Florida State University Libraries

Electronic Theses, Treatises and Dissertations

The Graduate School

2016

Automatic Fog Detection in Day and Night Images to Improve Highway Driving Conditions

Jigar Kamlesh Patel

FLORIDA STATE UNIVERSITY
COLLEGE OF ENGINEERING

AUTOMATIC FOG DETECTION IN DAY AND NIGHT IMAGES TO IMPROVE HIGHWAY
DRIVING CONDITIONS

By
JIGAR PATEL

A Thesis submitted to the
Department of Computer & Electrical Engineering
in partial fulfillment of the
requirements for the degree of
Master of Science

2016

Jigar Patel defended this thesis on September 09, 2016.
The members of the supervisory committee were:

Victor DeBrunner
Professor Directing Thesis

Uwe Meyer-Baese
Committee Member

Bruce Harvey
Committee Member

The Graduate School has verified and approved the above-named committee members, and certifies that the thesis has been approved in accordance with university requirements.

TABLE OF CONTENTS

List of Tables	v
List of Figures	vi
Abstract	xi
1 Background & Related Work	1
1.1 Fog Detection	1
1.1.1 California Weather System	4
1.1.2 Alabama Weather System	5
1.1.3 West Virginia Weather System	7
1.1.4 Virginia Weather System	7
1.1.5 Tennessee Weather System	8
1.1.6 Georgia Weather System	8
1.1.7 Utah Weather System	9
1.1.8 South Carolina Weather System	9
1.1.9 Maryland Weather System	9
1.1.10 Florida Weather System	10
1.1.11 New Jersey DOT	11
1.1.12 Other Studies	12
2 Visibility of Range Detector using a single camera	16
2.1 Preprocessing	17
2.1.1 Inverse Transformation	19
2.1.2 Logarithmic Transform	20
2.1.3 Power-Law (Gamma) Transformation	21
2.1.4 Contrast Stretching	21
2.2 Wavelet Transform	25
2.2.1 Wavelet Series Expansion	26
2.2.2 Discrete Wavelet Transform in one-dimension	27
2.2.3 Discrete Wavelet Transform in two-dimension	27
2.3 Sharpness Measurement	29
2.3.1 Global Sharpness	30
2.3.2 Local Sharpness	31
2.4 Calibration Problem	32
3 Study & Results	33
3.1 Preprocessing	33
3.2 Discrete Wavelet Transform	40
3.3 Sharpness Analysis	82
3.3.1 Global Sharpness	83
3.3.2 Local Sharpness	85

4 Conclusion & Future Work	98
---------------------------------------	-----------

Appendix

A MATLAB Code	99
A.1 Preprocessing Functions	99
A.1.1 Negative Transform	99
A.1.2 Logarithmic Transform	99
A.1.3 Power Law Transform	99
A.1.4 Contrast Stretching Transform	99
A.2 DWT Code	100
A.3 Sharpness Function	102
A.3.1 Local Sharpness Function	103
Bibliography	104
Biographical Sketch	107

LIST OF TABLES

1.1	Fog System information by state along with implementation year and effectiveness as reported by [20]	3
1.2	Fog system components for the WVU Active system	3
2.1	Contrast Stretching example parameters	23
3.1	DWT Image scales along with image sizes for the respective scale.	40
3.2	Legend information for Figures 3.62 to 3.67	84
3.3	Output values for Sharpness analysis based on image size and window size.	86

LIST OF FIGURES

1.1	Alabama DOT low visibility warning system screen shot [20]	6
1.2	Fog Density Prediction Model	14
2.1	Fog Detector workflow	16
2.2	Normalization (a) Day fog image (b) Night fog image	18
2.3	Preprocessing technique Examples (a) Original Image (b) Inverse Image (c) Log Transform (d) Power Law($\gamma = 0.3$) (e) Power Law ($\gamma = 4$) (f) Contrast Stretching	18
2.4	Histograms for the previously displayed images on page 18 (a) Original Image Histogram (b) Inverse Image Histogram (c) Log Transform Histogram (d) Power Law Histogram ($\gamma = 0.3$) (e) Power Law Histogram ($\gamma = 4$) (f) Contrast Stretching	19
2.5	Some basic intensity transformation functions. Curves are scaled to fit in the range shown [11]	20
2.6	Generic Power-Law Transformation plot for different values of γ with $\mathbf{c} = \mathbf{1}$ [11]	21
2.7	Intensity Transform Functions. (a) Contrast Stretching function. (b) Thresholding function. (c) Form of transformation function. [11]	22
2.8	Comparison of contrast stretching method using different tolerance values. The first row is all the images, the second row displays the histograms for the respective images, and the third row is the transformation profile applied to the image. Parameters for the images are: (a) Original Image (b) $\text{tol} = 1\%$, $(r_1, r_2) = (56, 248)$, $(s_1, s_2) = (0, 255)$ (c) $\text{tol} = 10\%$, $(r_1, r_2) = (89, 249)$, $(s_1, s_2) = (0, 255)$	25
2.9	Wavelet Profiles (a) Haar Wavelet (b) Various Daubechies Wavelets	26
2.10	Analysis filter bank showing the input and output of a 1-scale 2-D DWT (image used from [11])	28
2.11	Example image (a) original image (b) 1-scale DWT (c) 2-scale FWT (d) 3-scale DWT [11]	29
2.12	Gradient Image Example: The left image is the actual image. The image in the middle is the horizontal gradient image, and the image on the right is the vertical gradient image [22].	30
2.13	Square shaped window size filters used to compute the local sharpness of the images (a) window size = 2 (considered an edge detector filter) (b) window size = 4 (sometimes	

	considered an edge detector filter as well) (c) window size = 8 (d) window size = 16 (e) window size = 32	31
3.1	Original Images	34
3.2	Negative Transform	34
3.3	Logarithmic Transformation	34
3.4	Power Law $\gamma = 2$	35
3.5	Power Law $\gamma = 3$	35
3.6	Power Law $\gamma = 4$	35
3.7	Power Law $\gamma = 5$	35
3.8	Power Law $\gamma = 6$	36
3.9	Power Law $\gamma = 7$	36
3.10	Contrast Stretched Images	36
3.11	Original Image and it's histogram	37
3.12	Negative Transform Image and it's histogram	37
3.13	Logarithmic Transform Image and it's histogram	38
3.14	Power Law Transform Image and it's histogram for $\gamma = 2$	38
3.15	Power Law Transform Image and it's histogram for $\gamma = 3$	38
3.16	Power Law Transform Image and it's histogram for $\gamma = 4$	38
3.17	Power Law Transform Image and it's histogram for $\gamma = 5$	39
3.18	Power Law Transform Image and it's histogram for $\gamma = 6$	39
3.19	Power Law Transform Image and it's histogram for $\gamma = 7$	39
3.20	Contrast stretched Image and it's histogram	39
3.21	DWT images for original day clear (a) 1-Stage DWT (b) 2-Stage DWT (c) 3-Stage DWT (d) 4-Stage DWT (e) 5-Stage DWT	41
3.22	DWT for Original Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	42
3.23	DWT for Original Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	43

3.24	DWT for Original Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	44
3.25	DWT for Original Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	45
3.26	DWT for Negative transform Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	46
3.27	DWT for Negative transform Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	47
3.28	DWT for Negative transform Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	48
3.29	DWT for Negative transform Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	49
3.30	DWT for Logarithmic transform Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	50
3.31	DWT for Logarithmic transform Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	51
3.32	DWT for Logarithmic transform Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	52
3.33	DWT for Logarithmic transform Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	53
3.34	DWT for Power Law transform ($\gamma = 2$) Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	54
3.35	DWT for Power Law transform ($\gamma = 2$) Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	55
3.36	DWT for Power Law transform ($\gamma = 2$) Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	56
3.37	DWT for Power Law transform ($\gamma = 2$) Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	57
3.38	DWT for Power Law transform ($\gamma = 3$) Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	58
3.39	DWT for Power Law transform ($\gamma = 3$) Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	59

