also based on the color line model and hence a function of the color ellipsoid properties. The Laplacian matrix is [63]

$$L = D - W,$$

(3.37)

with

$$W(i,j) = \sum_{k((i,j) \in \Omega_k)} \frac{1}{|\Omega_k|} (1 + \frac{(\tilde{x}_i - \tilde{\mu}_k)^T (\tilde{\Sigma}_k + \frac{\epsilon}{|\Omega_k|} I_{3 \times 3})^{-1} (\tilde{x}_j - \tilde{\mu}_k)}{1 + \frac{(\tilde{x}_j - \tilde{\mu}_k)^T (\tilde{\Sigma}_k + \frac{\epsilon}{|\Omega_k|} I_{3 \times 3})^{-1} (\tilde{x}_j - \tilde{\mu}_k)}}).$$

(3.38)

and $D$ and $I_{3 \times 3}$ being diagonal identity matrices. The Laplacian matting matrix in (3.38) is influenced by the properties of the color ellipsoid ($\mu_k$ and $\Sigma_k$) within
the window $\Omega_k$. The ability of preserving depth discontinuity edges is afforded by the affinity matrix, $W$, which is effective in preserving edges and discontinuities because of its locally adaptive nature \[63\].

The DCP method estimates the transmission with the prototype in (3.2), just like the centroid prior. Additionally, the properties of the color ellipsoids play a key role in the DCP for initial estimation and Laplacian matting for refinement.

### 3.6.2 Fattal prior

The single image defogging method by Fattal \[3\] is a unique method that at first does not appear to be using the prototype in (3.2). We show that Fattal’s method, however, does indeed indirectly develop a dark prior and estimates the transmission with the same prototype in (3.2).

Fattal developed a way to create a raw estimate of the transmission and then employed a refinement step to improve the transmission estimate. We will first investigate how the raw transmission estimate is constructed.

Fattal \[3\] split the observed color $x_i$ into a shade $l_i$ and albedo $r_i$ product,

$$\tilde{x}_i = t_i l_i r_i + (1 - t_i) a,$$

with $x_i = l_i r_i$. The observation made by Fattal was that the sample covariance of the shading and transmission should be statistically uncorrelated over a patch $\Omega$,

$$C_\Omega(l,t) = E_\Omega[(l - E_\Omega[l])(t - E_\Omega[t])] = 0,$$

when the albedo $r$ is constant.

The airlight vector is used to create a subspace so that the observed color pixel is split into two components. The first is the color projected onto the airlight vector

$$\tilde{x}_{a,i} = \langle \tilde{x}_i, a \rangle / ||a||,$$

and the second is the residual which is the observed color pixel projected on the color vector perpendicular to $a$ ($a^\perp$)

$$\tilde{x}_{a^\perp,i} = \sqrt{||\tilde{x}_i||^2 - \tilde{x}_{a,i}^2} = \langle \tilde{x}_i, a^\perp \rangle / ||a||,$$
with \( ||a|| = ||a^\perp|| \) and \( \langle a, a^\perp \rangle = 0 \).

Using the statistically uncorrelated relationship in (3.40) and assuming the albedo \( r \) is constant, Fattal constructs the raw transmission estimate as (dropping \( i \) for clarity)

\[
\hat{t}_F = 1 - \left( \frac{\tilde{x}_a - \eta \tilde{x}_{a^\perp}}{||a||} \right),
\]

with \( \eta = \frac{\langle r, a \rangle}{||r_{a^\perp}|| ||a||} \)

(3.44)

The term \( ||r_{a^\perp}|| \) is the residual albedo projected onto \( a^\perp \).

The estimate \( \hat{t}_F \) also uses the prototype in (3.2) to estimate the transmission. Looking at the right hand side of the raw transmission (3.43) (reintroducing subscript \( i \)),

\[
\hat{t}_{F,i} = 1 - \theta_{F,i}
\]

with \( \theta_{F,i} = \frac{1}{||a||} (\tilde{x}_{a,i} - \eta_i \tilde{x}_{a^\perp,i}) \)

(3.46)

we see yet another prior, the Fattal prior \( \theta_F \). The Fattal prior should behave similar to the DCP (\( \theta_D \)) and centroid prior (\( \theta_C \)) since it is also used to estimate the transmission. The term \( \theta_F \) should match the intuition that it becomes darker (close to zero) when radiant objects are closer to the camera when fog is present.

The Fattal prior utilizes Lemma 2. Note that in (3.4) as the transmission increases, \( t \to 1 \), the foggy pixel moves farther away from the airlight vector, \( a \), while staying on the convex set \( a - \tilde{x} \). This causes more energy to go to the residual, \( x_{a^\perp} \), and less to \( x_a \). Therefore, according to (3.46), the Fattal prior decreases or becomes darker, \( \theta_F \to 0 \), as the transmission increases regardless of the value of \( \eta \).

The Fattal prior also utilizes Lemma 3. To observe this, we analyze the weight factor, \( \eta \), in (3.46) which is a measure of ambiguity. It increases as the albedo color becomes parallel with the airlight or becomes more ambiguous. A low \( \eta \) value means that it is not known whether the pixel is covered by fog or if it is truly the same color as the airlight, but not covered by fog.

The albedo is not known; therefore, the ambiguity weighting is measured by sampling values within a window \( \Omega \) such that the decorrelation in (3.40) is
satisfied,

\[ \eta = \frac{C_{\Omega}(\bar{x}_a, h)}{C_{\Omega}(\bar{x}_a^\perp, h)}. \]  

(3.47)

Since \( \eta \) is measured using a sample region \( \Omega \), we employ the color ellipsoid framework to show that the \( \theta_F \) is dependent on the color ellipsoid.

In order to find more intuition of \( \eta \) and its relationship with the color ellipsoid, let the distribution of \( \bar{x} \) from patch \( \Omega \) be Gaussian with centroid \( \mu \).

Dropping \( \Omega \) and \( i \) for clarity, \( \eta \) becomes

\[ \eta = \frac{E[\bar{x}_a h] - E[\bar{x}_a]E[h]}{E[\bar{x}_a^\perp h] - E[\bar{x}_a^\perp]E[h]}, \]

\[ = \left( ||a||E \left[ \frac{||a||\bar{x}_a - \bar{x}_a^2}{\bar{x}_a^\perp} \right] \right) / (||a||^2 - \mu_a - \mu_a^\perp \mu_h), \]  

(3.48)

where \( \mu_a \) is the centroid of the color ellipsoid projected onto the airlight vector, and \( \mu_h \) is the local average of \( h \),

\[ h = \frac{||a|| - \bar{x}_a}{\bar{x}_a^\perp}. \]  

(3.49)

We can rearrange (3.48) by approximating with Jensen’s inequality \( f(E[X]) \leq E[f(X)] \). The ambiguity weight factor \( \eta \) (3.48) has a lower bound expressed as

\[ \eta = k_1 \left( ||a||E \left[ \frac{||a||\bar{x}_a - \bar{x}_a^2}{\bar{x}_a^\perp} \right] - \bar{\mu}_a \mu_h \right) \]

\[ \geq k_1 \left( ||a|| \frac{||a|-E[\bar{x}_a^2]}{\bar{\mu}_a^\perp} - \bar{\mu}_a \mu_h \right), \]  

(3.50)

with

\[ k_1 = 1/(||a||^2 - \mu_a - \mu_a^\perp \mu_h). \]  

(3.51)

Let the variance of the observed colors projected onto the airlight vector (3.41) be

\[ \sigma^2_a = E \left[ \bar{x}_a^2 \right] - \bar{\mu}_a^2. \]  

(3.52)

Using (3.52) in (3.50), the inequality becomes

\[ \eta \geq k_2 \left( \frac{||a||\bar{\mu}_a - \sigma^2_a - \bar{\mu}_a^2}{\bar{\mu}_a^\perp} - \bar{\mu}_a \mu_h \right), \]  

(3.53)
with \( k_2 = ||a||k_1 \). To view the influence of the color ellipsoid shape, we simplify (3.53) into

\[
\eta \geq k_2 \left( k_3 - \sigma_a^2 \right), \quad (3.54)
\]

with \( k_3 = -\frac{\tilde{\mu}_a}{\tilde{\mu}_a^2} + \left( \frac{||a||}{\tilde{\mu}_a^2} - \frac{\tilde{\mu}_a}{||a||} \right) \tilde{\mu}_a \) \( (3.55) \)

As \( \eta \) increases, the transmission estimate has an increasing influence from the residual color \( \tilde{x}_{a\perp} \). The variance along the airlight vector is the color ellipsoid projected onto the airlight vector,

\[
\sigma_a = a^T \tilde{\Sigma} a. \quad (3.56)
\]

The shape of the color ellipsoid is utilized to influence the ambiguity weight \( \eta \). For example, consider the two ellipsoids in Figure 3.7, labeled \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \). The ellipsoids are positioned such that they both have the same \( \tilde{\mu}_a \). Their orientation and shape, however, are different. \( \mathcal{E}_1 \) has a very small variance projected onto \( a \), \( \sigma_{a,1} \), compared to the variance for \( \mathcal{E}_2 \), \( \sigma_{a,2} \). The \( \eta \) term for \( \mathcal{E}_1 \) is increased which effectively increases the transmission estimate. On the contrary, ellipsoid \( \mathcal{E}_2 \) has a very large variance projected onto \( a \) which produces a lower \( \eta \) value. Due to the shape and orientations, the transmission estimate for the color ellipsoid \( \mathcal{E}_1 \) is higher compared to the transmission estimated for the color ellipsoid \( \mathcal{E}_1 \). We then have the relationship \( t_2 < t_1 \) for the example in Figure 3.7.

The raw transmission estimate, \( \hat{t}_F \), is not complete because several pixels are ignored due to potential division by zero in Equation 3.47. Since the mixture weights \( \pi \) are not considered, depth discontinuities are not accounted for and will produce incorrect estimations. As a refinement step, Fattal uses a Gauss-Markov random field model by maximizing

\[
P(t_{FS}) = \prod_{i \in G} \exp \left( \frac{(t_{FS,i} - t_{F,i})^2}{\sigma_{t,i}^2} \right) \prod_{v_{i,j} \in \Omega_i} \exp \left( \frac{(t_{FS,i} - t_{FS,j})^2}{(\tilde{x}_{a,i} - \tilde{x}_{a,j})^2} / \sigma_s^2 \right), \quad (3.57)
\]

where \( t_{FS} \) is the refinement of \( \hat{t}_F \), and \( G \) are the pixels in \( \hat{t}_F \) that are good. The transmission variance \( \sigma_t \) is discussed in detail in [3] and is measured based on the noise in the image. The smoothing is controlled by the variance value \( \sigma_s^2 \).
Figure 3.7: Geometric interpretation of Equation 3.46. The figure contains two ellipsoids $\mathcal{E}_1$ and $\mathcal{E}_2$ with centroids $\tilde{\mu}_1$ and $\tilde{\mu}_2$, respectively. The projection of the centroids onto the airlight vector $\mathbf{a}$ are the same for both ellipsoids.

The statistical prior on the right hand side of (3.57) not only enforces smoothness but also that the variation in the edges in transmission matches the edges in the original image projected onto airlight. Therefore, if there is a depth discontinuity, the variation will be large in $(\tilde{x}_{a,i} - \tilde{x}_{a,j})^2$ enforcing $\hat{t}_{FS}$ to preserve depth discontinuity edges.

### 3.6.3 Tarel prior

In this section, we will explore the single image fog removal method presented by Tarel and Hautière [1] and relate their intuition with the properties of the color ellipsoids for foggy images. For this analysis, we will make the same assumption that Tarel makes where the foggy image, $\hat{x}$, has been white balanced such that the airlight component is pure white, $\mathbf{a} = (1, 1, 1)^T$.

Instead of directly estimating the transmission, Tarel and Hautière [1] chose to infer the atmospheric veiling,

$$\theta_{T,i} = (1 - \hat{t}_{T,i})a_s,$$  \hspace{1cm} (3.58)
(with $a_s = 1$) which is a linear function of the transmission. Similar to the DCP, we call this term, $\theta_T$, the Tarel prior. We show that this prior is also dependent on the color ellipsoid properties.

Tarel first employs an ‘image of whiteness,’

$$
\bar{w}_i = \min_{c \in \{r,g,b\}} \tilde{x}_i(c).
$$

The intuition in using the image whiteness is similar to the first step used in He’s method to obtain the DCP (3.30). The set of values $\bar{w}_i$ within $\Omega_i$ are the minimum distances from the points in the RGB cluster to either the R-G, G-B, or R-B planes. The atmospheric veiling is estimated by measuring the local average of $\bar{w}$, $\mu_w$, and subtracting it from the local standard deviation of $\bar{w}$, $\sigma_w$.

**Analysis without median operator**

For calculating local averages, Tarel does account for depth discontinuities using a median operator. First, let us consider the simple form to see how the Tarel prior uses the color ellipsoid properties. Tarel uses the local mean and standard deviation of the image of whiteness within the patch $\Omega_i$,

$$
\theta_{T,i} = \mu_{w,i} - \sigma_{w,i}.
$$

As we have done in previous sections, we will again assume that the transmission within the patch $\Omega$ is constant. The local mean at $\Omega_i$, $E_{\Omega_i}[w]$, can then be expanded using (3.4),

$$
\mu_{w,i} = E_{\Omega_i}[w] = E_{\Omega_i} \left[ \min_{c \in \{r,g,b\}} \tilde{x}_i(c) \right] \quad (3.61)
$$

$$
= t \left( E_{\Omega_i} \left[ \min_{c \in \{r,g,b\}} x_i(c) \right] - 1 \right) + 1, \quad (3.62)
$$

where we assume just as Tarel does that the airlight is pure white with a magnitude 1 for each color channel. If the color in the patch is pure white, $\mu_{w,i}$ becomes 1, hence the name image of whiteness. Moreover, if the color within $\Omega_i$ at least has one color component that is zero, then the local mean is only dependent on the atmospheric veiling, $\mu_{w,i} = 1 - t$. 

Suppose the original cluster of points (no fog) had a local average of $\mu_i$. Depending on the orientation of the color cluster, we may approximate the scalar $\mu_w$ by taking the minimum component of the color ellipsoid centroid,

$$\mu_{w,i} \approx \min_{c \in r,g,b} \tilde{\mu}_i(c),$$

where $\tilde{\mu}_i$ is the foggy centroid of the color cluster defined in (3.10). This approximation is illustrated in Figure 3.8.

Figure 3.8: Tarel prior and color ellipsoid relationship. Graphical example of the relationship between the image whiteness and color ellipsoids on the red-green plane. Examples of ellipsoid positions and orientations that are well approximated with measuring the minimum color component of the respective centroid are the clusters with dark ellipses. The dashed blue ellipse is an example of a cluster orientation where the approximation is not valid.

Using the approximation with (3.63), it can be shown that $\theta_T$ is dependent on the position and shape of the color ellipsoid. There are four different clusters in Figure 3.8 that exist from different sample patches, where three of the clusters have the true $\mu_{w,i}$ indicated with them. One can see that these local averages of the image whiteness for each cluster are essentially the minimum component value for the cluster centroid given that the orientation of the cluster is aligned to the
gray color line. Assuming that the orientation is along the gray color line is not
too strong of an assumption since the image itself has been white-balanced and
the dominant orientation is also along the gray color line due to shading or airlight
influence. The fourth cluster, indicated with a dashed blue ellipse, is an example
where this approximation is not valid due to the position and orientation of the
cluster points.

Up to this point, the Tarel prior $\theta_T$ is not a function of the mixture weights
within the sample patch $\Omega_i$ and thus will cause undesirable halo artifacts when
removing fog from the image.

**Analysis with median operator**

To account for estimating properly near depth discontinuities, Tarel and
Hautière [1] chose the median operator because of its edge preserving properties
in order to estimate the atmospheric veiling

$$
\theta_{T,i} = \text{med}_{j \in \Omega_i} w_j - \text{med}_{j \in \Omega_i} \left| w_j - \text{med}_{k \in \Omega_i} w_k \right|.
$$

(3.64)
The sample patch $\Omega_i$ is chosen to be large ($41 \times 41$) to enforce $\theta_T$ to be smooth.
Likewise, since the median operator works well with edge preservation [1], the
edges are considered limiting halo artifacts from being present.

We will show how the Gaussian mixture weights, $\pi_g$, presented in Sec-
section 3.4.2 are considered in the Tarel prior estimate $\theta_T$ (3.64). Let us assume that
the occlusion boundary parameters from the mixture model in Section 3.4.2 are
deterministic but unknown and apply the min operator to each and define the
values for foreground (mixture 1) as

$$
w_{i,1} = \min_{c \in r,g,b} \left(t_{i,1} \mu_{i,1}(c) + (1 - t_{i,1}) a(c) \right),
$$

(3.65)
and background (mixture 2)

$$
w_{i,2} = \min_{c \in r,g,b} a(c)
$$

(3.66)
with the foreground image of whiteness being strictly less than the foreground
image of whiteness $w_{i,1} < w_{i,2}$. When two distinct mixtures exist due to a depth
discontinuity, $\theta_T$ can be simplified to

$$
\theta_{T,i} = \begin{cases} 
    w_{i,1}, & \text{for } \pi_{i,1} > \pi_{i,2} \\
    w_{i,2}, & \text{otherwise,}
\end{cases}
\quad (3.67)
$$

with $|\Omega_i|$ odd. In addition to $\theta_T$ being dependent on the size and position of the color ellipsoid from the sample patch $\Omega_i$, we also show in (3.67) that the mixture weights are employed by Tarel to infer the atmospheric veiling.

A variation on the DCP was presented in [61], called the median DCP (MDCP),

$$
\theta_{M,i} = \text{med}_{j \in \Omega_i} \left( \min_{c \in \{r,g,b\}} \tilde{x}_j(a(c)) \right).
\quad (3.68)
$$

This is essentially a hybrid of both the DCP $\theta_D$ and the Tarel prior $\theta_T$ because of the use of the median operator. In the same fashion as the previous analysis for the DCP and Tarel priors, the MDCP is also a function of the color ellipsoid properties. It also accounts for depth discontinuities by being dependent on the mixture weights $\pi_g$.