3.40	DWT for Power Law transform ($\gamma = 3$) Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	60
3.41	DWT for Power Law transform ($\gamma = 3$) Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	61
3.42	DWT for Power Law transform ($\gamma = 4$) Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	62
3.43	DWT for Power Law transform ($\gamma = 4$) Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	63
3.44	DWT for Power Law transform ($\gamma = 4$) Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	64
3.45	DWT for Power Law transform ($\gamma = 4$) Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	65
3.46	DWT for Power Law transform ($\gamma = 5$) Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	66
3.47	DWT for Power Law transform ($\gamma = 5$) Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	67
3.48	DWT for Power Law transform ($\gamma = 5$) Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	68
3.49	DWT for Power Law transform ($\gamma = 5$) Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	69
3.50	DWT for Power Law transform ($\gamma = 6$) Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	70
3.51	DWT for Power Law transform ($\gamma = 6$) Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	71
3.52	DWT for Power Law transform ($\gamma = 6$) Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	72
3.53	DWT for Power Law transform ($\gamma = 6$) Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	73
3.54	DWT for Power Law transform ($\gamma = 7$) Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	74
3.55	DWT for Power Law transform ($\gamma = 7$) Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	75

3.56	DWT for Power Law transform ($\gamma = 7$) Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	76
3.57	DWT for Power Law transform ($\gamma = 7$) Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	77
3.58	DWT for Contrast Stretched Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	78
3.59	DWT for Contrast Stretched Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	79
3.60	DWT for Contrast Stretched Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	80
3.61	DWT for Contrast Stretched Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT	81
3.62	Global & Local Sharpness image comparison for the preprocessed images (Image size 480×640) (a) Global Sharpness (b) Local Sharpness ($win = 2$) (c) Local Sharpness ($win = 4$) (d) Local Sharpness ($win = 8$) (e) Local Sharpness ($win = 16$) (f) Local Sharpness ($win = 32$). <i>NOTE: Refer to Table 3.2 to understand the legend</i>	87
3.63	Global & Local Sharpness image comparison for Scale-1 DWT (Image size 240×320) (a) Global Sharpness (b) Local Sharpness ($win = 2$) (c) Local Sharpness ($win = 4$) (d) Local Sharpness ($win = 8$) (e) Local Sharpness ($win = 16$)	90
3.64	Global & Local Sharpness image comparison for Scale-2 DWT (Image size 120×160) (a) Global Sharpness (b) Local Sharpness ($win = 2$) (c) Local Sharpness ($win = 4$) (d) Local Sharpness ($win = 8$)	92
3.65	Global & Local Sharpness image comparison for Scale-3 DWT (Image size 60×80) (a) Global Sharpness (b) Local Sharpness ($win = 2$) (c) Local Sharpness ($win = 4$)	94
3.66	Global & Local Sharpness image comparison for Scale-4 DWT (Image size 30×40) (a) Global Sharpness (b) Local Sharpness ($win = 2$)	96
3.67	Global & Local Sharpness image comparison for Scale-5 DWT (Image size 15×20) (a) Global Sharpness	97

ABSTRACT

Visibility is an important part of highway driving. Weather-related visibility limitations have caused several major pile-ups (crashes) over the years all over the country. These limitations are caused due to, but not limited to, fog and rain. It is vital to be aware of the driving environment and conditions when traveling on interstates and highways at speeds upwards of 70 mph. Crashes caused at these speeds usually result in injuries and sometimes death.

In an effort to minimize/reduce the rate of such incidents, this project intends on automatically detecting driving visibility ranges in real time based on camera systems that already exist on the Florida Highway System. This will be done to alert travelers of driving hazards due to visibility limitations. We plan on doing this by performing analysis on images taken over time to detect the density of fog to determine the visibility distance.

In this thesis we provide a proof of concept study of an automatic fog detection system with the intention of designing a sensor that will be integrated with the current camera systems that are implemented on the highways in Florida. Although sensor design is the eventual goal, the focus of this thesis is the image processing part of the sensor, and to that effect most of the study has been done in constructing a proof of concept to determine what would be needed in order to design such a sensor. This thesis intends to investigate automatic fog detection in day-time and night-time images on a standard scale regardless of the image intensity. (i.e. Day-time and Night-time images have the same scale). Images are considered independently with no reference image to compare with. A review of related work is conducted for fog detection and sharpness measure of images to be able to design a sensor that would allow automatic fog detection.

This approach is different from those discussed in chapter 1 since our goal is to design a sensor that requires minimal (if any) modification to the existing camera systems that are already in place. This is a key element to cut cost while not compromising the quality of the analysis. Such a sensor is important as it would aid in lowering if not eliminating accidents that might be caused due to fog and possibly other such weather-related conditions in the future.

CHAPTER 1

BACKGROUND & RELATED WORK

As per the National Highway Safety Administration, the top six most common causes of automobile crashes are; distracted driver, speeding, aggressive driving, drunk driving, driver fatigue, and weather. Each year the number of weather-related accidents seems to rise, despite the improved road infrastructure. Of all the weather-related conditions, fog is considered to be the most dangerous since the visibility distance reduces exponentially as the fog becomes denser. As per the Federal Highway Administration twenty-two percent of crashes were weather-related in 2004-2013 with an average of 5,760,000 vehicle crashes each year. Each year, nearly 5,300 fatal crashes, over 314,600 injuries and almost 929,200 property-damage-only crashes occurred due to non-favorable driving conditions. Weather conditions affect highway driving greatly in terms of safety due to limited visibility. Foggy driving is one of the many weather-related causes of accidents. Several multiple-car crashes have been recorded through the years across the nation. The Federal Highway Administration has recorded 28,533 fog-related crashes during 2004-2013 [1]. In the presence of fog, drivers tend to drive at higher speeds due to the overestimation of visibility [12]. A measurement of available visibility would warn the driver of the vehicle speed or could automatically limit the speed to avoid a potential accident. A recent and fatal multi-vehicle fog crash in January of 2012 on I-75 near Gainesville killed 10 people and 18 severely injured. The reported cause of the accident was limited visibility and lack of suitable traffic control devices. It is for this reason that fog detection has become a very popular topic of research in the recent years. It is a primary cause of injury, property damage and in some cases death. Research and implementation of such devices is discussed in the following section to gain a better insight on some systems that are currently installed and effective on several highways across the nation.