### 3.7 Discussion

We have found that we can unify single image defogging methods. The unification is that all of these single image defogging methods use the prototype in (3.2) to estimate transmission using a dark prior. Additionally, each of these dark priors use properties of the color ellipsoids with respect to Lemmas 2 and 3.

We summarize the unification of the single image defogging methods in Table 3.1 by providing the equation used to measure the dark prior. The refinement step taken by each single image defogging method is also provided in Table 3.1.
Table 3.1: Summary of dark prior methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Dark prior</th>
<th>Estimate</th>
<th>Refinement step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centroid</td>
<td>$\theta_C$</td>
<td>$(a^T \hat{\mu} -</td>
<td></td>
</tr>
<tr>
<td>DCP</td>
<td>$\theta_D$</td>
<td>$\min_{j \in \Omega_i} \left( \min_{c \in {r,g,b}} \frac{x_{j,c}}{a(c)} \right)$</td>
<td>Spectral matting</td>
</tr>
<tr>
<td>Fattal prior</td>
<td>$\theta_F$</td>
<td>$\frac{1}{</td>
<td></td>
</tr>
<tr>
<td>Tarel prior</td>
<td>$\theta_T$</td>
<td>$\text{med}<em>{j \in \Omega_i} w_j - \text{med}</em>{j \in \Omega_i}</td>
<td>w_j - \text{med}_{k \in \Omega_i} w_j</td>
</tr>
<tr>
<td>MDCP</td>
<td>$\theta_M$</td>
<td>$\text{med}<em>{j \in \Omega_i} \left( \min</em>{c \in {r,g,b}} \frac{x_{j,c}}{a(c)} \right)$</td>
<td>None</td>
</tr>
</tbody>
</table>

Results from analyzing the single image defogging methods within the color ellipsoid framework are summarized in this table. Each method uses a dark prior, and some employ an extra refinement step with respect to depth discontinuities.

We have discovered that the color ellipsoid framework effectively exposes how the single image defogging methods estimate the transmission when the atmospheric dichromatic model is used mathematically and empirically. Another discovery was that a new dark prior method was created using Lemma 2. A cost function was designed to minimize the average centroid position while staying within the atmospheric dichromatic model. The color ellipsoid framework was the key in the development of this new method. More results can be seen in Figure 3.9.

3.8 Acknowledgment

The text of Chapter 3 is adapted from An Analysis of Single Image Defogging Methods using a Color Ellipsoid Framework, Kristofor Gibson, Truong Nguyen, published in EURASIP Journal on Image and Video Processing in July of 2013. The dissertation author is the primary author of this publication.
Figure 3.9: Additional examples of single image defogging methods. (a) Original house image from [3], \( \tilde{x} \). (b to f) Defogged images using Lemma 2, He et al. [4], Fattal [3], Tarel and Hautière [1], and Gibson and Nguyen [5], respectively. (g to k) Transmission estimates used for the above defogged images. Similarly, (l) is a original foggy image of pumpkins [3]. (m to q) Defog results. (r to v) Transmission estimates.
Chapter 4

Contrast Enhancement Metric

“It is unworthy of excellent men to lose hours like slaves in the labour of calculation which could safely be relegated to anyone else if machines were used.”
- Gottfried Wilhelm Leibniz

An engineer is informed that customers are not satisfied with the image contrast of a surveillance system that is located near an ocean environment. Fog, haze, different lighting conditions and various camera lens parameters (zoom, gain, iris) are just a few of many factors that lower the contrast of the images. The engineer decides to provide a suite of contrast enhancement filters to the users. The filters in the suite are designed to accommodate multiple contrast degradations the system will possibly encounter. The contrast enhancement filters available can only be used one at a time. The engineer is presented with a problem when customers request that they want the enhancement method automated—instead of taking time to select their desired enhancement—so they can focus their attention on more important tasks which involve analyzing the images and making decisions based off of what they observe. Thus the engineer is faced with a few problems. First, the images obtained by the surveillance system will need a contrast enhancement capability that can adapt to varying scenes (light condition, fog, haze, etc.). Second, since the consumer of the images are humans, the enhancement should match human preference. Finally, due to the nature of the problem there is no ideal reference image available to compare enhancement methods.

One possible solution to the problem is to determine which filter is best
for most scenes and simply provide that single enhancement filter. This approach, however, is not an optimal solution since not every possible scene will be enhanced properly. In addition, potentially useful contrast enhancement filters will be discarded and the enhancement suite will not be utilized efficiently. For example, suppose “Enhancement A” works well for images with heavy fog and long focal lengths whereas “Enhancement B” works well for light haze and small focal lengths. If only “Enhancement B” is chosen then the surveillance system is only effective for light haze/small focal length situations. Scenes with heavy fog at long focal lengths will not be enhanced properly and the dynamic enhancement capability of the surveillance system is limited.

A more sensible solution for the engineer is to develop a method that will automatically choose the best contrast enhancement filter by comparing the contrast quality of all available filters. This way all possible scenes are accounted for and the suite of filters is efficiently utilized. A current state of the art method for comparing contrast enhancement algorithms in foggy-day scenes is to use a known clear-day image and synthesized foggy-day images by Tarel et al. [64]. This chapter, however, addresses the more common situation where the clear-day image is not known.

Given that there is not an ideal contrast image available, the engineer must develop a no-reference perceptual based contrast enhancement metric that will provide a means for determining the best enhancement capability that satisfy human observers. A no-reference metric attempts to measure quality without a known high contrast reference image. The reference image is either not attainable or possible to simulate.

There are a growing number of methods that address quality metrics and no-reference type quality metrics that can be separated into two different approaches. First, the Contrast Enhancement Metric (CEM) could be designed to model either the human vision system (HVS) or the physical characteristics of the scene (i.e. fog). In this work, we designate a method as HVS based if it inherently correlates well with human perception or if it is inspired by the HVS. Second, the CEM could be trained from several observations made by humans and utilize a pool of useful
sub-metrics.

A sample of literature that addresses no-reference sharpness metrics are in [65, 66, 67, 68]. Caviedes and Gurbuz develop the metric based on measurements from the spatial domain image edge contrast and kurtosis of frequency domain algorithms [65]. Caviedes and Gurbuz also suggest their algorithm may be useful as a sub-metric to other no-reference sharpness metrics [65]. Zhang et al. propose no-reference perceived sharpness metrics and provide psychophysical experiments to measure correlation. Shaked and Testl [67] propose a no-reference sharpness metric that also correlates with perceived sharpness and is mostly invariant to scene content. Ferzli and Karam utilize the HVS element of just noticeable blur to develop a perceptual based no-reference sharpness metric [68].

The Tenengrad (Thresholded Gradient Magnitude Maximization) criterion was created as an image sharpness measure by Schlag et al. [69]. This Tenengrad metric, however, has also been used for objective contrast measurements since contrast is a measure of the difference between intensities of neighboring pixels. The Tenengrad criterion was used as a no-reference contrast enhancement metric by Chen et al. [14]. The performance of a fog removal method was evaluated with the Tenegrad in [70]. Chen et al. [14] and John and Wilscy [70] both informally state the measure produces results that agree with human visual evaluations. We provide evidence that supports these claims and demonstrate a modified form of the Tenengrad is a useful sub-metric for the CEM because it is a no-reference metric and does reasonably well in agreeing with human perception.

Munteanu and Rosa [71] developed an automatic gray-scale image enhancement and introduced a no-reference enhancement metric that is based on the Tenengrad and an entropy measurement. The edges of the enhancement and information are key factors in this metric. The goal of the entropic measure is to enforce a statistically uniform distribution, similar to Histogram Equalization. This is contrary to the notion of increasing information (minimizing entropy) however is still a measure that is inspired by the HVS. Likewise, this method is a no-reference contrast measure.

Simone et al. [72] developed a no-reference CEM that is heavily influenced...
Figure 4.1: Comparison of images with conflicting AGN and MOS values. An average gradient norm AGN measure from Eq. (4.1) and MOS is associated for each image in (a)-(c). A high AGN or MOS value means high contrast. Note that the MOS for image (b) is higher than the MOS for image (a) and likewise for their AGN values. However the MOS for image (c) is higher than (b) but the AGN value for image (b) is higher than (c).

by the HVS. First, they utilize the Difference Of Gaussian (DOG) model to measure contrast. The DOG is a center-surround pattern that has been found to be effective in modeling the HVS receptive field [73, 74]. Second, Simone et al. take note that local features of an image are more influential in contrast than a global measure. The local features are measured with a multi-level framework by using an image pyramid. Their implementation does well in matching human observations.

Local features were also investigated by Jobson et al. [75] where the local mean and variances were measured in an image. After images were visually optimized, the local features were measured and averaged together in order to measure the contrast globally. Jobson et al. [75] observed that “visually optimized images are more tightly clustered about a single mean value and have much higher standard deviations.”

An Automatic Color Enhancement algorithm (ACE) was introduced by Rizzi et al. [76]. The ACE algorithm was further investigated in [77, 78, 79] where it uses features based on human perception to best select energy functionals that achieve improved image contrast that is coherent with human perception. Though the approach is an automatic method, it has been found that tuning may need to be required for better enhancement results [76].
Hautière et al. [80] introduce a visibility restoration quantitative metric that measures globally three artifacts which are (1) the number of edges that became visible above a user specified threshold after the enhancement, (2) the quality of the edges that were enhanced and (3) the number of pixels that became saturated to white or black in the image after the enhancement. This metric is also used to compare other fog removal methods in [1].

Despite the existence of no-reference contrast enhancement metrics [69, 14, 71, 72, 75, 81, 80], there is no clear solution to determining which method is best for a particular class of scenes (i.e. ocean scenes). A no-reference contrast enhancement metric was proposed by Liu and Ndjiki-Nya [81]. In their work, they find that the development of the metric is heavily dependent on the type of scenes (i.e., natural, urban, etc.).

Choosing a CEM that will properly determine the filter that provides the best contrast improvement is a non-trivial task due to the context of the scene and subjective nature. It is likely that objective measurements and human opinion will conflict with each other. This human preference and objective metric conflict is also observed with respect to image sharpness by Choudhury and Medioni [82] where the highest Tenengrad measure produced images that were “too sharp” according to human preference. Here, we address the conflict with respect to image contrast. For example, consider the images in Fig. 4.1. Similar to the Tenengrad criteria [69], we measured for each image the average gradient norm \( AGN \)

\[
AGN = \frac{1}{N} \sum_{i=1}^{N} \left( \sqrt{G_h(i)^2 + G_v(i)^2} \right),
\]

where \( G_h(i) \) and \( G_v(i) \) are the horizontal and vertical Sobel gradient operators respectively at pixel \( i \), and \( N \) is the number of pixels in an image. The mean opinion scores (MOS) was obtained from 34 different observers who gave a score from 1 to 5 where 5 was very good contrast and 1 was low contrast. The scores were averaged among all observers to produce the MOS. In Fig. 4.1, there are three images of the same scene with different contrast levels. We expect the MOS values to be proportional to the \( AGN \) values such that if the MOS increases then so does the \( AGN \) value. Fig. 4.1(a) is the original low contrast image and as
expected has a low MOS value of 2.1 and a low AGN value of 0.04. In Fig. 4.1(a) and Fig. 4.1(b), the MOS and AGN values increase since the contrast is increased. The MOS and AGN values, however, do not agree with the images in Figs. 4.1(b) and 4.1(c). For the image set in Fig. 1, the highest MOS value of 3.9 is associated with Fig. 4.1(c) but the highest AGN value is associated with Fig. 4.1(b) with a value of 0.42.

To account for metrics that may conflict with human preference in certain scenes, it is possible to take advantage of the performance of multiple CEM values to form a higher aggregate CEM using machine learning. The aggregate CEM would take advantage of the strength of each sub-metric in order to correlate with human preference.

Employing machine learning techniques in the development of no-reference perceptual quality metrics has recently become a more common approach. This approach involves gathering data from human observations, objective measures, and train a model using all of the data to match the human observations. The automation of personalized image enhancements was proposed by Caicedo et al. [83]. In their work, they use data from hundreds of human observations in order to expose the existence of clusters for personalized preferences for image enhancement settings. When an observer is classified into a cluster, the observer’s preference for image enhancement settings are known and auto-enhancement can be applied. Tang et al. [84] develop a no-reference (or blind) image quality measure by mapping low-level image features to subjective observations using a machine learning framework.

The broad nature of images (mountains, trees, ocean, buildings) makes training a CEM difficult and requires a large amount of data for training and fine tuning. There is not enough data that exists where mean opinion scores are associated with images of scenes with contrast degradation—particularly images of the ocean.

We propose using a learning based approach to developing a no-reference perceptual CEM with an introduction of new low-level metrics to improve accuracy. The CEM will learn from human observations of foggy images and multiple sub-
metrics. The approach is similar to the CEM developed in [53] which was trained using several sub-metrics and AdaBoost as the learning framework. In our previous work, we developed a learning based no-reference contrast enhancement metric [53] where low-level image features and human observations were mapped using AdaBoost [85]. A database was created that contained images of ocean scenes with various levels of contrast along with mean opinion scores. A key observation from [53] was that the choice of sub-metrics influenced the performance of the aggregate CEM.

In this chapter, our contribution is to figure out, given a particular scene, which enhancement method would perform well (in correlation with the human observer) by using a learning based no-reference CEM. We will show that the CEM can outperform the existing methods when compared to mean opinion scores. In this work we provide new key contributions. First, the CEM performance is improved in accuracy with respect to [53]. Second, the sub-metrics used to train the CEM are simplified, and the performance is measured. Third, this chapter demonstrates how a few of the sub-metrics developed are based on the physics of the scene and the Color Ellipsoid Framework [86] (Chapter 3), which extends the usefulness of this framework.

4.1 Background

The approach to developing a perceptual based contrast enhancement metric (CEM) can be separated into two classes: First, the CEM could be designed to model the human vision system (HVS) or human vision perception. Second, the CEM could be trained from several observations made by humans. In this section, we will address previous work that fall under these two categories: HVS and Learning Based metrics.

4.1.1 HVS Based Metrics

A simple literature review for human perception of image quality will come across Structural Similarity (SSIM) [87] and the translation insensitive Complex
Wavelet SSIM (CW-SSIM) [37] which are reference based image quality metrics that match well with mean opinion scores [87]. An important feature of both SSIM and CW-SSIM is that they assume the HVS is adapted to a scene’s structural information. Likewise, SSIM and CW-SSIM attempt to measure the quality of an image based on its structure. However CW-SSIM is robust to contrast changes and both methods require a true reference image and are not applicable to a no-reference CEM.

The Tenengrad (Thresholded Gradient Magnitude Maximization) criterion (Eq. (4.1)) was used as an image sharpness measure by Schlag et al. [69]. More importantly, this criterion was used as a contrast enhancement metric in [14]. Though not directly inspired by the HVS, Chen et al. [14] informally state the measure produces results that agree with human visual evaluations. In this paper, we will demonstrate that the Tenengrad is a useful sub-metric for the CEM because it is a no-reference metric and does reasonably well in agreeing with human perception.

Munteanu and Rosa [71] developed an automatic gray-scale image enhancement and introduced an enhancement metric that is based on the Tenengrad and entropy type measure. The edges of the enhancement and information are key factors in this metric. Given a gray-scale image $x$ with $M$ pixels, the Munteanu and Rosa enhancement evaluator $\text{Eval}(x)$ is

$$\text{Eval}(x) = \ln(\ln(E(x) + e)) \frac{\eta(x)}{M} e^{H(x)}. \tag{4.2}$$

The Sobel operator is again used in $E(x)$ where

$$E(x) = \sum_{i=1}^{M} \left( \sqrt{S_x(i)^2 + S_y(i)^2} \right), \tag{4.3}$$

and $\eta(x)$ is actually a Tenengrad measure (Eq. (4.1)) with the threshold set according to the noise of the image. The entropic measure, $H(x)$, is utilized by $\text{Eval}(x)$ as a means to reward an enhancement if the entropy increases,

$$H(x) = \begin{cases} -\sum_i v_i \log_2(v_i), & \text{for } v_k \neq 0, \\ 0, & \text{for } v_k = 0 \end{cases}, \tag{4.4}$$

where $v_k$ is the frequency of gray-scale bin value $k$. This is not true entropy since the probability is not measured. The goal of this entropic measure is to push for
a uniform distribution, similar to Histogram Equalization. This is contrary to the
notion of increasing information (minimizing entropy) however is still a measure
that is inspired by the HVS.

Simone et al. [72] developed a CEM that is heavily influenced by the HVS. First, they utilize the Difference Of Gaussian (DOG) model to measure contrast. The DOG is a center-surround pattern that has been found to be effective in modelling the HVS receptive field [73, 74]. Second, Simone et al. take note that local features of an image are more influential in contrast than a global measure. The local features are measured with a multi-level framework by using an image pyramid. Their implementation does well in matching human observations.

In [88], Simone et al. attempt to add saliency maps for measuring perceptual contrast in order to mimic the human gaze across interesting features in a scene. Although inspired biologically, the saliency mapping did not improve the contrast measure. It was found that although local features are important in measuring contrast, the contrast is judged globally.

The concept of local variations were investigated by Jobson et al. [75] where the local mean and variances were measured in an image. After images were visually optimized, the local features were measured and averaged together. It was found simply that “visually optimized images are more tightly clustered about a single mean value and have much higher standard deviations” [75].

4.1.2 Learning Based Metrics

Employing machine learning techniques in the development of no-reference perceptual quality metrics is also a common approach. This approach involves gathering data from human observations, objective measures, and train a model using all of the data to match the human observations. A few examples of this approach follows. A video packet loss perceptual quality model was trained using subjective experiments and a Generalized Linear Model in [89]. Moorthy and Bovik [90] developed a Blind Image Quality Index (BIQI) that was trained to classify images with a type of distortion before selecting a distortion tailored image quality assessment.
More applicable is the automation of personalized image enhancements by Caicedo et al. [83]. In their work, they use data from hundreds of human observations in order to expose the existence of clusters for personalized preferences for image enhancement settings. When an observer is classified into a cluster, the observer’s preference for image enhancement settings are known and auto-enhancement can be applied.

Tang et al. [84] develop a no-reference (or blind) image quality measure by mapping low-level image features to subjective observations using a machine learning framework.