1.1 Fog Detection

Many state transportation departments have implemented extensive fog detection systems that accurately detect fog in terms of unknown depth, weather, and lighting conditions. Fog looks

different in different lighting and also is affected by weather conditions (i.e. harder visibility during rain when fog is present). These complexities related to fog make it a very hard weather condition to detect. These systems require many sensors that are installed into the highway section of concern along with communication devices to exchange information and for off-line computation and analysis of the sensor data. The West Virginia University (WVU) studied ten different state fog detection systems from California, Florida, Alabama, West Virginia, Virginia, Tennessee, Georgia, Utah, South Carolina, and Maryland. These ten studies cover almost all of the major fog detection systems. The systems proved effective and the state saw a reduction in fog-based car accidents in the presence of these systems, however, the implementation cost for such systems is relatively high. Apart from cost, another factor is the amount of hardware and other off-line resources needed in order to have such a system implemented which cuts into the fact that an engineer or a team of engineers is needed to troubleshoot and repair the system. Table 1.1 shows a comparison of fog system implementations in terms of cost and effectiveness based on the ten states mentioned above. The system implemented by WVU comprises of the following over a stretch of 5 miles, four Roadway Weather Information System (RWIS) that monitor the weather and pavement conditions of the different sections of the road, each supporting a different type of meteorological condition as well as pavement sensors which predict the weather condition based on the historical data and real-time data. Each RWIS consists of an Environmental Sensor Station (ESS), remote processing unit (RPU), and communication links for data display and collection. The ESS within the RWIS is the source for data collection of the different types of data such as air, temperature, precipitation, visibility, dew point, relative humidity, wind speed, and direction, surface pavement temperature, subsurface temperature, surface conditions (dry, wet, frozen), the amount of deicing chemical on the road, and freezing point of the road [20]. A full list of all the components is listed in Table 1.2.

The ten weather systems implemented in the different states are further discussed below to provide a more in-depth view of the systems, their operational and hardware cost and the limitations, along with an analysis of how they meet the needs of our goal of a state-wide implementation.

Table 1.1: Fog System information by state along with implementation year and effectiveness as reported by [20]

Location	Year	Cost of Fog System	Effectiveness
Stockon, California	1996	\$2.5million	80% reduction over 3 years
Big Savage, Keyzers Ridge, and Friendsville (Maryland)	2003	\$375,000 (all three locations combined)	<i>data unavailable</i>
I-64 Kanawha River Bridge, West Virginia	<i>data unavailable</i>	\$89,000	<i>data unavailable</i>
I-75 Tennessee-Georgia border	1990	\$4.5 million	<i>data unavailable</i>
I-64 (Virginia)	1997	\$5.3million	reduction in fog-related crashes
Adel, Georgia	2001	\$2.4 million	<i>data unavailable</i>
Salt Lake City, Utah (I-215)	<i>data unavailable</i>	<i>data unavailable</i>	<i>data unavailable</i>
Cooper River Bridge, South Carolina (I-526)	1992	\$5 million	almost 100% reduction reported due to fog
Florida	2010	<i>data unavailable</i>	<i>data unavailable</i>
Alabama	<i>data unavailable</i>	<i>data unavailable</i>	<i>data unavailable</i>

Table 1.2: Fog system components for the WVU Active system

Sensor/Equipment	Quantity
Roadway Weather Information System (RWIS)	4
Variable Message Signs (VMS)	4
Closed Circuit Television (CCTV)	6
Inductive Loop Surveillance	6
Environmental Sensor Stations	4
Highway Advisory Radio	3
Fiber optic cable	based on mileage
Conduit Design and Installation	based on mileage

1.1.1 California Weather System

California DOT implemented an advanced fog detection system called Caltrans Automated Warning System (CAWS). The total cost for the implementation of this system was \$ 2.5 million. This system was installed in Stockton, CA in 1996 in a highly dense traffic area . Caltrans involves the use of 9 stations for measuring environmental parameters, 36 speed detectors, and 9 variable message stations for displaying information to the travelers. Real-time visual data on time and location is transmitted via a network of CCTV cameras. These cameras send real-time video feed to the Transportation Management Center (TMC). Although this system was originally installed in this highly congested area with the goal to reduce traffic and surveil traffic accidents, a three year assessment showed a significant reduction in the number of fog based multi-vehicle accidents. The reason for this reduction is largely due to the VMS messages making drivers more aware of the environment.

A newer type of fog detection and warning system was inspired when a multi-car incident took place outside of the coverage area. This system is installed by Caltrans along I-99 and I-5 near Fresno area. This area is a major roads in the central part of California handling greater than 100,000 vehicles per day. Visibility limitations have been known in this area due to a dense and frequent fog names 'Tule'. This fog reduces visibility distance to a very low extent and sometimes the reported visibility is zero. Following a severe 101 vehicle crash the project was started and installation from start to finish took around 2 years. The system displayed messages to approaching vehicles using VMS after analyzing the road weather and traffic conditions by visibility sensors. The three components of this system were (i) Weather Station, (ii) Speed detectors, and (iii) Motorist Warning System. Data is collected by the various sensors that are part of the Weather Station and processed at the Caltrans TMC, based on the analysis an appropriate message is broadcasted to the VMS and Highway Advisory Radios (HAR). The messages sent to the TMC are done using wireless communication and the system also has the ability to continue working in the event of any disruptions in connection with the TMC. This is achieved by performing on-site analysis in the event communication to the TMC is unavailable. Part of the system includes the installation of visibility sensors at the driver's eye-level at a distance of every half a mile. Traffic congestion and speed is monitored by HD radar spot speed installed every quarter of a mile along the highway. VMS are installed every half mile along the highway on both sides. The system also takes advantage of the

HAR system and provides traffic and weather condition information to 511 travelers' information system, which allows commuters to plan their commute before leaving the house during bad weather conditions. According to preliminary assessments of the system the number of fog related crashes had significantly reduced. The total cost of installation was about \$ 12 million for a 13-mile stretch of highway. This is about 4 times the amount of the older system.

While all the information and implementation mentioned above proves to be a very efficient method for reducing fog related accidents. Having the VMS systems and HAR connectivity is nice and helpful. The output of the system is very user friendly, since commuters can either get this information while driving via the VMS or can get it via the radio on 511 or via the HAR. Although the system has a lot of perks, it is evident that installation of such a complex system is not cost effective on a state-wide level. Firstly, the integration of the system on a 13-mile stretch cost \$ 12 million, secondly the time to implement the entire system from start to finish took 2 years, and lastly the number of sensors (every half mile i.e. 26 sensors) and radars (every quarter mile i.e. 42 radars) installed along the 13-mile stretch implies that the cost of maintenance is going to be high, although this expense is not mentioned in the report. Monetary costs are not the only concern, the amount of resources needed to have such a system in a comparatively small patch of the highway is high, the system needs on-site computing resources in the event of communication disruptions as well as an off-site main processing center and some method of communication (in this case wireless) between the two sites. Although the system has some good features, it is not cost effective for our needs.

1.1.2 Alabama Weather System

This system was installed on a 6-mile stretch of I-10 after several multi-vehicle accidents occurred in the area. The system was integrated with a Tunnel Management System in Mobile, Alabama. The initial setup contained 25 CCTV cameras that reported the ongoing traffic in the area, and 6 visibility sensors (installed every mile) monitoring the visibility of the road. The data was transmitted to the main control unit using fiber optic communication. The system contained predefined standards for the speed and messages would appear on a 24 variable speed limit signs and the 5 VMS boards. The VMS also provided users with real-time information of the weather conditions. In 2008 a major upgrade to the system was made, changing the communication method to a point-to-point system of Ethernets and also Radar Vehicle Detection (RVD) devices were

installed every third of a mile. During low visibility the system was believed to have reduced the number of incidents caused due to fog. The output of the system is shown in a screenshot in Figure 1.1. Although the system is believed to have reduced the number of accidents, the system

Figure 1.1: Alabama DOT low visibility warning system screen shot [20]

requires a lot of components. In terms of hardware for a 6-mile stretch there were 5 VMS boards, 24 variable speed signs, 25 CCTVs, 3 RVD devices, and all required cabling and communication hardware along with power. This in terms of cost translates to a not so feasible system, also the output of the system requires some knowledge and understanding of the GUI shown in Figure 1.1. This added element translates to additional training and/or a specialist required in order to manage and keep the system up-to-date to provide real-time information to the commuters. Hence, it is determined that this system is not fit for our needs.