Similarly, we presented a learning based contrast enhancement metric in [53] where low-level image features and human observations were mapped using AdaBoost. In Chapter 4 we will present an improved version of this CEM along with modified sub-metrics.

4.1.3 AdaBoost

Boosting is a term used for describing a learning algorithm that in general produces accurate predictions based on several weak hypotheses. As long as the weak hypotheses perform slightly better than chance, the aggregation of the multiple observations is boosted to a much more accurate hypothesis. In a similar fashion, we propose to provide the weak hypotheses as observations made by sub-metrics and as long as these hypotheses can perform better than chance and are not heavily correlated with each other, the boosted result will be more accurate.

AdaBoost is a boosting algorithm that adapts each learning step to focus on the more difficult observations or ’hard’ examples [85]. In essence, AdaBoost is greedy in decreasing the loss exponentially [91].

In general, the AdaBoost training algorithm begins with $K$ labeled observations (examples) where the $k^{th}$ observation (or feature) vector is $f_k \in \mathbb{R}^L$. Each observation is associated with a label $y_k \in \{-1, +1\}$ (for binary classes). In this work, the CEM sub-metric results for each image is the observation vector $f_k$. The label $y_k$ is an indicator for which image has the best contrast for observation $k$.

AdaBoost chooses a weak hypothesis $h_i : \mathbb{R} \rightarrow \{-1, +1\}$ for each learning
step $i$ and for each example $k$ by using a weak learning algorithm which we will simply call \textbf{WeakLearn} [92]. The \textbf{WeakLearn} algorithm is a classifier that finds a threshold that optimally dissects the observations into 2 classes for each type of observation $f_k(l)$ for all $l$ data points. For example, the average value of pixels in an image can be a single type of observation and be indicated as the $l^{th}$ component of $f_k, f_k(l)$.

The \textbf{WeakLearn} algorithm can be any choice but for this work we choose the weighted decision stump weak learner that minimizes the error

$$e_{i,j} = \sum D_{i,k}|h_j(f_k(j)) - y_k|,$$

where the distribution vector $D_i \in \mathbb{R}^K$ of weights at iteration $i$ is updated for each observation $k$ and feature $j$.

Finally—after $T$ iterations—the strong hypothesis is a weighted majority of weak hypotheses.

More formally we follow the description in [92, 85] and repeat the algorithm with changes to nomenclature in order keep within the context of this chapter.

Given: $K$ Labeled features $(f_1, y_1), \ldots, (f_K, y_K)$
Initialize: Distribution $D_{1,k} = 1/K \forall k$
For $i = 1, \ldots, T$

- Train weak learners for each feature $j$ by minimizing

$$e_{i,j} = \sum D_{i,k}|h_j(f_k(j)) - y_k|$$

(4.6)

- Choose weak learner with minimum $e_{i,j} \rightarrow h_i$.

- Update $\alpha_i = \frac{1}{2} \ln \left( \frac{1-e_i}{e_i} \right)$ where $e_i$ is associated with the best performing weak learner $h_i$ at iteration $i$.

- Update the weighted distribution:
\[ D_{i,k} = \frac{D_{i,k} \exp(-\alpha_i y_k h_i(f_k))}{Z_i} \tag{4.7} \]

where \( Z_i \) is a normalization factor and \( f_k \) is the observations for the features associated with \( h_i \).

At the end of the iteration, the strong classifier is

\[ H(f) = \text{sign} \left( \sum_{i=1}^{T} \alpha_i h_i(f) \right), \tag{4.8} \]

for any feature vector \( f \).

The importance of this algorithm is the distribution vector \( D_i \) that is updated at each iteration. After each iteration, more weight is applied to the samples that had more error therefore focusing the algorithm to train on the more difficult observations.

### 4.1.4 Physics Model for Fog

Given a foggy color pixel \( \tilde{x}_i \in \mathbb{R}^3 \) at location \( i \), the physics based atmospheric dichromatic model is \([44, 93]\)

\[ \tilde{x}_i = t_i x_i + (1 - t_i) a, \tag{4.9} \]

where \( a \in \mathbb{R}^3 \) is the airlight, \( t \in \mathbb{R} \) is the transmission and \( x \in \mathbb{R}^3 \) is the clear-day pixel color. The goal in fog removal is to estimate the clear-day image \( x \).

A chromatic framework was introduced by Narasimhan and Nayar \([93]\) to provide geometric constraints on images with varying levels of atmospheric conditions using (4.9). In order to be valid, this framework requires multiple images of a scene with different levels of fog in order to estimate the clear-day scene color \( x \).

Single image fog removal methods attempt to recover \( \tilde{x} \) by estimating airlight \( a \) and transmission \( t \) from a single foggy-day image. The Color Ellipsoid Framework introduced by Gibson and Nguyen \([86]\) provides a framework for estimating transmission \( t \) from a single image by observing patches from a scene. If pixels from an image are sampled within a window \( \Omega \), a color ellipsoid would be
observed within an RGB color cube. There are two main properties from the color ellipsoids in foggy-day images that are utilized for estimating the transmission: (1) The centroid of the ellipsoid and (2) the size of the ellipsoid shape are dependent on the transmission in foggy-day scenes. In summary, if the color ellipsoid is dark (small centroid) and is large (high variance) then it mostly likely does not have fog. Gibson and Nguyen in [86] demonstrate how these properties are used in multiple single image fog removal methods. These properties are in principle the same properties used by Jobson et al. [75] to characterize visually pleasing images.

In this work, we develop sub-metrics based on the geometry in (4.9) and the color ellipsoid properties in order to improve the CEM performance.

4.2 Training Data

Before we propose a new method in the development of the no-reference CEM, we will discuss how the human observations were obtained, what data was used, and the reliability of the data. In order to acquire a wide range of observers numbering above 30 we provided the test through an online web-page. This approach, however, prevented the ability to capture the exact physical test environment. The design of the web-page required a monitor to meet or exceed 700 × 650 resolution and color blind tests verified proper color monitor settings were used during the subjective testing. This “crowd-sourcing” approach is the same approach for data collection by Caicedo et al. [83].

We use the same image data provided by [53]. (We refer the readers to http://videoprocessing.ucsd.edu/~kgibson to obtain the images used for this work.) The subjective data was obtained from 35 different observers. Each observer was presented with 448 image pairs oriented side-by-side. The observer was required to give an opinion score as follows:

1. Left image has much better contrast than the right image
2. Left image has slightly better contrast than the right image
3. Left and right images have same level of contrast
4. Right image has slightly better contrast than the left image

5. Right image has much better contrast than the left image

\subsection{Demographics}

A survey at the end of the subjective testing asked the observers questions in order to obtain demographic information. (The demographic information on the test subjects is located at \url{http://videoprocessing.ucsd.edu/~kgibson}.) In summary, there was an even split among the observers regarding expertise in image enhancements ranging from (a) no experience then (b) basic experience and finally (c) intermediate experience. About half of the observers have never used image enhancements while the rest of the test subjects have at least used an image enhancement tool a few times a year.

\subsection{Images}

The images presented to the observers were color 320 \times 240 in resolution. The type of images were all of ships in an ocean environment. Each original image—which had fog or haze—was manipulated with 7 different methods by He \textit{et al.} \cite{he}, Chen \textit{et al.} \cite{chen}, Gibson \textit{et al.} \cite{gibson}, Tarel and Hautière \cite{tarel} and with Histogram Equalization \cite{hist_eq}. In addition, a degraded contrast method was applied with \(\tilde{x}_i = cx_i + b\) where the global contrast level is \(c = 0.7\) and global brightness is \(b = 0.3\). The original image at pixel \(i\) is \(x_i\) and the degraded contrast image at pixel \(i\) is \(\tilde{x}_i\).

With 8 images for each scene (including the original image) there were 28 unique pairs presented to the observer. This totals to 448 different image pairs. The average amount of time a test subject needed to complete the test was 40 minutes. The test subjects were allowed to log off and log back into the test at a later time in order to provide rest breaks.
Figure 4.2: Subjective Data. (a) Sorted MOS values with confidence intervals. (b) Histogram of all MOS values.

4.2.3 Data Screening

The data screening process followed the specification from the ITU-R BT.500-13 recommendation [94] where the MOS, confidence interval and observer rejection criteria are given. The mean opinion scores for $A$ observers were computed with

$$\bar{u}_k = \frac{1}{A} \sum_{i=1}^{A} u_{ki},$$

where $u_i$ is the score from the $i^{th}$ observer from the $k^{th}$ image comparison. The standard deviation for each image pair was estimated with

$$S_k = \sqrt{\frac{A}{A-1} \sum_{i=1}^{A} (u_{ik} - \bar{u}_k)^2}.$$  

The 95% confidence intervals are then computed with

$$[\bar{u}_k - \delta_k, \bar{u}_k + \delta_k],$$

with $\delta_k = 1.96 \frac{S_k}{\sqrt{A}}$.  

The value of 1.96 ensures the computed average has a probability of 0.95 of being within the confidence intervals [95].
The rejection process according to ITU Recommendation [94] was applied once and one observer was rejected. The rejection was reasonable since the observer selected the opposite values for each image comparison compared to the majority, e.g., if the majority selected 5 the rejected observer selected 1.

An illustration of the averaged scores and its distribution is in Fig. 4.2. The average confidence interval after data screening was 0.257 where the 95% probability region is indicated by red dashed lines in Fig. 4.2(a). The histogram of the MOS values in Fig. 4.2(b) clearly indicate that MOS of 1 and 5 is not present. This shows that the observers had a “psychological reluctance to make extreme judgements” [94]. Even though the MOS values do not cover the full range of possible scores—we choose not to rescale the data. This choice is addressed in the next section where the data is thresholded to accommodate the CEM which is a binary classifier.

4.3 Proposed Method

4.3.1 Thresholded MOS

We discussed in Section 4.2 that the observers were allowed to make a selection between 1 and 5 to rate the level of contrast between two images. The histogram of MOS values in Fig. 4.2(b) demonstrate our data suffers from the subjects’ unwillingness to make extreme judgments. Effectively, the MOS values span the range [2, 4] instead of [1, 5]. This is a common problem and is addressed in the ITU recommendation [94], where the approach to fix the MOS boundary effects is to apply a non-linear scale transformation.

The scaled MOS approach would require the CEM to be trained with 5 classes. In order to properly train a multi-class CEM, there should be a sufficient amount of data for each class. Obtaining enough training data from user observations becomes difficult with the number classes. The performance will suffer from the lack of training data. The performance for a classifier who’s output is $H(f)$ on
examples not used for training, the generalization error [96] is

\[ \hat{P}_r[H(f) \neq y] + \tilde{O} \left( \sqrt{\frac{T v_c}{K}} \right), \]  

(4.14)

where the number of examples used for training is \( K \), the number of training iterations is \( T \) and the Vapnik-Chervonenkis dimension (VC) \( v_c \) is smaller than \( K \) [96, 92]. The number of required classes is dependent on the VC number and is higher for more complex dimensions. The empirical probability on the training data is \( \hat{P}_r[\cdot] \) and \( \tilde{O}(\cdot) \) informally is used to hide all logarithmic and constant factors. The generalization error is inversely proportional to \( K \), the number of unique training examples. In order to implement a multi-class implementation (increased \( v_c \)), multiple binary classifiers must be created [92] which increases the error with a fixed set of examples \( K \).

We take a simpler approach to account for the generalization error and limited range of MOS values by decreasing the number of classes to only two. The CEM is a binary classifier. Given two images, the CEM would choose which image is best.

This approach still works well for cases when images are similar because there would be no error in selecting either image as the best since they are both the same. One of the significant differences between this work and previous work in [53], is that the training examples within the range

\[ [3 - \bar{\delta}, 3 + \bar{\delta}], \]  

(4.15)

are ignored where the average confidence interval is

\[ \bar{\delta} = \sum_{k=1}^{K} \delta_k. \]  

(4.16)

The reasoning is that if the overall MOS for an image pair was close to 3, then it did not matter which image was chosen to be the better image. Therefore we discard image pairs that are 95% probable of having MOS to be 3 in order to acquire a realistic measure of performance.

Another reason for discarding scores near 3 is to improve the AdaBoost algorithm. AdaBoost focuses on the more difficult examples presented to it during
the training phase. The drawback is that AdaBoost is susceptible to noise or misclassified examples [85]. Providing the algorithm with several image pairs of score 3 would cause AdaBoost to focus on only those examples because of the difficulty to classify them which is not the desired behavior. We want AdaBoost to focus on image pairs that are different in contrast in order to provide a properly trained model.

The scores are averaged together with outliers removed (rejected observers) to create a MOS for each train and test example. For each image pair example, $k$, the MOS ($\bar{u}_k$) is compared to a threshold to create the binary class $y$,

$$y_k = \begin{cases} +1, & \text{for } \bar{u}_k > 3 \\ -1, & \text{otherwise} \end{cases}$$

(4.17)

For training and testing, we ignore $y_k : 3 - \delta < y_k < 3 + \delta$ in order to select unique image pairs useful for training with the AdaBoost algorithm.

### 4.3.2 CEM Sub-Metrics

For each image pair there exists a subjective measurement (MOS) contained in the feature pool. The images are then measured with multiple sub-metrics and stored in the feature pool matrix.

Theoretically, there is no limit to the number of sub-metric choices for training the CEM. Additionally, there is no closed form solution that defines how many sub-metrics to choose and which ones to select for an optimized CEM. Thus to have a definitive answer to how exactly the CEM should be developed is non-trivial.

A more tractable approach is to improve on another method. Our goal in this work is to present an improved CEM compared to the CEM developed in [53]. We achieve this by developing sub-metrics that are based on the atmospheric dichromatic model (4.9) and the Color Ellipsoid Framework since the images that are measured for contrast are potentially affected by fog. We also parameterize a few sub-metrics in order to provide more unique observations to the learning algorithm. Finally, we remove trivial image pairs from the training set using the MOS
Figure 4.3: Example of DCP. Two 320 × 240 size images are measured with the DCP with window size $|\Omega| = 15 \times 15$, (a) and (c) are the original and enhanced images, respectively. The DCP applied to (a) and (c) are in (b) and (d), respectively. The cumulative distribution function (CDF) of (b) and (d) are in (e). Note that the CDF of the DCP of the enhanced image has a higher CDF at lower pixel intensities (below 20% of maximum intensity).

confidence intervals in order to provide the learning algorithm with discriminative examples.

In the following sections, we enumerate and describe the sub-metrics used for developing the CEM that are based on the HVS. We then follow with sub-metrics based on the Color Ellipsoid Framework.

HVS Based Sub-Metrics

The following sub-metrics are categorized as *HVS Based* if they inherently correlate well with human perception or if they are inspired by the HVS.
\( M_1 \). \textbf{Average Gradient Norm}

The \( M_1 \) sub-metric is similar to the Tenengrad criterion developed by [69] which was introduced in Equation (4.1). It is possible that a high value from \( M_1 \) may not indicate a high preference from humans similar to the findings by Choudhury and Medioni [82]. To account for this issue, we propose to measure \( M_1 \) with different image scales in addition to aggregating measurements from other sub-metrics. Note that \( M_1 \) only measures neighboring pixels because the basis Sobel kernel is a 3 \( \times \) 3 matrix. To provide more observations to the learning algorithm we choose to measure intensity differences over larger pixel distances by down-sampling the image but using the same 3 \( \times \) 3 Sobel kernels. Instead of measuring only the original scaled version of the image as in [53], we also apply \( M_1 \) to scaled down versions of the images. Based on empirical results from testing, we measure the \( M_1 \) with scales 1, 1/4, and 1/8. For an image \( x \) scaled down by \( L \), the sub-metric of this scaled image would then be defined as \( M_1(x, L) \). With different scales measured, the Sobel operator is able to measure pixel intensity differences over different scale spaces. We define this sub-metric to be

\[
M_1(x, L) = \frac{1}{N} \sum_{i=1}^{N} \left( \sqrt{G_h(i, L)^2 + G_v(i, L)^2} \right),
\]

(4.18)

where \( G_h \) and \( G_v \) are the horizontal and vertical Sobel kernels, respectively and the scale of the image is \( L \). The number of pixels in the image is \( N \).

\( M_2 \). \textbf{Munteanu and Rosa Metric}

\( M_2 \) is the enhancement metric developed by Munteanu and Rosa in [71] (discussed in beginning of Chapter 4). This metric uses an HVS inspired entropic measure and a modified Tenengrad measure. \( M_2 \) applied to image \( x \) is \( M_2(x) \). For the sake of space in this chapter, we refer the reader to [71] for details on the algorithm.

\( M_3 \). \textbf{Rizzi-Simone-Cordone Metric}

The multi-level framework metric by Simone et al. [72] (discussed in beginning of Chapter 4) that uses the HVS inspired DOG as the main contrast measure is called the Rizzi-Simone-Cordone (RSC) metric. We refer the reader to [72] for a
detailed description of the implementation of RSC. One of the parameters for the RSC is the number of pyramid levels. Each level is the image down-sampled by a factor of 2. The RSC applied to image $x$ with $L$ levels is $RSC(x, L)$. Likewise, the sub-metric $M_3$ is parameterized by the level number $L$,

$$M_3(x, L) = RSC(x, L).$$  \hspace{1cm} (4.19)

**Physics Based Sub-Metrics**

The following sub-metrics are based on the atmospheric dichromatic model (4.9) and the Color Ellipsoid Framework [86].

$M_4$. *Probability of Fog in Image*

The development of the Dark Channel Prior (DCP) by He *et al.* [4] was inspired by a statistical observation of clear-day natural scene images. The DCP is a morphological operator,

$$\theta_{D,i}(x) = \min_{j \in \Omega_i(x)} \min_{c \in \{r,g,b\}} x_j(c),$$ \hspace{1cm} (4.20)

with $i$ as the pixel location, $c$ is the color value from an RGB image, and the window at $i$ is $\Omega_i$.

He *et al.* [4] observed that the DCP of non-foggy natural scenes had the majority (about 90%) of the pixels less than 10% of the maximum value of the pixel intensities. Applying a DCP to a clear-day image produces a dark image.

In our work, we investigate scenes with ocean water and boats which are not of the same class of images that He *et al.* investigated so the statistical nature of the DCP is not the same. The same observation, however, is made with foggy and clear-day images of marine type scenes where the DCP of the clear-day scene is darker.

An example of applying the DCP to an ocean scene is in Figure 4.3. In Figs. 4.3(a), and 4.3(c) there are two images, the original image and a defogged image. The DCP is applied to both Figs. 4.3(a) and 4.3(c) and the cumulative distribution function of their DCPs are in Fig. 4.3(e). Although only 30% of the pixel intensities are less than 10% of maximum intensity for the defogged DCP,
there are zero intensities at that value for the foggy image. The number of pixel intensities at darker values increases when the fog is removed in ocean scenes as well. We observe that there still exists the relationship of a darker DCP when the fog is not present for ocean scenes.

We introduced the relationship of the DCP to the transmission using the Color Ellipsoid Framework in [55]. We show that the DCP is dependent on the position, size, and orientation of the color ellipsoid within sample window \( \Omega_i \). The color ellipsoid from sample \( \Omega \) has properties influenced by the transmission. The DCP therefore is a rough approximation of the transmission or amount of fog present in the image. The darker the DCP becomes, the higher the transmission value where the transmission approximation is

\[
t_D(x) = 1 - w\theta_D(x),
\]

with \( w = 0.95 \) used for visual preference. We set \( w = 1 \) since visual preference is not required for our purposes.

We design a sub-metric that will increase in value when fog is decreased in the image. This is essentially an estimate on the probability of fog being present in the image. Therefore we want this metric to be

\[
M_4(x, a) \sim 1 - \Pr[t(x) \leq a].
\]

If no fog is present in an image, theoretically \( \Pr[t(x) \leq a] = 0 \ \forall a \neq 1 \) which would then produce \( M_4 = 1 \ \forall a \neq 1 \). Likewise, if the entire image is pure fog, then \( t_D = 0 \ \forall a \) which would produce \( M_4 = 0 \). We design \( M_4 \) to estimate the probability measure by using (4.21),

\[
1 - \Pr[t(x) \leq a] \approx 1 - \Pr[t_D(x) \leq a]
= \Pr[1 - \theta_D(x) > a]
= \Pr[\theta_D < 1 - a]
= \Pr[\theta_D < b],
\]

for any value \( b = 1 - a \) and \( w = 1 \). Therefore the sub-metric \( M_4 \) is constructed to empirically measure the global average probability of fog being present with (\( N \) is...
Figure 4.4: Example of sub-metric $M_5$ which uses the color ellipsoid centroids. (a) Geometric interpretation of the vector displacements illustrated on the Red-Green color plane without loss of generality. (b) $M_5$ applied to a foggy image without averaging. With averaging, $M_5 = 0.32$. (c) $M_5$ applied to enhanced image without averaging. With averaging, $M_5 = 0.90$.

In our previous work [53], the value measured was $b = 1/32$. In this project, we additionally measure value $b = 1/2$.

$M_5$. Color Ellipsoid Shift

The next metric is based on the atmospheric dichromatic model in (4.9) where the foggy pixels exist within the convex set defined by the displacement of the original color pixel $x$ and the airlight color $a$.

For a lucid example, consider two images $x$ and $\tilde{x}$. The centroids at pixel $i$ are $\mu_i$ and $\tilde{\mu}_i$, respectively. If fog is indeed removed in $x$, then the centroids, $\tilde{\mu}$, will shift along the vector $a - x$. Observe the geometric example in Fig. 4.4(a). The airlight vector is denoted as $a$, the foggy centroid at an arbitrary pixel location is denoted as $\tilde{\mu}$ and the non-foggy centroid is $\mu$ for the same pixel location. The magnitude of $a - \mu$ is larger than $a - \tilde{\mu}$. Formally, this metric is defined as

$$M_5(x) = \frac{1}{N} \sum_{i=1}^{N} ||a - \mu_i||.$$  

(4.25)
The intuition behind this measure is that if an image is a foggy or defogged image, then the color ellipsoid centroids will shift along the convex set subtended by the airlight and the non-foggy color ellipsoid center, $a - \mu$. Therefore this metric should exhibit a larger value for a non-foggy image compared to a foggy image. This is evident in Figs. 4.4(b) and 4.4(c). For this illustration, the averaging was omitted in order to show the result of measuring the magnitudes. Note the foggy image has much smaller displacements and results in a darker result compared to the defogged version which is brighter (higher magnitude) in most areas of the scene.

$M_6$. Color Ellipsoid Size Metric

Another property of the color ellipsoid that is dependent on the amount of fog present is the size of the ellipsoid. The size is smaller when more fog is present. This property is used in various forms in [1], [4], and [86] to estimate the transmission. Therefore the goal of this metric, $M_6$, is to measure the average size of the color ellipsoids within an image. This approach was observed by Jobson et al. [75]. In their work, they found that “visually pleasing” images exhibited high average value and high average standard deviation. Measuring local average and variances with $8 \times 8$ or $4 \times 4$ image blocks was also a tool used for estimating additive and multiplicative noise by Lee and Hoppel [97].

We use a similar implementation of measuring the average standard deviation of an image projected into grayscale which assumes the airlight is colorless. This metric, in essence, measures the size of the color ellipsoid. The metric first divides an image into $B$ non-overlapped blocks. The image is padded symmetrically to prevent partial block sizes. The block sizes can be chosen to be between $8 \times 8$ to $30 \times 30$ for image sizes of $320 \times 240$. For testing, we chose $26 \times 26$. The reasoning for this choice is in the following Section 4.4. For each block, the standard deviation is measured and the overall value is an average of all the standard deviations. The $M_6$ is defined as

$$M_6(x) = \frac{1}{B} \sum_{b=1}^{B} \sigma_b$$

(4.26)
where $\sigma_b$ is the standard deviation of pixels within a $26 \times 26$ block non-overlapped block within image $x$.

### 4.3.3 Sub-Metric Performance

Before we discuss using the sub-metrics for populating a feature pool matrix for learning a perceptually based CEM metric, let us first investigate the performance of each sub-metric.

We define the performance measurement as follows. Given the sub-metrics in the previous section, $M_1, \ldots, M_6$, each sub-metric is used to compare an image pair indexed by $k$, $(x_{0,k}, x_{1,k})$ and to acquire a binary class estimation, $\hat{y}_k$. Specifically, given an image pair $x_{0,k}$ and $x_{1,k}$ and sub-metric $M$, the class estimate is

$$\hat{y}_{M,k} = \begin{cases} +1, & \text{for } M(x_{0,k}) > M(x_{1,k}) \\ -1, & \text{else} \end{cases}$$

(4.27)

If the sub-metric comparison gives a positive estimate, $\hat{y}_k = 1$, then it is estimated that $x_{0,k}$ is better in contrast than image $x_{1,k}$. Likewise, if $\hat{y}_k$ is negative, then it is estimated that image $x_{1,k}$ is better in contrast than image $x_{0,k}$.

For each image pair, there exists an MOS score that indicates which image is believed to have better contrast and was previously defined in (4.17). The average error for each sub-metric is then

$$e_M = \Pr_M[\hat{y} \neq y] \simeq \frac{1}{2K} \sum_{k=1}^{K} |\hat{y}_{M,k} - y_k|.$$  

(4.28)

The factor of $1/2$ is needed in the right hand side of Eq. (4.28) because if $\hat{y}_{M,k} \neq y_k$ then $|\hat{y}_{M,k} - y_k| = 2$.

The average error and standard deviation for each sub-metric using (4.28) is summarized in Table 4.1. Overall, each metric has an error below 25%. (As mentioned in the previous section, examples with MOS within the confidence interval of 3 are discarded.) Each metric performs better than chance in matching human comparisons. We can improve upon this by leveraging the performance of each sub-metric by using AdaBoost.
Table 4.1: Sub-Metric Errors

<table>
<thead>
<tr>
<th>Index</th>
<th>Sub-Metric</th>
<th>Average Error</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$M_1(x, 1)$</td>
<td>0.178</td>
<td>0.383</td>
</tr>
<tr>
<td>2</td>
<td>$M_1(x, 1/4)$</td>
<td>0.167</td>
<td>0.374</td>
</tr>
<tr>
<td>3</td>
<td>$M_1(x, 1/8)$</td>
<td>0.151</td>
<td>0.358</td>
</tr>
<tr>
<td>4</td>
<td>$M_2(x)$</td>
<td>0.175</td>
<td>0.381</td>
</tr>
<tr>
<td>5</td>
<td>$M_3(x, 3)$</td>
<td>0.170</td>
<td>0.376</td>
</tr>
<tr>
<td>6</td>
<td>$M_3(x, 6)$</td>
<td>0.156</td>
<td>0.364</td>
</tr>
<tr>
<td>7</td>
<td>$M_3(x, 12)$</td>
<td>0.159</td>
<td>0.366</td>
</tr>
<tr>
<td>8</td>
<td>$M_4(x, 1/32)$</td>
<td>0.194</td>
<td>0.396</td>
</tr>
<tr>
<td>9</td>
<td>$M_4(x, 1/2)$</td>
<td>0.243</td>
<td>0.429</td>
</tr>
<tr>
<td>10</td>
<td>$M_5(x)$</td>
<td>0.173</td>
<td>0.378</td>
</tr>
<tr>
<td>11</td>
<td>$M_6(x)$</td>
<td>0.159</td>
<td>0.366</td>
</tr>
</tbody>
</table>
4.3.4 Learning Based Contrast Enhancement Metric

In this section, we develop the contrast enhancement metric based on the AdaBoost learning framework and the new sub-metrics discussed in the previous sections.

The data used for training is a collection of feature examples

\[
v(x_1)^T, \ldots, v(x_k)^T,
\]

where \(v(x_k) \in \mathbb{R}^L\) is the feature vector containing the results from the sub-metrics measured from the \(k^{th}\) image example \(x_k\). (Here we denote superscript \(T\) as the transpose.) We use the label, \(y_k\) as the MOS, \(y_k = \{-1, +1\}\).

The \(L\) sub-metrics are utilized as difference values for both the training and testing examples. Thus for the \(k^{th}\) example, the \((L + 1) \times 1\) labeled feature vector (for training) is

\[
f(x_k)^T = (v(x_{0,k})^T - v(x_{1,k})^T, y_k),
\]

where \((x_{0,k}, x_{1,k})\) is the image pair from the \(k^{th}\) example set \(x_k\). The label \(y_k\) in (4.29) is removed for testing or applying the CEM to new image pairs. The feature vector we use in this chapter is an \(11 \times 1\) vector that contains both HVS, atmospheric dichromatic model, and Color Ellipsoid Framework (CEF) based sub-metrics. We denote this feature vector with subscript \(hc\):

\[
v_{hc}^T(x) =
(M_1(x, 1), M_1(x, 1/4), M_1(x, 1/8), M_2(x), M_3(x, 3), M_3(x, 6), M_3(x, 12), M_4(x, 1/32), M_5(x), M_6(x)),
\]

where the \((0,k)\) and \((1,k)\) subscripts are dropped for clarity and \(M_j\) is the \(j^{th}\) metric enumerated in Section 4.3.2. We selected three different scales for \(M_1\) (1, 1/4, and 1/8), three different levels for \(M_3\) (3, 6, and 12), and two different bin values for \(M_4\) (1/32 and 1/2) to give more degrees of freedom for learning. The
HVS only feature vector denoted with subscript $h$ is
\[
\mathbf{v}_h^T(x) = (M_1(x, 1), M_1(x, 1/4), M_1(x, 1/8), M_2(x), M_3(x, 3), M_3(x, 6), M_3(x, 12)),
\]
and the CEF based feature vector denoted with subscript $c$ is
\[
\mathbf{v}_c^T(x) = (M_4(x, 1/32), M_4(x, 1/2), M_5(x), M_6(x)).
\]

The concatenation of both vectors is the proposed new feature vector
\[
\mathbf{v}_{hc}^T(x) = (\mathbf{v}_h^T(x), \mathbf{v}_c^T(x)).
\]

We include in this chapter a comparison to our previous work using the previous feature vector denoted with subscript $p$, $f_p$, and refer the reader to [53] for details.

The performance of a learning based CEM is dependent on the type of feature vector used for training. The AdaBoost algorithm generates a binary classifier $H(.)$ in Eq. (4.8) where the learned weights and hypothesis chosen for each iteration strictly depend on the input feature vector. Therefore we associate a CEM with each feature dependent binary classifier by denoting it with a subscript that is associated with the type of feature vector,
\[
CEM_p = H(f_p), \quad CEM_{hc} = H(f_{hc}), \quad CEM_c = H(f_c), \quad CEM_h = H(f_h),
\]
with the previous CEM is $CEM_p$, our proposed CEM is $CEM_{hc}$, the HVS based CEM is $CEM_h$ and the Color Ellipsoid Framework based CEM is $CEM_c$.

### 4.3.5 Difference from Previous Work

There are a few metrics in this work that are the same metrics used from our previous work in [53]. They are the HVS based metrics $M_1(x, 3)$ and $M_2(x, 3)$ and the CEF based metric $M_4(x, 1/32)$. All other metrics we present in this chapter are new.
4.3.6 Implementation

We return to our main application of automating the choice of the best contrast enhancement filter. We choose to develop a metric that will provide feedback needed to select the best filter. With $\gamma$ filters in a filter suite, there would need to be $\Gamma = \binom{\gamma}{2}$ image comparison measurements made by the CEM. From those $\Gamma$ comparisons, the image with the highest vote is chosen. For each $k^{th}$ image pair (indexed by $k_0$ and $k_1$) created by applying two different contrast enhancements, a feature vector $f$ is obtained. A contrast enhancement vote vector $r \in \mathbb{R}^\gamma$ is created where the vote for each image is updated with

$$
    r(k_0) = r(k_0) + 1, \quad \text{for} \quad H(f) < 0
$$
$$
    r(k_1) = r(k_1) + 1, \quad \text{otherwise}.
$$

(4.34)

After the $\Gamma$ comparisons, the best filter is chosen whose index is associated with $\hat{k}$ where $\hat{k}$ is the index associated with the highest value (most votes) in $r$,

$$
    \hat{k} = \arg\max_k \tau(k).
$$

(4.35)

4.4 Results and Discussion

In this section, we provide the results from training the $CEM_{hc}$. The work in this chapter is compared to our previous method, $CEM_p$, in [53]. We also investigate the performance when using only HVS based sub-metrics ($M_1 - M_3$), $CEM_h$, and with only Color Ellipsoid Framework (CEF) sub-metrics ($M_4 - M_6$), $CEM_c$.

The only parameter required for the AdaBoost algorithm is the number of training iterations, $T$. This can also be considered the number of weak classifiers to boost. For all data sets, we used $T = 80$.

4.4.1 Averaged Classification Error

A 2-fold cross-validation method is used where half of the data is selected for training and the other half is selected for testing. For each 2-fold cross-validation
Figure 4.5: ROC curves for $CEM_p$, $CEM_h$, $CEM_c$, and $CEM_{hc}$. (a) The test and training ROC curves for $CEM_p$ and $CEM_{hc}$. (b) A comparison of test curves with all methods. The Area Under the Curve (AUC) values are included.

step, a random set is chosen for training and the other half is used for testing. There is never overlap between test data and training data. Each 2-fold cross-validation is applied for 1000 iterations and the results are averaged. The data set consisted of $K = 448$ examples with 224 used for testing and 224 used for training.

The average classification errors for testing were calculated with Eq. (4.28). In summary, the test error (average, standard deviation) after 1000 cross-validation iterations are

$$CEM_p: \ (12.8\%, \ 0.03), \quad CEM_h: \ (12.3\%, \ 0.02),$$
$$CEM_c: \ (11.8\%, \ 0.02), \quad and \ CEM_{hc}: \ (10.4\%, \ 0.02).$$

These averaged errors show that using all CEM methods outperform any non-learning based sub-metric from Table 4.1. Note that the samples with $3 - \delta \leq \text{MOS} < 3 + \delta$ are ignored for a more realistic comparison of performance with $\delta$ defined in Eq. (4.16). We have found that using the $CEM_{hc}$ method provides a 90% chance of selecting the image with best contrast with respect to human opinion.

We also present the average Pearson correlation over all the cross-validation iterations. The correlation was measured between the 224 estimated outputs $\hat{y}$ and
the known threshold MOS values applied to threshold $y$ for each cross-validation step. For each method, the correlation between the estimates from the CEM and the subjective values (MOS) are

\[
CEM_p : 0.75, \quad CEM_h : 0.76,
CEM_c : 0.77, \quad \text{and} \quad CEM_{hc} : 0.80.
\]

The proposed $CEM_{hc}$ method has the highest correlation on average over 1000 cross-validation iterations followed by $CEM_c$, $CEM_h$ and then $CEM_p$ in decreasing order.

Both the average errors and average Pearson correlation measures agree with the proposed $CEM_{hc}$ method being the better performing method with lowest error rate and highest correlation with human observations.

### 4.4.2 Receiver Operator Characteristic Curves

The Receiver Operator Characteristic (ROC) curve is a useful tool for visualizing the training and testing performance from the CEM variations. In this work, we use the confidence measures from the AdaBoost algorithm to generate the ROC curves. The confidence measure, $\alpha_i h_i(f)$, [85] is the estimation of the class before it is applied a threshold. We can rewrite Eq. (4.8) by using a threshold $th$,

\[
H(f, th) = \begin{cases} 
+1, & \text{if } \sum_{i=1}^{T} \alpha_i h_i(f) \geq th \\
-1, & \text{otherwise.} 
\end{cases} \tag{4.36}
\]

The confidence is high when $|\sum_{i=1}^{T} \alpha_i h_i(f)|$ is high. Eq. (4.36) and Eq. (4.8) are equivalent when $th = 0$. The true positive set, $TP$, and the false positive set, $FP$, are

\[
TP(th) \in \{H(f_k, th) = +1 \cap y_k = +1 \ \forall k\}, \tag{4.37}
\]

and

\[
FP(th) \in \{H(f_k, th) = -1 \cap y_k = +1 \ \forall k\}. \tag{4.38}
\]

The number of true positives and false positives are $|TP(th)|$ and $|FP(th)|$ respectively.
At each 2-fold cross-validation iteration, $|TP(th)|$ and $|FP(th)|$ are measured for each threshold $th$. After all cross-validations have been completed, these sets of values are averaged together at each threshold value to produce a single set, $\overline{|TP(th)|}$ and $\overline{|FP(th)|}$. For $K$ training (or testing) examples, the TP and FP rates are $\overline{|TP(th)|}/K$, and $\overline{|FP(th)|}/K$ respectively. The ROC curves are constructed as FP vs. TP plots.