1.1.3 West Virginia Weather System

The fog detection system is installed on I-64 Kanawha River Bridge, a highly fog prone area. The system consists of two distinct RPU units that make decisions independently, data is collected using infrared technology, with sensors located one mile east and one mile west of the bridge, near St. Albans interchange. Flashers are used instead of VMS, these flashers are activated when a visibility distance of less than 1200 feet is reported, warning the commuters to slow down and be cautious. The system operation is monitored through communication with Vaisal network and the highway operators are notified when the system is triggered. Visual confirmation is also available to the Highways' traffic engineers in the headquarters, this is achieved through the use of video cameras. The installation cost of this entire system is around \$ 89,000. While this system is fairly simple in terms of installation, and is autonomous there is no information available that discusses the performance of the system. Hence it is inconclusive of the effectiveness of this system for reduction in fog based accidents.

1.1.4 Virginia Weather System

Two areas in Virginia experience dense fog, the I-77 Fancy Gap Mountain and I-64 Fancy Gap. Multi-vehicle crashes have been occurring since the highway opened in 1972, in 1977 the DOT decided to implement a pavement lighting system along 5.8 miles of I-64 to guide cars in bad weather conditions. A short term study suggested reduction in fog based incidents, however, increases number of crashes occurred in clear visibility conditions. This rise in crashes was a result of commuters traveling faster due to the improved visibility. In 1997 the system was upgraded drastically costing \$ 5.3 million for the same 5.8 mile stretch, this was done since fog related crashes still existed in the area. The components and effectiveness of this system are not available, however, in 2002 recommendations had been made by a team of experts to incorporate VMS, video cameras, variable speed limits signs, supplementing strobes and lasers to VMS and a few other recommendations were made to improve driving conditions for motorists. The fact that further recommendations were provided suggests that the system is inefficient in reduction of fog related crashes, hence unfit for our purposes.

1.1.5 Tennessee Weather System

The system installed in Tennessee is actually on I-75 across the Tennessee-Georgia border. This system was installed after a fog based accident occurred in 1990 involving 99 vehicles. The installation covers about 19-miles of the highway which is considered to have a fog problem. Weather conditions in this area change almost instantly decreasing the visibility range to less than 10 feet. Driver's sight of visibility is checked by two weather forecasting stations and eight fog detection sensors. The weather stations measure the temperature, wind speed and directions, and dew points. Fog is classified in four categories:

1. **Clear** no visual impairment
2. **Moderate** moderate visual impairment
3. **Severe** severe visual impairment
4. **Critical** critical visual impairment

Based on the density of the fog, one of the following four messages appears to alert the motorists. The system is also able to update the speed signs based on the severity of the fog. This entire system cost \$ 4.5 million at the time of installation as per the Tennessee ITS State Report 2000 [20]. Apart from the high cost of installations, there is no reported performance statistics of this system's effectiveness in reducing fog based incidents, hence we classify this system as not fit for our needs.

1.1.6 Georgia Weather System

In Adel, Georgia an augmented fog and smoke detection and warning system was implemented along a 19 mile stretch costing \$ 2.4 million. The data collected by the various sensors along this patch of the highway were transmitted via fiber optics to a local on-site computer for analysis. The data is fully processed and a message is transmitted to the VMS accordingly. The data is also transmitted to the GDOT officials for monitoring and remote control purposes of the system. Four levels of visibility ranges were defined with the possibility of a interstate closure in extreme weather conditions. The effectiveness of this system is not provided and not sufficient information is provided to conclude if the system meets our requirements.

1.1.7 Utah Weather System

In Salt Lake City, Utah the UDOT implemented a low visibility detection and warning system along a two-mile stretch of I-215 due to a recurrent fog occurrence. Like many other systems visibility was measured using visibility sensors installed along the highway and the data was transmitted to the commuters by the DMS. The goal of the system was to reduce the speed of the commuters to achieve a steady stream of traffic, this was done by the use of two DMS systems which sent out the proper messages to the commuters warning them of the existing weather conditions. An Adverse Visibility Information System Evaluation (ADVISE) system was installed by UDOT and Utah Traffic Lab (UTL) to study the effectiveness of the system. The study revealed that the system was effective in reducing the driving speed of the commuters by 15%, however no data is provided regarding the reduction of fog-based incidents. The output of this system was displayed to VMS systems installed and conveyed to the commuters directly which makes it a desirable system in terms of autonomous fog detection, however, not enough information is available to be certain of this decision, hence this system is not considered fit for our needs.

1.1.8 South Carolina Weather System

On I-526 in Cooper River Bridge, SC a fog detection and warning system was installed over a stretch of about 7 miles. The primary goal was to inform commuters of weather conditions through advisory speed limits and warning messages. The system consists of five visibility sensors, eight CCTV cameras, and eight VMS boards. The information from the sensors and cameras is transmitted to an on-site RPU as well as an off-site central computing unit. The data link to the off-site location is achieved using fiber optics. Studies suggested that no major incidents occurred after the implementation of the device and improvement in travelers' safety in foggy conditions was observed. The initial cost of the entire system for the 7-mile stretch was \$ 7 million. The system is similar in components to the system that was implemented by Alabama, and the system seems to be effective in reduction of fog based accidents, however, due to the number of components needed for the system it is not feasible on a state-wide implementation scale.

1.1.9 Maryland Weather System

In Maryland, the most hazardous fog areas are considered to be Big Savage, Friendsville, and Keyzers Ridge. There are fog detection and warning systems installed in these areas for the purpose

of detecting fog and warning commuters along I-68 of the weather conditions. These areas were chosen based on the type and density of fog, available infrastructure, highway profile and line of sight. The state decided to install this system in response to a 90 vehicle accident that occurred in 2003. It is the first of its kind in the state of Maryland having the capability of identifying three levels of visibility and notifying commuters prior to entering the fog area, expedite installation the system is designed to take advantage of the existing infrastructure. Weather and pavement condition information is gathered using a RWIS which consists of precipitation sensors, wind sensors which measure wind speed and direction, air temperature, humidity and visibility sensors. An on-site RPU performs preliminary analysis on the data, the RPU is capable of using this data to control the warning signs directly via a speed spectrum radio connection. Commuters do not get information on the severity or the density of the fog via a VMS-like board, however, they are alerted by a Fog sign with flashing lights that advises them to practice caution due to the presence of fog. The collected data is also transmitted to the SHA District 6 operation center for additional processing and data backup. The communication between the site and SHA is done using a dial-up telephone. The system is run on solar power since AC power was not available, the total cost for all three locations combined is \$ 375,000. Although this system is a possible solution due to the low cost of implementation, as well as the fact that it does not require AC power, and is self sufficient through the use of solar power, it is not acceptable for our purposes due to the fact that the system only outputs a warning sign with no actual information on the severity of the fog and or the driving conditions.

1.1.10 Florida Weather System

Florida is reportedly among the top states in the country in dealing with fog related problems. Several systems are in place throughout the state, however, it is still suffering from fog related crashes. In 2012 a fatal multi-vehicle crash near Gainesville on I-75 caused 10 deaths and 18 severe injuries, the reported cause of this incident were limited visibility due to fog and lack of suitable traffic control devices. The first studies in Florida started with the Tampa Bay area which saw 829 fog related accidents between 1987-1995. The study showed that there was no specific fog prone site in Tampa bay and recommended focusing on a driver awareness campaign instead of the installation of a fog-detection system as a economical method to reduce fog related accidents.