The goal of a classifier is to choose a threshold that minimizes the false positive rate and maximizes the true positive rate. On a plot where the TP axis is vertical and the FP axis is horizontal, improving the performance would push the bend of an ROC curve up and left. Therefore the Area Under the Curve (AUC) would increase where ideally AUC = 1.

The AdaBoost training and testing ROC curves are presented in Fig. 4.5(a). Both the test ROC curve and training ROC curve associated with our proposed method, $CEM_{hc}$, show the classifier has a higher TP rate compared to our previous method, $CEM_p$, for all false positive rates. Since the same number of training iterations was used at $T = 80$, this shows that AdaBoost was able to minimize the training error more effectively when using the feature vector $f_{hc}$. This indicates that the previous feature set, $f_p$, limited AdaBoost in its ability to train the classifier. The proposed feature vector, $f_{hc}$, enhanced AdaBoost’s ability to train on the image pairs.

A comparison of the test ROC curves for all CEM methods is shown in
Fig. 4.5(b). Each CEM was trained off a subset of the data and evaluated on the other half of the data, therefore these curves demonstrate each CEM’s ability to classify new data. From worst to best performance (low to high AUC) is the $CEM_c, CEM_p, CEM_h,$ and $CEM_{hc}$. The various curve positions demonstrate the importance of the choice of features. Since the $CEM_h$ uses a subset of features from $f_{hc}, CEM_h$ can be improved by including the physics based feature set, $f_c$.

4.4.3 Correlation of Contrast Enhancement Rankings

We revisit the main application of automatically selecting the image with the best contrast. For our test, we had 16 different scenes with 7 different contrast enhancements and the original image. Thus 8 images were ranked for each scene. Eq. (4.34) was used to create a vote vector for each scene and for each CEM method ($CEM_{hc}, CEM_h, CEM_c, CEM_p$). We also generated a vote vector based on the subjective MOS values $r_m \in \mathbb{R}^\gamma$,

$$
\begin{align*}
    r_m(k_0) &= r_m(k_0) + 1, \quad \text{for } \bar{u}_k < 3 \\
    r_m(k_1) &= r_m(k_1) + 1, \quad \text{otherwise},
\end{align*}
$$

(4.39)

where $k$ is the index for the image pair comparison and $k_0$ and $k_1$ indicate which images were compared for example $k$. For the size of the vote vector in this chapter, $\gamma = 8$ because there were 8 images for each scene.

We used a Leave-one-out cross-validation method for training the CEM for generating the vote vectors. The reasoning is that the performance of AdaBoost is dependent on the number of training examples according to Eq. (4.14). We choose to maximize the number of training examples while removing all examples associated with the scene being evaluated. For example, if Scene 0 is being evaluated, then Scenes 1 through 15 were used for training. This effectively shows the CEM’s ability to rank new data.

Each CEM method vote vector ($r_{hc}, r_h, r_c, r_p$) was compared to the MOS vote vector. A Pearson correlation (also used in [72]) and Kendall rank correlation [98] along with their p-values were measured for each vote vector pair. These correlations measure how well each method performs in rank ordering images according to their contrast with respect to subjective rank orderings.
For the sake of space and clarity, we only discuss the results from the $CEM_{hc}$ and $CEM_h$ methods because they produced the highest averaged correlations. (The details of the correlation results can be found at http://videoprocessing.ucsd.edu/~kgibson/cem.htm.)

On average, the $CEM_{hc}$ method is slightly more correlated to subjective data compared to the $CEM_h$ method with averaged correlations of 0.81 and 0.79 respectively.

We also measured the Kendall rank correlations for each CEM method. We have found empirically that a Kendall rank correlation near 0.7 is reasonably high where only 4 values in the vote vector may be incorrect. For example, the Kendall rank correlation between the vectors $(1, 2, 3, 4, 5, 6, 7, 8)^T$ and $(3, 2, 1, 4, 5, 6, 8, 7)^T$ is 0.71. The $CEM_{hc}$ method on average is more correlated than the $CEM_h$ method with averaged correlations of 0.72 and 0.67 respectively. According to each correlation measure, no method performs the best for every scene. On average our proposed $CEM_{hc}$ method is more correlated with the subjective rankings. Adding the physics based features to the HVS based CEM improves the rank correlation performance.

In Fig. 4.6, we present the Kendall rank performance of $CEM_{hc}$ with varying block sizes for $M_6$ (4.26). We measured the performance with block sizes ranging from $8 \times 8$ to $32 \times 32$. The performance is peaked at $26 \times 26$ and drops significantly above $30 \times 30$. Since the image ranking is compared to averaged human observations, there is not a sound reasoning for why Kendall ranking is highest for a block size of $26 \times 26$. The performance of $CEM_{hc}$ is reasonable for ranges $8 \times 8$ to $30 \times 30$ for image sizes of $320 \times 240$.

The p-values indicate the significance of the correlations for each scene. A p-value of 0.10 means there’s a 10% chance a random vector could achieve the same correlation. Thus the higher the p-value, the less significant the correlation measure. There were three different scenes that had high p-values for both methods. The CEM correlation values for these scenes were also very low (below 0.7). This indicates that the CEM had difficulty in ranking these scene types with respect to the averaged human rankings. This also indicates AdaBoost either doesn’t have
An image set was ranked according to contrast with $CEM_{hc}$. The results are in Fig. 4.7. The images were chosen to be diverse to demonstrate the robustness of the algorithm. The training set for $CEM_{hc}$ were from the same 16 different scenes with ships in fog with different levels of contrast (Scenes 0 through 15). The ranked order in Fig. 4.7 is reasonably accurate with respect to human preference.
4.5 Acknowledgment

The text of Chapter 4 is adapted from *A No-Reference Perceptual Based Contrast Enhancement Metric for Ocean Scenes in Fog*, Kristofor Gibson, Truong Nguyen, published in *IEEE Transactions on Image Processing* June of 2013. The dissertation author is the primary author of this publication.
“Since then I never pay attention to anything by “experts”. I calculate everything myself.” - Richard P. Feynman

Imaging systems positioned near the ocean often suffer in performance, perceptually and objectively, because of atmospheric turbulence, fog, sun-glare, camera motion from wind buffeting and many other adverse weather conditions. For long distance imaging, the most prominent are camera motion, fog, atmospheric turbulence and blur (from optics and atmosphere). The specific imaging environment we address in this work is an optical system that is observing targets of interest with the optical axis being parallel with the horizon (over-the-horizon viewing). The environment itself will have fog or haze, wind and heat that causes eddy currents which is observed as turbulence in an imaging system. For each wavelength (color), we represent the atmospheric effects in the imaging system with

\[ i_k(x) = h_k(x) * \left( t_k(x_k) o_k(x_k) + (1 - t_k(x_k))a \right) + w_k. \]  

The ideal image of the object without any atmospheric artifacts at frame \( k \) and pixel position \( x \in \mathbb{R}^2 \) is \( o_k(x) \in \mathbb{R} \). The observed image with the dominant artifacts of blur, turbulence, motion and fog is \( i_k(x) \in \mathbb{R} \). The contrast degradation from fog is represented with the terms transmission \( t(x) \in \mathbb{R} \) and airlight \( a \in \mathbb{R} \). The
airlight is the light imaged when the object at \( \mathbf{x} \) is infinitely far away from the imaging system. An example of \( a \) is the color and brightness observed of fog at the edge of the horizon. The amount of light decay over distance from the atmospheric scattering medium is spatially varying and depends on the distance from the sensor to the object \( d(\mathbf{x}) \in \mathbb{R} \) and scattering medium parameter \( \beta \in \mathbb{R} \),

\[
t_k(\mathbf{x}) = e^{-\beta d_k(\mathbf{x})}. \tag{5.2}
\]

The global blur from the atmosphere and optical system is modeled with the blur kernel \( h(\mathbf{x}) \). The pixel motion from turbulence and camera motion is

\[
\tilde{\mathbf{x}}_k = \Delta g, k + \Delta l, k(\mathbf{x}), \tag{5.3}
\]

with global motion represented with \( \Delta g, k \) and local motion from the object and turbulence is \( \Delta l, k(\mathbf{x}) \).

The model in (5.1) is a similar model used by Joshi and Cohen [31] except we model the blur, \( h(\mathbf{x}) \), independent of local motion \( \tilde{\mathbf{x}} \).

Our main goal in this work is to develop a joint turbulence mitigation and fog removal method that can recover the image of an object fast enough for near real-time performance. To accomplish this goal, we first develop an analysis in turbulence mitigation that also includes the fog model. We then propose a method that is based on the results from the analysis, performs well for most atmospheric conditions and is efficient for near-real time processing.

The performance of an optical system imaging through the atmosphere for short and long exposure settings has been sufficiently presented by Fried [99]. In [99], the Fried seeing parameter \( r_0 \) is used to describe the scale of the turbulence. If an optical system has a diameter smaller than \( r_0 \), then the turbulence is minimized. The modulation transfer function (MTF) of the optical system (including the atmosphere) degrades as the diameter increases above \( r_0 \). For our application, we assume the diameter of the optical system is larger than \( r_0 \) which is a very common scenario.

The structure constant, \( C_n^2 \), from Kolmogorov theory [100] is a parameter that “characterizes the strength of index-of-refraction fluctuations” [101]. The
structure constant is a function of altitude we assume is a constant in all horizontal viewing angles when an optical system is in an ocean environment.

In this work, we do not specifically provide an analysis for blur or methods to remove it. We assume the imaging system is under isoplanatic conditions (spatially invariant blur) because the optical system is positioned close to the heat source (turbulent generator) which is most common for ground based optics in astronomy and imaging systems positioned near the ocean [102].

Recovering the original image without knowledge of the blur, transmission, local motion, or noise statistics is a difficult process. Many methods have been proposed that account for some knowledge of the physics of the scene, however, no method exists that accounts for both turbulence and fog. The methods proposed by Anantrasirichai et al. [103] and Joshi et al. [31] do incorporate contrast enhancements but are only used as a post-processing solution. In this work we address how to incorporate information from the contrast enhancement process into the turbulence mitigation process.

5.1 Object Tracking in Low Contrast Analysis

Given an image of an object \( o(x) \) (for this analysis we consider this image grayscale) the global translation from frame \( k \) to frame \( k + 1 \) is characterized as

\[
o_{k+1}(x) = o_k(x - \Delta_g),
\]

with the frame number \( k \) and global translation vector as \( \Delta_g \) and position of the pixel at \( x \).

A common method for tracking a globally translating object is to obtain measurements from phase correlation. Phase correlation techniques take advantage of the shift property of Fourier transformed signals. We denote the Discrete Fourier Transform (DFT) of an image with

\[
\text{DFT} \{ o(x) \} = O(u) = \sum_x o(x) e^{-jx^T u},
\]

where the frequency vector is \( u \). We assume, without loss of generality, the spatial sampling period is 1. The global translation vector \( \Delta_g \) is extracted with the phase
correlation measure,

\[
\frac{O_k(u)O^*_k(u)}{|O_k(u)O^*_k(u)|} = e^{ju^T \Delta_g}. \tag{5.6}
\]

The complex conjugate of \( O(u) \) is written as \( O^*(u) \). The transpose of the vector \( u \) is \( u^T \). Taking the inverse DFT of Eq. (5.6) will ideally produce a delta peak at location \( \Delta_g \). Therefore, finding a vector \( \hat{\Delta}_g \in \mathbb{R}^2 \) that minimizes the cross-correlation cost function

\[
\sum_u \left| \frac{O_k(u)O^*_k(u)}{|O_k(u)O^*_k(u)|} - e^{ju^T \Delta_g} \right|^2, \tag{5.7}
\]
provides the global translation estimate.

The model in (5.4) is an ideal case where a single peak can be found in (5.6). In practice, the signal will contain noise which produces false peaks when taking the inverse DFT of (5.6). For this analysis, we consider the signal to include a zero mean, white noise signal, \( w(x) \), with variance \( \sigma_w^2 \), similar to the work of Manduchi and Mian [104],

\[
o_{k+1}(x) = o_k(x - \Delta_g) + w(x). \tag{5.8}
\]

We denote the true translation as \( \Delta_g \) and translation estimate as

\[
\hat{\Delta}_g : \arg \max_{\Delta} \text{DFT}^{-1}\left\{ \frac{O_k(u)O^*_k(u)}{|O_k(u)O^*_k(u)|} \right\}. \tag{5.9}
\]

The translation estimation error is the difference between the true translation and estimated translation,

\[
\Delta_e = \Delta_g - \hat{\Delta}_g. \tag{5.10}
\]

Based on the analysis in [104], if the magnitude of the DFT of the noise signal is significantly smaller than the object signal, \(|W(u)| \ll |O(u)|\), then the translation estimate error is approximately

\[
\Delta_e \approx \frac{\sum_u \phi_n(u)|O_k(u)|^2 u}{\sum_v |v|^2|O_k(v)|^2}, \tag{5.11}
\]
where the phase of the object at frame \( k + 1 \) is \( \phi_{k+1}(u) \) and phase noise is \( \phi_n(u) = \phi_{k+1}(u) + \Delta g^T u \). This approximation simply shows the performance is based on the noise power and amount of image information or bandwidth. (A more in depth analysis including image information can be found in Robinson and Milanfar [105].)

The variance of the error (5.11), similar to [104], is

\[
\sigma^2_{\Delta e} = \frac{\sum_u \sigma_{\phi_n}^2(u)|O_k(u)|^4|u|^2}{\left( \sum_v |v|^2|O_k(v)|^2 \right)^2},
\]

(5.12)

where we assume the phase noise is statistically uncorrelated for all frequency directions. Given Eqs. (5.11) and (5.12), we observe the error is influenced by the presence of phase noise from the next image frame, \( a_{k+1} \). Therefore with respect to the previous image, any phase change due to noise, local object motion, or atmospheric turbulence will increase the global translation error.

If we assume as in [104], that the phase noise is upper-bounded by the ratio of noise power and signal power,

\[
\phi_n(u) \leq \arcsin \left( \frac{|W(u)|}{|O_k(u)|} \right) \approx \frac{|W(u)|}{|O_k(u)|},
\]

(5.13)

for \( |W(u)| \ll |O_k(u)| \), then the translation error is upper-bounded by a function of the noise power and signal power,

\[
\Delta_e \leq \frac{\sum_u |W(u)||O_k(u)|u}{\sum_v |v|^2|O_k(v)|^2}.
\]

(5.14)

We now modify the model to include global contrast reduction with a contrast coefficient \( t \in [0, 1] \) and a brightness value \( a \) such that the contrast degraded object \( i(x) \) is

\[
i(x) = t o(x) + a + w(x)
\]

(5.15)

and DFT of (5.15) is

\[
I(u) = t O(u) + a \delta(u) + W(u).
\]

(5.16)

Using (5.14) and (5.16), the translation error including global contrast degradation is

\[
\Delta^t_e \leq \frac{\sum_u t|W(u)||O_k(u)|u}{\sum_v t^2|v|^2|O_k(v)|^2}
\]

(5.17)

\[
= t^{-1} \Delta_e.
\]

(5.18)
(a) The first two images from left to right, o₁ and o₂, respectively, are the images used for measuring PSR with various \( t \) and \( \sigma_w \) values. The last two (from left to right) are an example of the images with \( t = 0.4, \sigma_w = 0.1, \) and \( a = 0.3 \). (b) PSR of translation estimation with cross correlation method with varying \( t \) and \( \sigma_w \) values.

It is evident in (5.18) that the translation estimation error increases exponentially when the contrast factor \( t \) decreases. In a similar fashion, the variance of the translation estimator error is also influenced by the global contrast,

\[
\sigma^2_{\Delta t} \leq t^{-2} \sigma^2_{\Delta e}. \tag{5.19}
\]

Because \( t \in [0, 1] \), the upper-bound limits of the translation error and variance of translation error are increased exponentially as the contrast is decreased. According to (5.18) and (5.19), the presence of brightness, \( a \), does not influence the tracking error. In real applications, however, the brightness may cause pixel values to become saturated which could be modeled as signal dependent quantization noise in (5.15). In this work, we do not consider saturation as a significant noise artifact.

For image registration or global translation estimation, it is common to measure the Peak to Side-lobe Ratio (PSR) to measure the peak sharpness of a correlation based tracker [106, 107]. The higher the PSR, the better the estimation.
For our experiment, we translate an image and vary the noise variance $\sigma_w^2$ and contrast level $t$. An example of this is in Fig. 5.1(a). The first two images (from left to right) are the original image $o_1$ and translated image $o_2$, respectively. Their counterparts, $i_1$ and $i_2$, that include contrast degradation and noise are the 3rd and 4th images, respectively. We measure the PSR with

$$PSR = \frac{\text{peak} - \mu}{\sigma}, \quad (5.20)$$

where $\text{peak}$ is the value of the response from the inverse DFT of the cross-correlation in (5.9) located at $\Delta_g$. The values (excluding the peak at $\Delta_g$) within a $21 \times 21$ neighborhood centered at the true translation position are then used to measure the sample mean $\mu$ and standard deviation $\sigma$.

The results from varying contrast $t$ and noise $\sigma_w$ and with a constant translation $\Delta_g$ are in Fig. 5.1(b). The PSR was measured for each $t$ and $\sigma_w$ value in 100 trials. The mean and variance of the PSR values are in Fig. 5.1(b). The solid curve is the average PSR value and the errorbars are the standard deviation of PSR values. With a fixed contrast value $t$, increasing $\sigma_w$ decreases the PSR. With fixed noise $\sigma_w$, the PSR decreases exponentially with decreasing contrast $t$. The variance of PSR values increase as the global contrast $t$ is decreased. These observations satisfy our approximations in (5.18) and (5.19).