In 2010, however, University of Central Florida came up with a system called Early Detection System for Reduced Visibility which was installed. The system comprised of inexpensive and available components and could be installed as a fixed or portable system using DC power instead of AC power. The system consisted of four different stations that were connected to visibility sensors along with a unit that controlled the system operations. The communication between components within the system was done using an internal communication link, while communication with the main control unit was done using a cellular communication. Visibility conditions are monitored continuously and the measured data is transmitted to the base station every 15 minutes. This is done to ensure proper working of the system and all the components. In the presence of fog, which is determined by predefined thresholds, the base station sends out proper messages to output units such as DMS, it is also capable of sending out emails to the TMC. Although the system is proven to detect fog efficiently there isn't enough information to determine if the system effectively reduces fog related accidents.

1.1.11 New Jersey DOT

Similar to the WVU fog detection system, New Jersey Department of Transportation (NJDOT) did a similar implementation that consists of several sophisticated sensors. This fog detection system was implemented by NJDOT on the Wanaque Bridge on Route 287 which is considered an area that is affected by adverse weather conditions. This system consists of a weather detection system that can generate alarms for low visibility conditions, in-road temperature sensors which are capable of generating alarms for adverse road surface conditions due to low temperatures, CCTV for visually assessing the weather conditions and also verification of traffic accidents, traffic sensors for monitoring the traffic and for automatic detection of incidents and variable message signs that are used to inform drivers of weather warnings and reduced speed limits[15]. NJDOT detects fog under three conditions:

1. Daylight, with no lights on the target vehicle or the observer's vehicle
2. Nighttime, taillights, and headlights are on for both vehicles
3. Daylight, taillights, and headlights are on for both vehicles
 - (a) The taillights of the leading car is seen first
 - (b) The body of the leading car is seen first

Case 1 uses Koschmieder's Theory of Air Light

Definition 1.1.1. Koschmieder's Theory of Air Light (aka Koschmieder's Law) is law relating the apparent contrast C_d of an object against a sky background, at a given distance of observation d , to the inherent contrast C_0 and to the atmospheric transmissivity T , which is assumed to be uniform $C_d = C_0 \times T^{d/d_0}$, where d_0 is the length specified for the definition of T .

Case 2 relies on background luminance and object luminance using Allard's Law to determine visibility.

Definition 1.1.2. Allard's Law law relating the illuminance E produced on a surface by a light source to the luminous intensity I of the source in the direction to the surface, to the distance d between the surface and the source, and to the atmospheric transmissivity T , which is assumed to be uniform; the surface normal to the direction of the surface and sufficiently distant for the source to be considered as a point source $E = (I/d^2) \times T^{d/d_0}$, where d_0 is the length specified in the definition of T

The visibility in Case 3a is determined similarly to case 2 using Allard's Law. While in case 3b the visibility is measured similarly to case 1 since no change is measured in the background luminance when headlights are on/off during daytime. Although these fog detection systems prove to be effective in the reduction of fog-related incidents, the cost of these projects is relatively high for a state-wide implementation. Recent research has been geared more towards fog-detection using only on-board sensors and cameras as seen in [12][3][21][2]. These methods use only a single camera or sensor reducing the cost to the fraction of the amount usually less than a thousand dollars versus the state-implemented systems which range from several thousand to millions of dollars.

1.1.12 Other Studies

Hautière [12] estimates fog visibility using a dynamic implementation of Koschmieder's Law. This approach requires the use of a single on-board camera and the presence of the road and sky in the scene and is limited to only day-time images. Region growing procedures for detecting inflection point in the image is used to detect if fog is present. The visibility distance is then computed once the inflection point and the horizon line is known in the image.

In Bronte et. al. 2009 [3] computation of the vanishing point is used to detect fog using the road lines as the reference point using an on-board camera. This process starts with the acquisition of the image, using Sobel filtering it is classified as sunny/foggy. If the image is foggy then optimal binarization is applied to the Sobel filtered image to enhance the clear edges. Computation of the vanishing point is done using the parallel lines of the road that cross the horizontal. Using this vanishing point the road/sky segmentation is performed using a region growing process from which the road and sky limits are estimated. From this image finally, the visibility is determined and classified as low, moderate, or high fog based on threshold that are manually set. The fog detected is segmented into three stages, low fog > 300 m, $300 \text{ m} > \text{moderate fog} > 100$ m, and high fog < 100 m.

Negru [21] uses grey-scale images, their process detects fog and provides recommendations of driving speeds for a given road segment. The image is first processed using Canny-Deriche edge detector. After which the horizon line and inflection points are estimated to assess for the presence of fog. If fog is present, visibility estimation and speed recommendations. Their approach divides fog into five categories, the categories are no fog, low fog, moderate fog, dense fog, and very dense fog. These categories are based on visibility distances with minimum distances as 1000, 300, 100, 50, and 0 meters, respectively and maximum visibility distances as ∞ , 1000, 300, 100, and 50 meters, respectively. The techniques mentioned in this section detect fog using a single-camera, however, the camera in these cases is mounted physically on the car, whereas our goal is to use information from a single-camera that is mounted highway and/or VMS/signboards.

A more sophisticated and statistical approach is performed by Kwon [17]. The basis of the analysis is Koschmeider's Law however, they use 9 fog aware features to effectively design a Fog Density Prediction Model. This model uses a set of 160 foggy and 160 fog-free images to form the basis of the model. The 9 fog aware features are

1. Variance of Mean Subtracted Contrast Normalized (MSCN) coefficients
2. Variance of the vertical product of MSCN coefficients
3. The sharpness
4. Coefficient of Variance of Sharpness
5. Michelson contrast

Figure 1.2: Fog Density Prediction Model

6. Image Entropy
7. Dark channel prior in a pixel-wise
8. Color saturation in HSV color space
9. Colorfulness

The model extracts the fog aware statistical features and fits them to a multivariate Gaussian (MVG) model using the 160 foggy and 160 fog-free images. A test image is then processed to compute a Mahalanobis-like distance measure between the MVG fit and the statistical features of the fog aware features of the 160 foggy and 160 fog-free images, respectively. Each level is described as a foggy or a fog-free level with the final fog density displayed as a ratio of the foggy level to the fog-free level of the test image. The sharpness measure used in their implementation is based on the variance of the image. Figure 1.2 shows the prediction model that is discussed in [17]. Although this prediction model's performance is desirable, there are a few limitations that are inherent to this model. Since the basis of this model is the Koschmeider's Law, this method is only effective for day-time images, which is a significant drawback since we intend on creating a sensor that detects fog independent of intensity. Another drawback of this system is that it uses a sample data set of foggy and fog-free images. This is not desirable since there are several different types of fogs and they exist in different densities based on the weather conditions so accurately simulating and accounting for all the various different types of fog scenarios is relatively hard.

Most of the fog detection and warning systems studied are effective in the reduction of fog-related accidents using a spectrum of sophisticated sensors, however, our goal is to design a sensor that is capable of integration with the existing infrastructure and uses only a single-camera as the input information for detecting fog. Systems explained in Section 1.1.12 use the methodology that we intend on using however, the difference is that images we use are based on cameras that are mounted on the sides of the highway, this means that our images may or may not have the horizon line present in the image causing the horizon line detection approach to fail. Also the Koschmieder's Law uses atmospheric luminance which means that the approach is only valid for the day time images requiring ample ambient lighting and fails during night images. These techniques would fail for the sensor we intend on designing since we plan on detecting fog during the day as well as night, for this purpose we analyze a few different techniques. We perform a few different preprocessing techniques and we also apply a 5-scale wavelet decomposition to the images to see how the sharpness of the image is affected. The computed sharpness of the images is a scalar value that will hypothetically help us in determining the density of the fog which will be used to alert commuters of the driving conditions in the area/areas of concern. Each of the analysis parts is explained in detail in the following chapter.

CHAPTER 2

VISIBILITY OF RANGE DETECTOR USING A SINGLE CAMERA

The fog detector proposed in this thesis is as shown in Figure 2.1, we start by first obtaining the images and running some pre-processing techniques to enhance the contrast of the images. Once the images are pre-processed a discrete wavelet transformation is applied to each image five times. Finally, we compute the global and local sharpness for all the images (i.e. the original set of images as well as the five transformed set of images). Each set contains four images, day clear, day fog, night clear and night fog image.

Figure 2.1: Fog Detector workflow

2.1 Preprocessing

Preprocessing is an important step in any image processing/analysis. There are several techniques used for pre-processing images. Preprocessing is a branch of image processing that deals with manipulating the image prior to extracting any sort of relevant information. Preprocessing techniques usually affect the image contrast and the histogram of the image. Several techniques exist, for example sharpening algorithms that are used for de-blurring images, or blurring algorithms that are used to soften the image visually. The techniques used for our application are in the group of Normalization. Normalization is a technique that manipulates the pixel intensity values, effectively expanding the range of values. Figure 2.3 shows the visual differences of running the different preprocessing techniques. Normalization is a commonly applied preprocessing technique in image processing. It is usually applied to images with poor contrast due to glare or lighting effects. In the signal processing domain as a whole it is commonly also referred to as dynamic range expansion. This is applied to the signal (in our case images) to bring the image into a range that is more "normal" to the scene, hence the name normalization. Another way of looking at normalization is by thinking of it as a scaling process. Images taken at different times generally have different scale (i.e. pixel values) normalizing/scaling these pixel values allows us to compare them on a standard scale instead of have a scale that is unique to each image, we normalize so that the same scale or set of scales can be used across the images. This concept is better understood by example. Given the two 8-bit gray scale images below which are of the same scene taken at two different times of the day. We will refer to these as image 1 (Figure 2.2(a)) and image 2 (Figure 2.2(b)). Image 1 has pixel values ranging from 6 (minimum pixel value) to 253 (maximum pixel value), while image 2 has pixel values ranging from 89 (minimum pixel value) to 253 (maximum pixel value). Since the two images have different ranges of pixel values, the comparison done on these images would require the use of two independent scales. Although scales differ from image to image based on the ambient lighting, this effect is more visual when analyzing day and night images. Since the objective of this research is to device a sensor that outputs a single value that notifies travelers of the fog intensity, normalization plays a key role in the analysis of the data since we want both day and night images to be analyzed on the same scale. This is discussed later in the thesis as the calibration problem. In order to be able to tackle the calibration problem we use the following four techniques for our preprocessing, the inverse transform, logarithmic transformation,

power law, and contrast stretching. These methods are explained in more detail in the following sections followed by image samples for each of the methods in Figure 2.3.

(a)

(b)

Figure 2.2: Normalization (a) Day fog image (b) Night fog image

(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.3: Preprocessing technique Examples (a) Original Image (b) Inverse Image (c) Log Transform (d) Power Law($\gamma = 0.3$) (e) Power Law ($\gamma = 4$) (f) Contrast Stretching

Figure 2.4: Histograms for the previously displayed images on page 18 (a) Original Image Histogram (b) Inverse Image Histogram (c) Log Transform Histogram (d) Power Law Histogram ($\gamma = 0.3$) (e) Power Law Histogram ($\gamma = 4$) (f) Contrast Stretching

2.1.1 Inverse Transformation

As the name suggest this transformation basically inverts the pixels (i.e. black to white and white to black). This is a simple inversion that leaves the histogram unchanged, however it is reflected about the x-axis in comparison to the original histogram, this reflection can be seen by comparing Figure 2.4 (a) and (b). The inverse transformation is computed based on the pixel values. Since we use an 8-bit gray-scale image the pixel values range from 0-255. Therefore to compute the inverse the following formula is used:

$$\mathbf{s} = 255 - \mathbf{r} \quad (2.1)$$

where $\mathbf{r}$ is the input image of M -by- N dimensions and $\mathbf{s}$ is the output image of the same size as the input. The output image $\mathbf{s}$ is computed by subtracting each pixel value in $\mathbf{r}$ by $2^n - 1$, where $n = \#$ of bits. So for an 8-bit gray-scale image $n = 8$. This transformation is also known as the negative image transformation which can be seen in Figure 2.3(b). This type of image is applied usually to images to enhance the white or gray detail that might be otherwise unseen in dark regions of an image.

Figure 2.5: Some basic intensity transformation functions. Curves are scaled to fit in the range shown [11]

2.1.2 Logarithmic Transform

Logarithmic transformation is given by the (2.2)

$$\mathbf{s} = c \times \log(1 + \mathbf{r}) \quad (2.2)$$

where $\mathbf{c}$ is a constant, and $\mathbf{r}$ is the input image pixel values of size M -by- N which is greater than or equal to 0, $\mathbf{s}$ is the output of the transformation same in size as the input image. The shape of the log curve can be seen in Figure 2.5 which shows that a narrow range of input values are expanded into a wider range of output values. This transformation works well for low intensity input values, if the original input contains high intensity pixel values then an inverse log transformation is applied to expand the range of narrow input values into an output of wide range values. An important characteristic of the log function is that it compresses the dynamic range of images with large variations in pixel values. An example of this transform can be seen in Figure 2.3(c) and the corresponding histogram can be seen in Figure 2.4(c).

2.1.3 Power-Law (Gamma) Transformation

The Power-Law Transformation is similar to the log transformation, however, they are much more versatile in terms of the expansion/compression of pixel value ranges. The Power-Law equation is of the form:

$$\mathbf{s} = \mathbf{c} \times \mathbf{r}^\gamma \quad (2.3)$$

where $\mathbf{c} = \mathbf{1}$ for simplicity, however, $\mathbf{c}$ can be any constant, and $\mathbf{r}$ is the input image of M -by- N dimensions and $\mathbf{s}$ is the output image of the same size as the input, γ is a constant based on Figure 2.6. Similar to the log transformation mentioned in Section 2.1.2 for $\gamma < 1$ the power-law transformation maps narrow ranges of dark input values into a wider range of output values, with the inverse being true for $\gamma > 1$ and higher input values. Contrary to the log transformation, the power-law transformation is a lot more versatile, since a range of different curves can be obtained (Figure 2.6) just by modifying the value of γ . An example of this transform can be seen in Figures 2.3(d-e) for γ values of 0.3 and 4, respectively. Also the corresponding histograms can be seen in Figures 2.4(d-e) for γ values of 0.3 and 4, respectively. A special case of the power-law is when $\mathbf{c} = \gamma = \mathbf{1}$ at which point the output image is an exact replica of the input image. This case is not displayed explicitly since the transformed image and its histogram is identical to the original image, implying that no change has taken place by applying the transformation.