We now address whether applying a contrast enhancement before global image alignment (tracking) improves performance. We begin with modeling the contrast enhanced image with

$$\hat{o}(x) = \frac{i(x) - a}{t} = o(x) + n(x), \quad (5.21)$$

where $n(x) = w(x)/t$.

With Eqns. (5.18) and (5.21), we find that enhancing before tracking does not change the performance. With DFT $\{n(x)\} = N(u)$, the translation estimation error with contrast enhancement before tracking is

$$\Delta_e' \leq \frac{\sum_u |N(u)||O_k(u)||u|}{\sum_v |v|^2|O_k(v)|^2}$$

$$= \frac{\sum_u t^{-1}|W(u)||O_k(u)||u|}{\sum_v |v|^2|O_k(v)|^2}$$

$$= t^{-1}\Delta_e, \quad (5.24)$$
**Figure 5.2:** Simulation of object tracking in low contrast noisy images and noise reduced images. The dashed curves are PSR values from object tracking with images containing noise and contrast reduction. The solid curves are PSR values from object tracking with images with low contrast but with noise reduced using a locally adaptive Wiener filter. Each curve pair is associated with the same noise variance (before noise reduction).

which is the same translation error with fog present in (5.18). The reasoning is that when the contrast is enhanced, the noise variance is amplified with $n(x) = w(x)/t$ since $t \in [0,1]$. Therefore enhancing before tracking will not improve the tracking performance when image noise is present. We illustrate the improvement when noise is reduced in Fig. 5.1. The Locally Adaptive Wiener filter [108, 109] is used to reduce the image noise before object tracking and measuring the PSR. For $t > 0.4$, all of the methods are improved in PSR when the noise is reduced first. There is no improvement in PSR for $\sigma_w = 0.1$ and $t \leq 0.3$. This is because the signal and noise powers are not significantly different and may be limited by the ability of the Locally Adaptive Wiener filter. Additionally, the object tracking error model in (5.17) relies on the power of the noise to be significantly less than the power of signal noise. The error model is not valid because the requirement is not true at higher noise power and lower contrast values.
5.2 Defogging and Frame Averaging Analysis

It is common to average frames together in order to reduce the local motion induced by hot air turbulence in the atmosphere [110, 31, 111]. In this approach, there is an assumption the object does not exhibit rigid local motion and that there is the possibility of global translation (e.g., shaking camera). In [31], the number of frames selected for averaging is investigated and we reiterate in this work that choosing a large number of frames will reduce the noise variance, however, the image will become blurred due to inaccurate stabilization or local motion due to turbulence or object motion. Therefore it is best to choose the smallest number of frames as possible, however, large enough to suppress noise.

In this work, we investigate mitigating turbulence in a foggy scene which requires a model of fog. The common model used for imaging in foggy scenes is

\[ i(x, \lambda) = t(x, \lambda) o(x, \lambda) + (1 - t(x, \lambda)) a(\lambda) + w(x), \]  

(5.25)

where for each wavelength of light \( \lambda \), \( i(x, \lambda) \) is the foggy image at pixel location \( x \). The clear-day image of the object is \( o(x, \lambda) \). The transmission, \( t(x, \lambda) \), is an exponentially decaying function of wavelength, size of scattering particles and scene depth,

\[ t(x, \lambda) = e^{-\beta(\lambda)d(x)}. \]  

(5.26)

The amount of fog observed in an image is scene depth dependent \( d(x) \) and size of scattering particles \( \beta(\lambda) \). We also introduce a white noise signal \( w(x) \sim N(0, \sigma_w^2) \).

In this work we assume the scattering is spatially invariant and the scene depth is spatially varying. We simplify notation of (5.25) by dropping the wavelength parameter,

\[ i(x, \lambda) = i(x) = t(x) o(x) + (1 - t(x)) a + w(x), \]  

(5.27)

but still model the scattering as a function of wavelength. (Later in this work, we will relax this color dependence in order to develop a single image defogging method for color images.)
Observe that the general contrast degradation model in (5.15) and the foggy image model in (5.27) differ only in the structure of brightness where it is spatially varying in foggy scenes but globally constant in a general image. The object tracking analysis also applies to images with fog.

For the analysis in this section, we first determine which method is best; defog first and then average frames temporally or to average frames temporally and then defog the frames. After we develop a model for both approaches we will then investigate how the number of frames to average can be determined by the amount of fog present in an image.

Assume at this point that the transmission \( t(x) \) and airlight \( a \) are exactly known. A defogging algorithm, \( \text{defog}() \), would be simply an algebraic function

\[
\hat{o}_k(x) = \text{defog}(i_k(x)) = \frac{i_k(x) - a}{t_k(x)} + a
\]

where we reintroduce the frame number \( k \). Since the airlight is slowly varying over time [26], we do not annotate airlight with a frame number.

We annotate the temporal averaging function using \( K \) frames with \( \langle \cdot \rangle_K \) which is a sample average of a frame sequence,

\[
\langle f(x) \rangle_K = \frac{1}{K} \sum_{k=1}^{K} f_k(x + \Delta_{l,k}(x)),
\]

where the local motion from turbulence in the \( k^{th} \) frame is \( \Delta_{l,k}(x) \). The expectation and variance of (5.29) respectively are

\[
\mathbb{E}[\langle f(x) \rangle_K] = \mu_f(x),
\]

and

\[
\text{var}[\langle f(x) \rangle_K] = \frac{1}{K}\sigma_f^2(x).
\]

The variance in (5.31) captures the pixel value variation induced by the hot-air turbulence.

Our goal in this section is to analyze the difference between defogging a temporally averaged image \( \text{defog}(\langle i(x) \rangle_K) \) and a temporally averaged defogged image sequence \( \langle \text{defog}(i(x)) \rangle_K \).
If \( t \) and \( a \) were exactly known, then using (5.27) and (5.28), the defogged estimate has amplified noise,
\[
\hat{o}_k = o_k(x) + \frac{w_k(x)}{t_k(x)},
\]
(5.32)
because \( 0 \leq t \leq 1 \). Observe in (5.32) that suppressing noise by averaging frames temporally will require more frames as \( t \to 0 \). We want to understand if averaging frames before or after defogging makes a difference in performance. First, averaging a defogged image sequence is
\[
\langle \text{defog}(i(x)) \rangle_K = \langle o(x) \rangle_K + \frac{\langle w(x) \rangle}{\langle t(x) \rangle}. \tag{5.33}
\]
If the object within \( x \) is stationary then \( \langle o(x) \rangle_K = o(x) \) and transmission is statistically independent of noise then (5.33) becomes
\[
\langle \text{defog}(i(x)) \rangle_K = o(x) + \frac{\langle w(x) \rangle_K}{\langle t(x) \rangle_K}. \tag{5.34}
\]

The process of defogging an averaged sequence is modeled with
\[
\text{defog}(\langle i(x) \rangle_K) = \frac{\langle i(x) \rangle_K - a}{\hat{t}(x)} + a
\]
\[
= \frac{\langle t(x) o(x) \rangle_K}{\hat{t}(x)} + \frac{\langle t(x) \rangle_K}{\hat{t}(x)} a + a + \frac{\langle w(x) \rangle_K}{\hat{t}(x)}. \tag{5.35}
\]
In (5.35), the transmission used for defogging, \( \hat{t}(x) \), is distinguished from the true transmission, \( t(x) \). Assuming, however, object local motion and global motion are correctly compensated, then \( \hat{t}(x) = t(x) = \langle t(x) \rangle_K \) and (5.33) becomes
\[
\text{defog}(\langle i(x) \rangle_K) = \frac{\langle t(x) o(x) \rangle_K}{\langle t(x) \rangle_K} + \frac{\langle w(x) \rangle_K}{\langle t(x) \rangle_K}. \tag{5.36}
\]
In order for \( \text{defog}(\langle i(x) \rangle_K) \) to be equivalent to \( \langle \text{defog}(i(x)) \rangle_K \), the following must be true:
\[
\langle t(x) o(x) \rangle_K = \langle t(x) \rangle_K \langle o(x) \rangle_K. \tag{5.37}
\]
Eq. (5.37) can only be valid when the transmission and object radiance are statistically independent in the temporal domain. In general this is true, however, the statistical independence assumption is not true at object depth discontinuities in
both the spatial and temporal domain. The edge in the image is the same edge from the transmission.

Consider the case where the transmission is a scaled version of the object radiance,

\[ t(x) = qo(x), \]  

then the expected value of (5.36) becomes

\[ E[\text{defog}(\langle i(x) \rangle_K)] = E \left[ \frac{\langle qo^2(x) \rangle_K}{q\langle o(x) \rangle_K} \right] + E \left[ \frac{\langle w(x) \rangle_K}{q\langle o(x) \rangle_K} \right]. \]  

(5.39)

Using the approximation for the expected value of a ratio from [112, 113],

\[ E \left[ \frac{R}{S} \right] \approx \frac{E[R]}{E[S]} - \frac{\text{cov}(R, S)}{E[S]^2} + \frac{\text{var}[S]E[R]}{E[S]^3}, \]  

(5.40)

where the covariance measure is \( \text{cov}(\cdot, \cdot) \). The expected values of (5.34) and (5.36) (dropping \( (x) \) for clarity) respectively are

\[ E[\text{defog}(\langle i(x) \rangle_K)] \approx \mu_o, \]  

(5.41)

\[ E[\text{defog}(\langle i(x) \rangle_K)] \approx \frac{\mu_o^2 + \sigma_o^2}{\mu_o} - 2 \frac{\text{cov}(q\langle o^2(x) \rangle_K, q\langle o(x) \rangle_K)}{\mu_o^2} + \frac{q\sigma_o^2(\mu_o^2 + \sigma_o^2)}{\mu_o^3}, \]

\[ \geq \mu_o + \frac{\sigma_o^2}{\mu_o}. \]  

(5.42)

The simplifications in (5.41) and (5.42) are based on the statistical independence between \( w \) and \( t \) and that \( w \) has zero mean. Additionally, the simplification in (5.42) is due to the covariance between \( \langle o^2(x) \rangle_K \) and \( \langle o(x) \rangle_K \) being low with respect to the other terms.

Assuming the transmission is a scaled quantity of the object radiance, the method of defogging an averaged sequence is biased with the ratio of the object radiance variance due to turbulence and average object radiance, \( \sigma_o^2(x)/\mu_o(x) \).

The variance of the estimators is approximated using [112, 113],

\[ \text{var} \left[ \frac{R}{S} \right] \approx \frac{\text{var}[R]}{E[S]^2} - 2 \frac{E[R] \text{cov}(R, S)}{E[S]^3} + \frac{E[R]^2 \text{var}[S]}{E[S]^4}. \]  

(5.43)
The variance of the estimators are (dropping the pixel position term \(x\) for clarity)

\[
\text{var} [\langle \text{defog}(i(x)) \rangle] \approx \frac{1}{K} \left( \sigma_o^2 + \frac{\sigma_w^2}{\mu_t^2} \right),
\]

(5.44)

\[
\text{var} [\text{defog}(\langle i(x) \rangle)] \approx \frac{1}{K} \mu_t^2 \left( \sigma_w^2 + \left( \frac{\mu_t,o}{\mu_t} \right)^2 \sigma_t^2 - 2 \left( \frac{\mu_t,o}{\mu_t} \right) \sigma_t,o \sigma_t + \sigma_t,o^2 \right),
\]

(5.45)

where the joint expectation of the object radiance and transmission is \(\mu_{t,o,joint}\), variance is \(\sigma_{t,o}^2\), and correlation coefficient \(\rho\) is the correlation between \(\langle t(x) o(x) \rangle_K\) and \(\langle t(x) \rangle_K\). If the object radiance and transmission are statistically independent then (5.45) simplifies to (5.44) and hence \(\text{var} [\langle \text{defog}(i(x)) \rangle] \approx \text{var} [\text{defog}(\langle i(x) \rangle)]\); however, if we assume again that the transmission is a scaled version of the object radiance in (5.38), then (5.45) becomes

\[
\text{var} [\text{defog}(\langle i(x) \rangle)] \approx \frac{1}{K q \mu_o^2} \left( 3 \sigma_o^2 + \sigma_w^2 + \sigma_o^2 \left( 4 + q^2 - 2q^2 \rho \sqrt{4 \mu_o^2 + 3} \right) \right),
\]

(5.46)

for \(\mu_o \gg \sigma_o^2\) and

\[
\text{var} [\text{defog}(\langle i(x) \rangle)] \approx \frac{1}{K q \mu_o^2} \left( 3 \sigma_o^2 + \sigma_w^2 \right) \text{ for } \mu_o \rightarrow 0.
\]

(5.47)

For dark pixels, \(\mu_o \rightarrow 0\), the variance of the estimator in (5.47) is higher than the variance in (5.44). The \(3\sigma_o^2\) term is also in (5.46); however, claiming (5.46) has a higher variance than (5.44) requires \(4 + q^2 > 2q^2 \rho \sqrt{4 \mu_o^2 + 3}\), which is true for \(q \in [0, 1]\). Therefore the variance of the estimator \(\text{defog}[\langle i(x) \rangle]\) is higher than the estimator \(\langle \text{defog}[i(x)] \rangle\) for pixel regions where the transmission and object radiance are statistically dependent (depth discontinuities) and the statistics of the estimators are equivalent everywhere else. The fact that \(\text{defog}[\langle i(x) \rangle]\) has higher variance than \(\langle \text{defog}[i(x)] \rangle\) at depth discontinuities means that more frames (larger \(K\)) are needed to minimize the turbulence through averaging.

### 5.2.1 Turbulence Simulation

In order to demonstrate the validity of the analysis we introduce a turbulence simulation method similar to the approach by Zhu and Milanfar [114] to
Figure 5.3: $32 \times 32$ grid control point examples used to simulate hot-air turbulence by image warping. (a) Original grid points. (b) Warping with $\sigma_\Delta^2 = 0.5$ and $L = 8$. (c) Warping with $\sigma_\Delta^2 = 1$ and $L = 8$. (d) Warping with $\sigma_\Delta^2 = 0.5$ and $L = 4$. (e) Warping with $\sigma_\Delta^2 = 0.5$ and $L = 2$. (f)-(j) Siemens star chart images associated with the grid warpings from the first row.

generate the local motion parameter $\Delta_l(x)$ with $x = (x, y)^T$. Instead of B-spline interpolation, we employ bicubic interpolation. Given an image with size $N_x \times N_y$, a grid of anchor points spaced $N_x/L$ pixels apart in the vertical direction and $N_y/L$ pixels in the horizontal direction are randomly perturbed with a normally distributed random variable with zero mean and $\sigma_\Delta^2$ variance. The local translation caused by simulated turbulence is the bicubic interpolation of the randomly perturbed control points $(x^q, y^q)^T$:

$$\Delta_l(x) = A(x)p,$$  \hspace{1cm} (5.48)

where $p = (\delta x_1, \ldots, \delta x_m, \delta y_1, \ldots, \delta y_n)^T$. The perturbations on each grid point are normally distributed, $\delta x, \delta y \sim N(0, \sigma_\Delta^2)$. The interpolation matrix,

$$A(x) = \begin{bmatrix} c_1 & \cdots & c_n & 0 & \cdots & 0 \\ 0 & \cdots & 0 & c_1 & \cdots & c_n \end{bmatrix},$$ \hspace{1cm} (5.49)
is composed of coefficients from the cubic interpolation function $b(\cdot)$,

$$c_i = b \left( \frac{x - x_i^g}{N_x / L} \right) b \left( \frac{y - y_i^g}{N_y / L} \right).$$

With the new grid points, $\tilde{x} = x + \Delta_l(x)$, cubic interpolation is used to generate the $k^{th}$ image with simulated turbulence,

$$i_k(x) = t_k(\tilde{x}) o_k(\tilde{x}) + (1 - t_k(\tilde{x})) a + w_k(x).$$

We refer the reader to Keys [115] for the details on the interpolation functions.

We illustrate this grid warping in Fig. 5.3. The original $32 \times 32$ grid in Fig. 5.3(a) is manipulated with different parameters for $L$ and $\sigma^2_\Delta$. We have observed that the grid variance parameter $\sigma^2_\Delta$ can be intuitively associated with the structure constant $C_n^2$ which characterizes the strength of the index of refraction fluctuations [111]. We have also found that manipulating the grid spacing parameter $L$ has a similar effect to varying the Fried seeing parameter $r_0$ [111].

In order to simulate frame to frame turbulence changes, we assume the turbulence in the frame $i_k(x)$ is statistically independent of the frame $i_{k+1}(x)$. This is effectively similar to simulating short exposure image capturing in order to freeze the effects of the atmosphere [101]. Otherwise, each frame (or frames) would have to be averaged together to simulate long exposure image capture.

**Figure 5.4:** PSNR values from turbulence simulation experiment using (5.51) with two different methods: Defog Before (5.56) and Defog After (5.57) frame averaging. (a) With fixed $\sigma^2_w = 0.1$ and $L = 8$. (b) With $\sigma^2_\Delta = 1$ and $\sigma^2_w = 0.1$. (c) With $\sigma^2_\Delta = 0.5$ and $L = 8$. 
5.2.2 Simulation with Defogging and Frame Averaging

We demonstrate the performance difference between averaging defogged images and defogging an averaged image sequence with an experiment using the turbulence simulation method discussed in Section 5.2.1. We produce $K = 30$ images of a Siemens star chart, each with random turbulence parameterized by the variance $\sigma_{\Delta}^2$ and scale $L$ along with the image noise variance $\sigma_w^2$. We also add fog to each image

$$i_k(x) = t(\tilde{x}_k) o(\tilde{x}_k) + (1 - t(\tilde{x}_k)) a + w_k(x).$$  \hspace{1cm} (5.52)

The turbulence is simulated with only local translation and no global translation,

$$\tilde{x}_k = x + \Delta_{l,k}(x).$$  \hspace{1cm} (5.53)

In the experiment, we compare the original image, $o(x)$ with both defogging methods and averaging $K = 30$ frames. We use a Peak-to-Signal Noise Ratio (PSNR) measure for an objective comparison with

$$\text{PSNR}(o, f) = 20\log_{10} \frac{1}{\sqrt{MSE(o, f)}},$$  \hspace{1cm} (5.54)

$$MSE(o, f) = \frac{1}{M} \sum_x (o(x) - f(x))^2,$$  \hspace{1cm} (5.55)

and $M$ being the total number of pixels in the image. We distinguish the measurements for both methods with

$$\text{Defog Before : } \text{PSNR}(o(x), \langle \text{defog}(i(x)) \rangle)$$  \hspace{1cm} (5.56)

$$\text{Defog After : } \text{PSNR}(o(x), \text{defog}(\langle i(x) \rangle)).$$  \hspace{1cm} (5.57)

The defogging algorithm in (5.28) was used for both methods.

In the experiment, we fix two of the three parameters, $\sigma_{\Delta}^2, L, \sigma_w^2$, and vary the other parameter in order to see how the performance varies for each type of parameter.

The analysis in Section 5.2 shows that both methods, $\text{defog}(\langle i(x) \rangle)$ and $\langle \text{defog}(i(x)) \rangle$, are statistically the same when the transmission is uncorrelated with the object radiance. We first conducted the experiment with a spatially invariant
transmission map with \( t(x) = t = 0.2 \ \forall \ x \). This is to simulate a flat depth area. The PSNR values for both methods were indeed identical.

The analysis in Section 5.2 also indicates that the methods perform differently when the transmission and object radiance are correlated. Therefore we used a transmission map that was created with (5.38), \( t(\tilde{x}) = qo(\tilde{x}) \) with \( q = 0.2 \) and \( a = 0.8 \), to simulate the situation where the transmission is statistically correlated with the object radiance at depth discontinuities. The PSNR results from this approach are plotted in Fig. 5.4.

We first fix the scale \( L \) and image noise \( \sigma_w^2 \) parameters and vary the turbulence variance \( \sigma_\Delta^2 \) and plot the PSNR results in 5.4(a). The “Defog Before” method has higher PSNR than the “Defog After” method for all variances and each decrease and converge closer in PSNR as \( \sigma_\Delta^2 \) increases. Next, we fix \( \sigma_\Delta^2 \) and \( \sigma_w^2 \) and vary the scale parameter \( L \) and plot the PSNR results in 5.4(b). For all scales, the “Defog Before” method has higher PSNR. Finally, we fix the turbulence parameters, \( \sigma_\Delta^2 \) and \( L \), and vary the image noise variance and plot the PSNR results in 5.4(c). Both methods have near identical PSNR values at low \( \sigma_w^2 = 0.01 \) but diverge rapidly as the noise is increased with again the “Defog Before” method having higher PSNR.

In Fig. 5.5, we provide a zoomed in view of the images from the experiment. All parameters were fixed with \( \sigma_\Delta^2 = 2, \ L = 64, \ \sigma_w^2 = 0.05 \) and the number of frames generated was \( K = 30 \). The original image, \( o(x) \) is in Fig. 5.5(a). The images in the first row (Figs. 5.5(b)-5.5(d)) are from the experiment with a globally constant transmission \( t = 0.2 \). Image 5.5(b) is one image sample out of the 30 images generated. Images in Fig. 5.5(c) and 5.5(d) are from the “Defog After” and “Defog Before” methods, respectively. Both images are identical because Eq. (5.37) is satisfied and both methods are statistically the same.

When Eq. (5.37) is not true then the methods behave differently. We use the same spatially varying transmission, \( t(x) = q \ast o(x) \), for the images in the second row of Fig. 5.5. The image in Fig. 5.5(e) is a single frame from the simulated images. Figs. 5.5(f) and 5.5(g) are from the “Defog After” and “Defog Before” methods, respectively. Here we see a difference in images where regions in
Figure 5.5: Zoomed in view of images from simulation using (5.51) with parameters set to $\sigma^2_\Delta = 2$, $L = 64$, $\sigma^2_w = 0.05$, $a = 0.8$ and $K = 30$ frames. (a) Original image $o(x)$. (b) Single image from experiment with $t(x) = 0.2$. (c) “Defog After” method applied to image sequence with $t(x) = 0.2$. (d) ”Defog Before” method applied to image sequence with $t(x) = 0.2$. (e) Single image from experiment with $t(x) = 0.2o(x)$. (f) “Defog After” method applied to image sequence with $t(x) = 0.2o(x)$. (g) “Defog Before” method applied to image sequence with $t(x) = 0.2o(x)$.

the center of the “Defog After” image is brighter than the “Defog Before” image which is expected according to the analysis in (5.42).

5.3 Proposed Method

In this work, we propose a new method that is designed to mitigate atmospheric turbulence and remove fog in video fast enough for near real-time applications: Contrast Enhancement Turbulence Mitigation (CETM). Here we provide
an overview of the design illustrated in Fig. 5.6 and discuss the details of each component in the following subsections. Given a foggy turbulent image at frame $k$ $i_k(x)$, single image noise removal (denoising) is applied and passed into a single image defogging algorithm. The defogging algorithm estimates the transmission and $t_{max}$ and enhances the contrast image which then goes into the object tracker to estimate the global motion vector $\hat{\Delta}_{g,k}$. The enhanced image is then aligned with $\hat{\Delta}_{g,k}$ and locally aligned with an optical flow method which estimates $\hat{\Delta}_{l,k}(x)$. The globally and locally aligned contrast enhanced image is then temporally averaged with $K$ frames and finally atmospheric blur is removed (Deblur) to produce the estimated image of the object $\hat{o}_k(x)$. A turbulence mitigation metric (TMM) is used to measure the performance of the CETM by comparing the turbulence between the input and output image sequences.

### 5.3.1 Motion Compensation in Turbulence

Our design is based on the analyses in Sections 5.1 and 5.2. In Section 5.1 we found that contrast enhancement does not improve the object tracker performance when image noise is present. The result in terms of error, is the same if enhancement is before or after global tracking. Reducing image noise, however, can improve the object tracker performance therefore we propose to use single image denoising for improving the estimate of the global motion parameter $\hat{\Delta}_g$. Additionally, Matlin and Milanfar [116] observe that removing noise before fog removal is an important approach in order to improve transmission $\hat{t}$ and airlight $\hat{a}$.
estimation. We therefore propose to remove noise, enhance the contrast and then estimate the global motion $\hat{\Delta}_g$.

In order to reduce the complexity of our algorithm such that the processing speed is near real-time, we use a two dimensional median filter for each color channel for fast single image denoising. It is important the denoising step is not too aggressive in reducing spatial frequencies because important details can be lost and unable to be recovered during the frame averaging and deblurring steps. This is especially true in scenes under anisoplanatic conditions [117]. In anisoplanatic conditions, regions within an image vary in resolution over time and can even reach near diffraction limits. If noise removal is too aggressive, it effectively imposes premature bandwidth limitation and high frequency regions will be lost before the bandwidth recovery process (deblurring). In this work, we focus on isoplanatic conditions thus we use a very small window $3 \times 3$ for the median filtering step for denoising.

We employ the adaptive correlation filter, Minimum Output Sum of Squared Error (MOSSE), from Bolme et al. [107] to estimate the global motion parameter $\Delta_g$. This filter is fast and efficient and works well with turbulent video because the phase correlation update averages out the zero mean phase perturbations induced by turbulence [111].

As observed in [31, 101], the local motion induced by turbulence must also be compensated in order to have a sharper image after frame averaging. We use a similar approach to Joshi et al. [31] by estimating the dense optical flow with the method by Farneback [118].

We choose to enhance before optical flow in order to improve the motion estimation as was found by Gibson et al. [61]. In [61], it was found in theory and experiment that many motion vectors were possible candidates in low contrast producing inaccurate motion estimation. Enhancing before motion compensation reduces the number of possible motion vectors for each pixel. For video coding, intra-frame coding is used more when enhancing first therefore details are preserved whereas inter-frame coding is used more when there is low contrast which results in loss of details.
5.3.2 Single Image Defogging

To remove fog effectively and efficiently for each frame and have an estimate of transmission, we use the Locally Adaptive Wiener Defogging method by Gibson and Nguyen [8]. In their work, the Wiener Defogging method is 50 to 100 times faster than existing methods and can operate at real-time speeds for frames of sizes $720 \times 480$ and larger.

We propose a modification to the Wiener Defog method in order to automate the defogging process. The important step in removing fog with the Wiener Defogging method is the proper choice of size for the smoothing window $\Omega$. In [8], the optimal $\Omega$ must be heuristically chosen for each image. This can be a cumbersome task for real-time video enhancements. The choice of window size is a common problem in single image defogging. A choice of two window sizes are needed in [1] to properly refine the transmission estimate but yet preserve depth discontinuities. The window size must be sufficiently large in [4] to estimate a smooth transmission map that best reflects the scene depth. We present a method for automatically selecting $|\Omega|$ based on the statistics of the frames.

Given a foggy color image $i \in \mathbb{R}^3$, we wish to remove the fog by estimating the transmission $t$ and recover the true non-foggy image $o \in \mathbb{R}^3$. We make the same assumptions as in [4, 1, 8] that the transmission is independent of wavelength, therefore the image model (without motion, turbulence and blur) is the atmospheric dichromatic model [10]

$$i(x) = t(x) o(x) + (1 - t(x)) a + w. \quad (5.58)$$

The main idea in estimating the transmission with an adaptive local Wiener filter in [8] is the observation model is composed of the veiling $v = (1 - t)$ and texture $n_{txt}$. The observation is the dark pixel measurement (also used in [4, 1]),

$$d(x) = \min_{c \in r, g, b} i_c(x), \quad (5.59)$$

where $i_c(x)$ is the $c^{th}$ color channel of the image $i(x)$. The dark pixel measure in (5.59) takes advantage of the atmospheric dichromatic model in (5.58) by assuming at least one color component is possibly dark therefore exposing the atmospheric
veiling: $d(x) \sim (1 - t(x))a_c + w_c$ for $\min_{c \in r, g, b} o_c(x) \sim 0$. Thus the model is

$$d = v + n_{txt}, \quad (5.60)$$

where $v$ is the true veiling and $n_{txt}$ is the textured noise. The goal in the Wiener Defog method is to filter out the texture from the observation $d$ by treating the texture $n_{txt}$ as noise but preserve edges from depth discontinuities. For images with a large amount of texture (forest floor with leaves), the size of window must be sufficiently large to filter out the texture variations. The choice of $\Omega$ arises from the local moment estimators

$$\hat{\mu}_v(x) = \frac{1}{|\Omega(x)|} \sum_{j \in \Omega(x)} d(x), \quad (5.61)$$

$$\hat{\sigma}^2_v(x) = \frac{1}{|\Omega(x)|} \sum_{j \in \Omega(x)} d(x)^2 - \hat{\mu}_v^2(x), \quad (5.62)$$

where $|\Omega(x)|$ is the number of pixels within the local sample window positioned at pixel location $x$.

The Wiener Defogging algorithm in [8] estimates the fog-free image $\hat{o}$ and transmission $\hat{t}$,

$$\hat{o}, \hat{t} \leftarrow \text{WienerDefog}(i, \Omega, t_0, d_w, 1), \quad (5.63)$$

where $\Omega$ is the $|\Omega|^{1/2} \times |\Omega|^{1/2}$ size sample window, $t_0$ is the minimum possible transmission value used for numerical conditioning, $d_w$ is the weight used to change the strength of the enhancement and the last parameter, 1, indicates the single-step approach is used. Gibson and Nguyen [8] observed that the transmission estimation is invalid when there is significant variation in fog causing the model in (5.60) to be invalid because the veiling and noise are not statistically independent. The texture variation $n_{txt}$ is dependent on the amount of fog. Therefore the first step naively assumes the model is valid in order to remove fog and to provide the ability to re-estimate the local moments $\sigma^2_n$ and $\sigma^2_v$ that are indeed statistically uncorrelated. The authors have observed that images with significant depth discontinuities (buildings in near foreground with sky as backdrop), the 2-step approach is needed. In this work, a 1-step approach has been found to be sufficient.
for most scenes therefore we utilized the 1-step Wiener Defogging method in this work.

The approach to estimating the optimum window size $|\Omega|$ is to choose a size sufficient enough to filter scene texture noise from the observation $\hat{\mu}_v$ in (5.61) to a desired variance $\sigma_e^2$:

$$\sigma_e^2 = \frac{\hat{\sigma}^2_{n_{txt}}}{|\Omega|} \Rightarrow |\Omega| = \frac{\sigma^2_{n_{txt}}}{\sigma_e^2}. \quad (5.64)$$

To determine $|\Omega|$, the variance of the texture must be estimated. Taking the maximum variance from the local variance measurement $\hat{\sigma}^2_v(x)$ in (5.62) is not practical since the higher variance measurements are from the depth discontinuities and edges in the scene. The goal in smoothing $d$ is to preserve these edges. We take a similar approach to Lim [108] in estimating the texture noise in an image. The approach is to measure the average variance in all color channels from the observed image $i(x)$,

$$\sigma^2_{n_{txt}} = \frac{1}{3M} \sum_{c=1}^{3} \sum_{x \in \Phi(x)} \hat{\sigma}^2_{i,c}(x), \quad (5.65)$$

where

$$\hat{\mu}^2_{i,c}(x) = \frac{1}{|\Phi(x)|} \sum_{y \in \Phi(x)} i_c(y), \quad (5.66)$$

$$\hat{\sigma}^2_{i,c}(x) = \frac{1}{|\Phi(x)|} \sum_{y \in \Phi(x)} i_c(y)^2 - \hat{\mu}^2_{i,c}(x), \quad (5.67)$$

and where $M$ is the total number of pixels in an image. A sample window size $|\Phi|$ must be selected initially in order to estimate the local mean and variance of the image. We use a sample window $\Phi = 32 \times 32$ for all image sequences in this work which has been found effective for most scenes. We estimate the sample window size $|\Omega|$ by using (5.64) and (5.65) with $\sigma_e^2 = 5e-6$ (for images scaled to pixel values to $[0, 1]$).

We provide an example of single image defogging using the Wiener Defogging method in Fig. 5.7. Given the foggy image in 5.7(a), the defogged image in Fig. 5.7(b) is generated with the transmission estimate in Fig. 5.7(c). For this
Example of Wiener Defogging method. (a) Foggy image \( i(x) \). (b) Defogged image \( o(x) \) using the 1-step Wiener Defogging method [8]. (c) The transmission map estimate \( \hat{t} \) (normalized for display purposes). The sample window size \( |\Omega| = 14 \) was estimated with (5.64).

\[ a_c = \frac{1000}{M} \sum_y i_c(y) : \hat{t}(y) \leq T, \quad (5.68) \]

where \( T \) is a threshold chosen such that there are \( M/1000 \) pixels that have a value lower than \( T \).

### 5.3.3 Contrast Dependent Frame Averaging

Recall from Section 5.2 that the variance of averaging a defogged image sequence that has already been aligned with motion compensation is

\[ \text{var}[\langle \text{defog}(i(x)) \rangle] \approx \frac{1}{K} \left( \sigma^2_o(x) + \frac{\sigma_w^2}{\mu^2_t(x)} \right), \quad (5.69) \]
The noise variance is reduced with $K$ but amplified with $\langle t(x) \rangle_k^2 = \mu_t^2(x)$. (See Eq. (5.29).) In order to reduce the noise power, the averaging parameter $K$ must depend on the average transmission $\mu_t(x)$. One approach is to choose a large $K$ value but this is not ideal for most scenes where a smaller $K$ is a better choice to prevent excessive blur.

In [31], the choice of number of frames $K$ to average is critical such that too few frames will not reduce the image noise or turbulence however too many frames will produce a blurred (bandlimited) image. In their work, Joshi et al. [31] propose a pixel-based weighted averaging method. The weights were designed to account for image texture, resampling, and selectivity. The pixel-based selectivity weight is similar to our approach in that the fewer the number of frames used for averaging, the less blur will occur if the frames are well aligned. In [31], this selectivity parameter was dependent on the texture of the image. The higher the texture, the fewer number of frames were used.

It is possible to design a spatially variant parameter $K(x)$, that is dependent on the statistics and average transmission estimation $\hat{\mu}_t^2(x)$,

$$K(x) = \begin{cases} K_{\text{max}}, & \text{if } \tilde{K}(x) > \tau_{\text{max}} \\ K_{\text{min}}, & \text{if } \tilde{K}(x) < \tau_{\text{min}} \\ \text{round}(\tilde{K}(x)), & \text{otherwise.} \end{cases} \quad (5.70)$$

with $\tilde{K}(x) = \frac{1}{\sigma_e^2} \left( \sigma_o^2(x) + \frac{\hat{\sigma}_w^2}{\hat{\mu}_t^2(x)} \right), \quad (5.71)$

Here, $\sigma_e^2$, is the desired variance in time for each pixel after frame averaging. In a similar approach to estimating the sample window size $|\Omega|$ in Section 5.3.2, we adapt the sample window size according to the measured statistics only here it is in the temporal domain. Similar to estimating from a subset of pixels based on the transmission for estimating the airlight in (5.68), we estimate the noise variance in time, $\hat{\sigma}_w^2$, in pixel regions where the estimated average transmission $\hat{\mu}_t(x)$ is low because in those regions only the noise power is present,

$$\text{var}[i(x)] = \sigma_w^2 \forall x : t(x) \sim 0, \quad (5.72)$$

where we indicate the variance measure in time with $\text{var}_k[\cdot]$. (Note that we assume the noise power $\sigma_w^2$ is independent of direction in space or time therefore has the
same variance.) The signal variance from atmospheric turbulence is estimated with
\[ \hat{\sigma}_o^2(x) = \frac{\langle i(x)^2 \rangle_{K_i} - \langle i(x) \rangle^2_{K_i}}{\langle i(x) \rangle^2_{K_i}} - \hat{\sigma}_w^2. \]  
(5.73)

The noise variance in time is estimated within the pixels with low transmission values according to (5.72),
\[ \hat{\sigma}_w^2 = \frac{1000}{M} \sum_y \langle i(y)^2 \rangle_{K_i} - \langle i(y) \rangle^2_{K_i}, \]  
(5.74)
where \( T_k \) is chosen to select the \( M/1000 \) pixels with low transmission values.

As in the case for estimating the sample window \( |\Omega| \), an initial window is required. An initial set of number of frames \( K_i \) is used to estimate the temporal mean and variance parameters in (5.73). For all experiments, we set \( K_i = 30 \).