Figure 2.6: Generic Power-Law Transformation plot for different values of γ with $\mathbf{c} = \mathbf{1}$ [11]

2.1.4 Contrast Stretching

Contrast stretching, also known as normalization is a piece-wise linear transformation that increases the contrast of the image. It expands the range of intensities to span the full intensity

range from 0 to $L-1$, where $L = 2^n$ and n represents the number of bits. For an 8-bit image the full range of intensities is from 0 to 255. It also preserves order of the intensity levels preventing the creation of any artifacts that are common in processed images. This profile of the transform function is based on the low and high limits of the input and output image. These limits are the location points, let (r_1, r_2) be the low and high points of the input image and (s_1, s_2) be the low and high points of the output image. If $r_1 = s_1$ and $r_2 = s_2$ then the transformation is linear and doesn't modify the image. If $r_1 = r_2 = k$, $s_1 = 0$ and $s_2 = L - 1$, then the transformation becomes a thresholding function, this profile can be seen in Figure 2.7(b). If $r_1 = r_2$, this is a special case on contrast stretching and (2.4) does not apply, the equations for this special case are simple, all values less than the threshold are set to 0 and all values greater than the threshold are set to $L - 1$, the user can define how the threshold value is set. Figure 2.7(c) shows a typical transformation function used for contrast stretching.

Figure 2.7: Intensity Transform Functions. (a) Contrast Stretching function. (b) Thresholding function. (c) Form of transformation function. [11]

In general $r_1 \leq r_2$, $s_1 \leq s_2$, $r_1 \neq s_1$, and $r_2 \neq s_2$ to obtain a monotonically increasing transformation function. These values for the low and high values of the input and output can be defined by user input or by an automated process. If the system needs to be automated, selection of the low and high values for the contrast stretching function becomes challenging, however, it is still possible with the introduction of an additional variable, let the variable be tol , which will be discussed a bit later in this section. For the purposes of explanation we will use the values defined in Table 2.1 to show how to compute the values by example. Note: for computational purposes the

matrix size is small (i.e. 10×10), however, all images of the highway scene are of size 480×640 , unless otherwise specified.

Table 2.1: Contrast Stretching example parameters

Variable	Dimension or Value	Description
n	8	number of bits
$\mathbf{r}$	10×10	Input Image
elems	100	Total number of elements
L	256	$2^n = 256$ i.e. $n = 8$ -bit image
tol	[1 10]	tolerance values
r_1	$\mathbf{r}[tol]$	Input image's lower limit
r_2	$\mathbf{r}[100 - tol]$	Input image's upper limit
s_1	0	Output image's lower limit
$s_2 = L - 1$	255	Output image's upper limit
$\mathbf{s}$	10×10	Output Image

Matrix $\mathbf{r}$ is defined as follows for our purposes, the values chosen here are random between the range of 0-255 and have no relation to the actual set of images:

$$\mathbf{r} = \begin{bmatrix} 108 & 135 & 199 & 132 & 65 & 234 & 45 & 69 & 165 & 117 \\ 24 & 167 & 108 & 241 & 57 & 3 & 184 & 195 & 173 & 169 \\ 153 & 104 & 23 & 163 & 170 & 118 & 121 & 48 & 162 & 196 \\ 120 & 209 & 68 & 244 & 215 & 108 & 39 & 73 & 241 & 89 \\ 177 & 183 & 39 & 61 & 88 & 118 & 87 & 23 & 53 & 169 \\ 178 & 247 & 72 & 172 & 199 & 196 & 155 & 147 & 181 & 106 \\ 163 & 135 & 112 & 74 & 172 & 82 & 49 & 174 & 60 & 215 \\ 9 & 83 & 134 & 171 & 3 & 200 & 188 & 139 & 30 & 212 \\ 18 & 27 & 117 & 177 & 154 & 120 & 62 & 109 & 155 & 65 \\ 81 & 156 & 223 & 17 & 99 & 9 & 234 & 164 & 115 & 156 \end{bmatrix}$$

To obtain maximum contrast stretching, the output values (s_1, s_2) should be set to $(0, 255)$, respectively. The input values are set based on a tolerance threshold. The role of the tolerance is to saturate equal fractions of the image at the low and high pixel values. So $tol = 10\%$, translates to saturating 20% of the input image, 10% from the lower intensities and 10% from the higher intensities, it maps these values to the output low and high (in our case 0 and 255, respectively). Which means that the output image will be a higher contrast image of the middle 80% of the input image. From this we can find the (r_1, r_2) values by converting the matrix into a vector and looking at the 11^{th} element and the 90^{th} element, this is the case since we have a 10×10 image, which

has a total of 100 elements. For our actual image the image size is 480×640 which means the total number of elements is 307,200 so for a $tol = 10\%$ we would have to look at the 30721^{st} and 276480^{th} element. The resulting (r_1, r_2) for the two mentioned tolerance values from Table 2.1 will then give us the following:

$$when, \mathbf{tol} = \mathbf{1} \quad then \mathbf{(r_1, r_2) = (3, 244)}$$

$$when, \mathbf{tol} = \mathbf{10} \quad then \mathbf{(r_1, r_2) = (30, 209)}$$

Once these limits are set, the output image can be computed using a simple interpolation method using (2.4).

$$\mathbf{s} = (\mathbf{r} - r_1) \frac{s_2 - s_1}{r_2 - r_1} + s_1 \quad (2.4)$$

where, $\mathbf{r}$ and $\mathbf{s}$ are input and output images of size M-by-N, and (r_1, r_2) are the input values and $r_1 \neq r_2$ and (s_1, s_2) are the output values, low and high pixels, respectively. By this process using $tol = 10$, and input matrix $\mathbf{r}$ would give us the following output:

$$\mathbf{s} = \begin{bmatrix} 111 & 150 & 241 & 145 & 50 & 255 & 21 & 56 & 192 & 124 \\ 0 & 195 & 111 & 255 & 38 & 0 & 219 & 235 & 204 & 198 \\ 175 & 105 & 0 & 189 & 199 & 125 & 130 & 26 & 188 & 236 \\ 128 & 255 & 54 & 255 & 255 & 111 & 13 & 61 & 255 & 84 \\ 209 & 218 & 13 & 44 & 83 & 125 & 81 & 0 & 33 & 198 \\ 211 & 255 & 60 & 202 & 241 & 236 & 178 & 167 & 215 & 108 \\ 189 & 150 & 117 & 63 & 202 & 74 & 27 & 205 & 43 & 255 \\ 0 & 76 & 148 & 201 & 0 & 242 & 225 & 155 & 0 & 255 \\ 0 & 0 & 124 & 209 & 177 & 128 & 46 & 113 & 178 & 50 \\ 73 & 179 & 255 & 0 & 98 & 0 & 255 & 191 & 121 & 179 \end{bmatrix}$$

It is evident by this process that the output image spans over the entire range of values (i.e. 0 - 255) where as the input image has a pixel value range of (3 - 247) for our example. Figures 2.8(a), 2.8(b), and 2.8(c) show the contrast stretched images for an image of size 480×640 , and Figures 2.8(d), 2.8(e), and 2.8(f) show the histograms of the images and 2.8(g), 2.8(h), and 2.8(i) show the transformation profiles for each. Figure 2.8(g) is just the original image showing a linear transformation where $r_1 = s_1$ and $r_2 = s_2$.