We illustrate spatially variant averaging in Fig. 5.8. The image of a Siemens star chart (grayscale) in fog and turbulence is simulated such that the left half of the image has \( t(x) = 0.1 \) and the right half of the image is \( t(x) = 0.9 \). The airlight was set to \( a = 0.8 \). We set the turbulence with variance \( \sigma_\Delta = 0.5 \) and scale \( L = 64 \).

We simulated 150 frames with a single frame illustrated in Fig. 5.8(a) \( (\sigma_w^2 = 0.09) \). The sequence was averaged with a global number of frames \( K = 30 \) and the spatially variant \( K(x) \) parameter from (5.70) in Figs. 5.8(b) and 5.8(c) respectively. The desired \( \sigma_e^2 \) was set to 0.01. The noise in the left half of the image (from global \( K \)) in Fig. 5.8(b) is more noticeable than in Fig. 5.8(c) (from spatial variant \( K(x) \)).

We measure the PSNR with various noise values in Figs. 5.8(d) and 5.8(e). The desired \( \sigma_e^2 \) is set to 0.0001 in Fig. 5.8(d) and \( \sigma_e^2 = 0.01 \) in Fig. 5.8(e). Overall, using the spatially varying \( K(x) \) is best in PSNR. The PSNR performance is similar or lower when the desired noise power \( \sigma_e^2 \) is set too high. This is evident in Fig. 5.8(e) with noise power less than 0.01 \( (\sigma_w^2 \leq 0.1) \). In that region, the averaging parameter \( K(x) \) is lower than the global averaging parameter \( K \) which produces an image with more noise variance.

An advantage of using \( K(x) \) is that only a single parameter is needed and the averaging adapts to the noise power present. This is evident in Fig. 5.8(d). The PSNR for using \( K(x) \) stays the same for \( \sigma_w^2 \leq 0.03 \) and gradually decreases at a slower rate than using \( K \) as \( \sigma_w^2 \) increases.
Figure 5.8: Comparison of averaging with global $K$ and spatially variant $K(x)$. (a) Single image from sequence with fog, noise and turbulence. (b) Image result from averaging $K = 30$ frames. (c) Image result frame averaging with $K(x)$ frames in (5.70). (d) PSNR comparison with $\sigma_e^2 = 10^{-3}$. (e) PSNR comparison with $\sigma_w^2 = 0.01$.

Even though the PSNR results are better in the experiment with using a spatially varying $K(x)$, this approach is not the best in practical applications. The image sequence suffers from flickering artifacts caused by regions of the image being averaged more than others due to inaccuracy in estimating the parameters in (5.71). In addition to the flickering artifacts, the complexity of averaging per pixel with different $K$ values is high which is not desirable for near real-time applications.

We desire to use a global averaging method to reduce flickering and achieve low complexity and yet adaptive to the signal noise and contrast. We propose to solve the problem by providing a method for determining the lower bound for $K$ given a desired $\sigma_e^2$. This can be achieved if in (5.71) $t_{max}$ is used in place of $\mu_x(x)$, where $t_{max}$ is the highest transmission value estimated from the image sequence. Therefore, the minimum number of frames to average for an image sequence with $M$ pixels per frame can be determined with a contrast dependent measure and averaged $\sigma_o^2$,

$$K_{min} = \frac{1}{\sigma_e^2} \left( \frac{1}{M} \sum_x \sigma_o^2(x) + \frac{\sigma_w^2}{t_{max}^2} \right),$$

(5.75)

with $t_{max} : t_{max} \geq \hat{t}(x) \forall x$. 

5.3.4 Deblur

When the image $i(x)$ has been defogged, motion compensated, and then frame averaged, the final step is to remove the atmospheric blur (deblur). The method we present in this work uses a Wiener Deconvolution method proposed by Roggemann and Welsh [111] and most recently by Droge et al. [119]. In order to remove the blur, a kernel must be known. One approach is to apply a blind deconvolution method as proposed in [116]. For isoplanatic conditions, the blur kernel is not exactly known but the structure is. Fried [99] noted that there were two different forms of the optical transfer function (OTR), or blur kernel, due to turbulence: one for long exposure and the second for short exposure. The exposure time is in relation to the atmospheric turbulence. Short exposure measurements effectively “freezes” the turbulence during the image capture integration time. Only one instance of turbulence is captured in a short exposure frame. Long exposure is an integration of multiple turbulent instances in one frame capture. Given an OTF that represents the blur from turbulence, $H(u, \lambda)$, where $u$ is the spatial frequency and $\lambda$ is the optical wavelength (color channel), the long exposure OTF is

$$H_{LE}(u, \lambda) = e^{-\frac{1}{2}6.88\left(\frac{\lambda l_0}{u}\right)^{5/3}}, \quad (5.76)$$

and short exposure OTF is

$$H_{SE}(u, \lambda) = H_{LE}^{1-\alpha}\left(\frac{\lambda l_0}{u}\right)^{1/3}. \quad (5.77)$$

The optic focal length is $l$ and diameter $D$. The parameter $\alpha$ is for near ($\alpha = 1$) and far field ($\alpha = 0.5$). We address far field imaging therefore $\alpha = 0.5$. The Fried parameter is $r_0$. We have observed that modifying the ratio $l/r_0$ for both short and long exposure and setting $D$ manually is sufficient for situations when the optical system and seeing parameters are not known and in fact is a common situation because $r_0$ is typically unknown.
5.3.5 Turbulence Mitigation Metric

In order to illustrate that the turbulence has been mitigated in time with our proposed method, we present in this work a Turbulence Mitigation Metric (TMM). The goal in turbulence mitigation is to not only recover the true structure (phase) of the scene but also to enforce temporal consistency in each color channel. Instead of using a subjective measure by displaying multiple frames of a video sequence, we develop an objective measure. Our goal in designing the TMM is to make it an objective measure of turbulence mitigation, simple to construct, and easily accessible such that it can be used in any other turbulence mitigation method. One approach to developing a TMM is to utilize the motion estimated from optical flow and global estimations. This approach, nonetheless, is complex given that not every method uses optical flow as in [110]. We wish to only require an input image sequence and output image sequence.

Suppose an image sequence is subjected to turbulence where only local motion exists, \( \Delta_g = 0, \Delta_l(x) > 0 \). If a sample window is placed at pixel position \( x \) and extracts colors within the window \( \Omega(x) \), those pixels exhibit a color ellipsoid shape in the RGB domain [86]. Sample windows placed in regions with edges and textures which are the most affected by turbulence will exhibit color ellipsoid shape change in time. We wish to measure this change with the TMM. This effectively is a measure of the variance of colors in time.

The TMM is a measurement between the input image sequence \( i \) and output image sequence \( o \),

\[
TMM(i, o) = \frac{\text{VOC}(i) - \text{VOC}(o)}{\text{VOC}(i)},
\]

where the Variance of Colors variations measurement \( \text{VOC}(\cdot) : \mathbb{R}^{M_1 \times M_2 \times 3 \times K} \mapsto \mathbb{R} \) is

\[
\text{VOC}(f) = \text{mode}_x \left( \text{var}_K \left[ \text{gray} \left( \sqrt{\text{var}_{\Omega(x)}[f]} \right) \right] \right),
\]

where each image sequence is of spatial dimension \( M_1 \times M_2 \), has 3 color channels and is of length \( K \) frames. The \( \text{VOC} \) first measures the spatial standard deviation in color for every frame in the image sequence using a sliding window \( \Omega(x) \) with
var_{\Omega(x)}[\cdot]. Then the image is converted to grayscale with gray(\cdot). The variance in time for each pixel is then measured with var_K[\cdot]. For the VOC measurement, we set \( K \) to 10 for all experiments. Finally, since the image of the variation of color variances in time is a summation of powers, we measure the mode given the \( M \) pixels in the image.

The sample window size \(|\Omega|\) in (5.79) is an important parameter to select when measuring the VOC. If \(|\Omega|\) is too small, the color variations will not be estimated properly due to lack of information. If, however, the sample window is too large then the time to measure the metric will be too long. In our experiments, we use the same window size estimated from the Wiener Defogging method in (5.64). The sample window size \(|\Omega|\) is adaptive to each scene.

If the turbulence is reduced then the value of the TMM approaches unity. The TMM may be negative if the input sequence exhibits less turbulence than the output sequence. When contrast enhancement is involved, it is important to measure the TMM between the output image sequence and the input image sequence immediately after contrast enhancement. This is because the color variances are smaller when fog is present [8] and the color variations in time are smaller compared to the same image sequence with the contrast enhanced which would produce negative TMM values.

5.4 Implementation

Our experiments were conducted on a 64-bit machine with Intel(R) Core (TM) i7-2520M CPU @ 2.7GHz (4 processors) and 8GB of RAM (crucial for large \( K \) values). The software was written in Python v2.7 using OpenCV, SciPy and Numpy libraries \(^1\) similar to the tools used by Bolme et al. [107].
5.5 Results and Discussion

To demonstrate the proposed method along with the TMM, we applied our Contrast Enhancement Turbulence Mitigation (CETM) method to the video sequence that was constructed in Section 5.3.3. We measured the PSNR and TMM values.

Figure 5.9: Performance measurement of the CETM with simulated turbulent/foggy sequence in Fig. 5.8. (The turbulence variance $\sigma_\Delta = 0.5$ and scale $L = 64$. (a) The TMM measurements for various $K$ values. (b) The PSNR values for each $K$ values.) (c) The PSNR performance of the CETM at different stages of the algorithm.

\[\text{http://www.opencv.org and http://www.python.org}\]
values with different $K$ values and plot the results in Fig. 5.9. The minimum frames to average $K_{\text{min}} = 44$ was estimated from (5.75).

The TDD measure starts low and on average increases for after each new image. The TDD starts at 0 at frame $K = 10$ because the sample size in time was set to 10. The highest TDD is with $K = 90$ and lowest with $K = 10$. What is interesting is that there is a significant improvement from 10 to 44 frames but not much improvement when the number of frames to average is almost doubled from 44 to 90. A similar observation can be made with the PSNR values in Fig. 5.9(b). The PSNR is improved dramatically at the beginning and the performance reaches a limit at each $K$ value.

The performance of the method at each stage of the algorithm is illustrated in Fig. 5.9(c). The same simulated sequence is used and each measurement is with $K = 44$. The PSNR of the contrast enhanced sequence is very low because noise is introduced after enhancement. The optical flow slightly improves the result and a dramatic improvement occurs with frame averaging. Frame averaging without optical flow performs about 1 dB lower than with optical flow.

We now demonstrate the performance of our proposed CETM method with real image sequences. For each image sequence, the minimum number of frames $K_{\text{min}}$ in (5.75) was used with $\sigma^2_e = 2 \times 10^{-5}$. In Fig. 5.10, we have 4 different scenes with various atmospheric artifacts present (fog, haze and turbulence). We apply the CETM method to the images in the top row of Figs. 5.10(a) - 5.10(d). The images just below are the results from the CETM method estimating $\hat{o}(x)$. There is a significant improvement for each image sequence in contrast and image structure recovery.

The metrics in Table 5.1 also provide information on how well the CETM does in the temporal domain. From left to right in each column is the name of the video sequence, the input VOC, output VOC from the CETM method, the TMM, the size of each frame and the processing speed measurement, frames per second (FPS). The highest TMM values $TMM = 0.94$ were with the Tower and Star image sequences. The Star sequence exhibited the most severe turbulence with $\text{VOC}(i) = 4.7 \times 10^{-3}$. This sequence in fact is obtained within anisopla-
natic conditions and yet the CETM method does well in recovering the color and resolution.

For the boat scene (and for most scenes with motion from the ocean surface), the TMM will be difficult to increase to unity regardless of number of frames chosen because of the periodic local motion induced by the dynamic textures from the ocean. This is why the TMM is close to 0.7 instead of 0.9. A better approach would be to weight the VOC measurements according to regions of interest and ignore areas with that exhibit local motion not caused by the turbulent atmosphere.

The TMM is very low in the Moon image \( TMM = 0.26 \), however, subjectively the video sequence is acceptable after applying the CETM method. This is because the image sequence initially has slowly varying turbulence where the VOC of the input sequence is the lowest of the sequences with \( VOC(i) = 1.4 \times 10^{-4} \). Additionally, the number of frames was selected to be low with \( K = 7 \).

We also provide a comparison with the results\(^2\) from Zhu and Milanfar [9] in Figs. 5.10(e) and 5.10(f). In comparison, the images appear similar in quality for both the Water and Moon images except that the contrast is increased with CETM. We also emphasize that the time to process these sequences for each frame for the CETM method is fractions of a second (3 to 8 fps). It has been found that the method by Zhu and Milanfar [9] takes orders of magnitude more time to process with respect to the CETM method. More results can be found at http://videoprocessing.ucsd.edu/~kgibson.

5.6 Acknowledgment

The text of Chapter 5 is adapted from An Analysis and Method for Contrast Enhancement Turbulence Mitigation, Kristofer Gibson, Truong Nguyen, in submission for IEEE Transactions on Image Processing in July of 2013. The dissertation author is the primary author of this publication.

\(^2\)Original and processed images obtained from http://users.soe.ucsc.edu/~xzhu/doc/turbulence.html
Table 5.1: Performance metrics for CETM

<table>
<thead>
<tr>
<th>Video</th>
<th>$K_{\text{min}}$</th>
<th>VOC(i) $\times 10^{-4}$</th>
<th>VOC(o) $\times 10^{-4}$</th>
<th>TMM</th>
<th>w×h</th>
<th>fps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boat</td>
<td>16</td>
<td>9.2</td>
<td>2.4</td>
<td>0.74</td>
<td>432×240</td>
<td>4</td>
</tr>
<tr>
<td>Star</td>
<td>56</td>
<td>47.0</td>
<td>3.0</td>
<td>0.94</td>
<td>108×90</td>
<td>8</td>
</tr>
<tr>
<td>Tower</td>
<td>41</td>
<td>8.2</td>
<td>0.5</td>
<td>0.94</td>
<td>304×224</td>
<td>8</td>
</tr>
<tr>
<td>Moon</td>
<td>7</td>
<td>1.4</td>
<td>1.0</td>
<td>0.26</td>
<td>432×400</td>
<td>3</td>
</tr>
</tbody>
</table>
Figure 5.10: CETM applied to real image sequences. Images (a) - (d) are frame captures from the image sequences where the original sequence is on top and the result from applying the CETM method is on the bottom. Images (e) and (f) are from Zhu and Milanfar’s proposed method in [9]. (a) Boat image sequence. (b) Star image sequence. (c) Water Tower. (d) Moon.
Chapter 6

Conclusion

“My religion consists of a humble admiration of the illimitable superior spirit who reveals himself in the slight details we are able to perceive with our frail and feeble mind.” - Albert Einstein

In Chapter 2, we opened with a hypothetical but realistic situation where an engineer needed to decide on choosing a *Pre* or *Post* method for enhancing and compressing video for a surveillance application where fog and haze are prevalent. Current enhancement techniques are explored and a physics model that is a popular choice for dehazing images is introduced.

We present a novel defogging method that works well without producing *halo* artifacts and is fast because of its use of a single median filter operation. We then analytically showed how using a *Pre* method produces fewer blocking and ringing artifacts compared to the *Post* method when JPEG compression is used. We then simulated the results that confirmed the analysis. We extended our analysis to motion estimation which again showed that the *Post* method suffers from lower SNR after decoding which causes more erroneous block MV estimations than the *Pre* method.

Subjective and objective test results using H.264 video sequences are provided to support the claim that the *Pre* method is a better choice. The subjective test showed the observers preferred the *Pre* sequences over the *Post* sequences. The observers also found it difficult to distinguish between a low bitrate *Pre* sequence and a higher bitrate *Post* sequence which differed by a factor of 4. This again
supported the claim that the Pre method is a better choice than the Post method when perceptual quality along with reduction of blocking and ringing artifacts is important.

The development of the Color Ellipsoid Framework in Chapter 3 is a contribution to the field of work in single image defogging because it brings a richer understanding to the problem of estimating the transmission. This paper provides the tools necessary to clearly understand how transmission is estimated from a single foggy image. We have introduced a new method, the Centroid Prior, that is visually more aggressive in removing fog which affords an image that is vibrant in color.

We presented in Chapter 4 a new perceptually based contrast enhancement metric that outperforms existing contrast enhancement metrics and even our previously developed method in [53]. The results from this project advances the field of contrast enhancements by demonstrating that HVS type contrast enhancement metrics can be improved in performance using a machine learning approach. Another key message is that we introduce new features by using the physics of the scene, image scale, and sub-metrics with different parameters in order to improve the performance of a perceptually based contrast enhancement metric. The new sub-metrics based on the Color Ellipsoid Framework in fact perform individually as well as previous HVS type methods as illustrated in Table 4.1. We provide a novel method for training the CEM properly with AdaBoost by filtering out images according to confidence measurements in order to force AdaBoost to focus training on important difficult image pairs. We also demonstrate the CEM is capable of training from a small image set and able to sort diverse images according to their contrast (see Fig. 4.7).

Finally, in Chapter 5, we developed an analysis of how global motion estimation is affected when the contrast decreased. We have found that increasing contrast before tracking does not improve the performance when spatial noise is present. For removing turbulence it is common to average motion compensated images together in order remove the turbulent artifacts. We provide an analysis on how removing fog before frame averaging is a better approach than removing fog
after frame averaging because of the depth discontinuities in scenes. If there are no depth discontinuities than both approaches are the same. We then present a new CETM method that is designed according to the analysis. We developed a contrast dependent frame averaging technique that measures the minimum possible number of frames needed to achieve a desired temporal variation. We then develop a new turbulence mitigation metric (TMM) and variance of color (VOC) measure that provides an objective measure for temporal consistency which is desired in turbulence mitigation. Our proposed method performs comparable to existing complex methods and operates within fractions of a second which is a desirable capability for near real-time video processing.

In summary, we begin explanation of the need for enhancements before compression when fog is present, we introduce a framework that explains how fog removal is successful and use properties from the color ellipsoid framework to propel a contrast enhancement metric into near 90% accuracy. Finally, we provide a solution to the problem of imaging in the atmosphere: contrast enhancement and turbulence mitigation. The contributions from this overall work is a dissertation in how knowledge is gained from light propagated in the maritime domain.
Bibliography