Figure 2.8: Comparison of contrast stretching method using different tolerance values. The first row is all the images, the second row displays the histograms for the respective images, and the third row is the transformation profile applied to the image. Parameters for the images are: (a) Original Image (b) $\text{tol} = 1\%$, $(r_1, r_2) = (56, 248)$, $(s_1, s_2) = (0, 255)$ (c) $\text{tol} = 10\%$, $(r_1, r_2) = (89, 249)$, $(s_1, s_2) = (0, 255)$

2.2 Wavelet Transform

Discrete Wavelet Transform (DWT) is a wavelet transform used in numerical analysis as well as functional analysis on any discrete input signals. Signals can be of any dimensions, i.e. audio signals which are considered one dimensional, images which are considered two dimensional, or videos sequences which are considered three-dimensional. Wavelet transforms are similar to Fourier Transforms in many ways but a key advantage of using Wavelet transforms is that it contains both

frequency and location-in-time information. Wavelet transform can be computed using a wavelet profile, or by using high and low band coefficients. The different wavelet profiles are shown in Figure 2.9. Figure 2.9(a) shows the Haar wavelet profile which is the first DWT. The Haar Wavelet transform stores the difference and passes the sum for an input that is represented by 2^n numbers. This is a recursive process that pairs $2^n - 1$ differences and a final sum. Figure 2.9(b) shows 9 different wavelets that are called the Daubechies wavelet. This is a newer wavelet and are more commonly used since they create progressively finer samples of the original wavelet function, which is twice the resolution of the previous scale. The first one is the Haar wavelet and the other 9 are shown in Figure 2.9(b), many other wavelets do exist.

Figure 2.9: Wavelet Profiles (a) Haar Wavelet (b) Various Daubechies Wavelets

2.2.1 Wavelet Series Expansion

Wavelet Series Expansion (WSE) for a function $f(x) \in L^2(\mathbf{R})$ in relation to the wavelet $\psi(x)$ and the scaling factor $\varphi(x)$ can be defined by the following formula:

$$f(x) = \sum_k \mathbf{c}_{j_0}(k) \varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty} \mathbf{d}_j(k) \psi_{j,k}(x) \quad (2.5)$$

where j_0 is an arbitrary scale index, and $\mathbf{c}_{j_0}(k)$ are real-valued approximation/scaling coefficients and $\mathbf{d}_j(k)$ are real-valued detail/wavelet coefficients. These coefficients can be defined as follows:

$$\mathbf{c}_{j_0}(k) = \langle f(x), \varphi_{j_0,k}(x) \rangle = \int f(x) \varphi_{j_0,k}(x) dx \quad (2.6)$$

$$\mathbf{d}_j(k) = \langle f(x), \psi_{j,k}(x) \rangle = \int f(x) \psi_{j,k}(x) dx \quad (2.7)$$

2.2.2 Discrete Wavelet Transform in one-dimension

Mathematically WSE and DWT are similar, however, DWT uses discrete inputs, while WSE uses continuous inputs. An example of DWT for the function $f(n) = f(x_0 + n\Delta x)$ for some $x_0, \Delta x$, and $n = 0, 1, 2, \dots, M - 1$, the wavelet coefficients for the function $f(x)$ defined by (2.6) and (2.7) become the forward DWT coefficients for the sequence $f(n)$:

$$\begin{aligned} W_\varphi(j_0, k) &= \frac{1}{\sqrt{M}} \sum_n f(n) \varphi_{j_0, k}(n) \\ W_\psi(j, k) &= \frac{1}{\sqrt{M}} \sum_n f(n) \psi_{j, k}(n) \quad \text{for } j \geq j_0 \\ f(n) &= \frac{1}{\sqrt{M}} \sum_k W_\varphi(j_0, k) \varphi_{j_0, k}(n) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_k W_\psi(j, k) \psi_{j, k}(n) \end{aligned} \quad (2.8)$$

Here W_φ is the scaling coefficient expansion and W_ψ is the wavelet coefficient expansion correspond to the WSE as explained in Section 2.2.1. In (2.8), we let $j_0 = 0$ and select M to be a power of 2 ($M = 2^J$) so that the summations are performed over $n = 0, 1, 2, \dots, M - 1$, $j = 0, 1, 2, \dots, J - 1$, and $k = 0, 1, 2, \dots, 2^j - 1$. Also note that the integrations of the WSE have been replaced by $1/\sqrt{M}$ normalizing factor has been added to the equations.

2.2.3 Discrete Wavelet Transform in two-dimension

DWT in 2-D is no different from the 1-D process. In this case the 2-D signals (inputs) are fog images that are used for analysis. In 2-D the scaling and/or approximation function is $\varphi(x, y)$ instead of $\varphi(x)$, however the detail and/or wavelet function is a combination of three two dimensional functions which are $\psi^H(x, y)$, $\psi^V(x, y)$, and $\psi^D(x, y)$. Each of these function is a product of two 1-D functions, and excludes any products that result in a one-dimensional result such as $\varphi(x)\psi(x)$. The remaining products consist of the following functions that are called the separable scaling function (eq. 2.9) and the directionally sensitive wavelets eqs. 2.10 to 2.12):

$$\varphi(x, y) = \varphi(x)\varphi(y) \quad (2.9)$$

$$\psi^H(x, y) = \psi(x)\varphi(y) \quad (2.10)$$

$$\psi^V(x, y) = \varphi(x)\psi(y) \quad (2.11)$$

$$\psi^D(x, y) = \psi(x)\psi(y) \quad (2.12)$$

H, V, D in Eqs. 2.10 to 2.12 represent horizontal, vertical, and diagonal directions, respectively, which represent the directional sensitivity of the wavelets. This directional sensitivity is introduced as a natural consequence of the separability feature of the 1-D to 2-D analysis. The wavelets measure variations in intensities for the images along the respective directions. Extending the 1-D into separable two dimensions is done by defining the scaled and translated basis functions as

$$\varphi_{j,m,n}(x, y) = 2^{(j/2)} \varphi(2^j x - m, 2^j y - n) \quad (2.13)$$

$$\psi_{j,m,n}^i(x, y) = 2^{j/2} \psi^i(2^j x - m, 2^j y - n) \quad i = H, V, D \quad (2.14)$$

Therefore, the DWT of an image $f(x, y)$ of size $M \times N$ is given by the following equations

$$W_\varphi(j_0, m, n) = 1/\sqrt{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \varphi_{j_0, m, n}(x, y) \quad (2.15)$$

$$W_\psi^i(j, m, n) = 1/\sqrt{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \psi_{j, m, n}^i(x, y) \quad i = H, V, D \quad (2.16)$$

Figure 2.10 shows the analysis filter bank for a 1-scale 2-D DWT process in a block diagram form. This process can be chained at the $W_\varphi(j, m, n)$ output to compute the same for a multi-scale 2-D DWT process.

Figure 2.10: Analysis filter bank showing the input and output of a 1-scale 2-D DWT (image used from [11])

Figure 2.11: Example image (a) original image (b) 1-scale DWT (c) 2-scale FWT (d) 3-scale DWT [11]

2.3 Sharpness Measurement

Sharpness measurement of the image provides information on the existence of fog. The sharper the image the lower the fog. Sharpness is estimated using gradients, this method is very basic and simple to implement, yet very practical. Gradient or the first derivative of an image is a way of determining the rate of change in the image, that points in the direction of greatest increase of a function. In image processing the gradient is implemented using the magnitude of the gradient. Since images are two-dimensional the gradient of $\mathbf{r}(x,y)$ is given as a two-dimensional column vector

$$\nabla \mathbf{r} = \text{grad}(\mathbf{r}) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{r}}{\partial x} \\ \frac{\partial \mathbf{r}}{\partial y} \end{bmatrix} \quad (2.17)$$

where $\mathbf{r}$ is the input image, and (g_x, g_y) are the directional column vector of the input image $\mathbf{r}$. The magnitude then of $\mathbf{r}$ is denoted as $\mathbf{M}(x,y)$ and is called the gradient image, where

$$M(x,y) = \text{mag}(\nabla \mathbf{r}) = \sqrt{g_x^2 + g_y^2} \quad (2.18)$$

$M(x,y)$ is the same size as the input image $\mathbf{r}$. We measure the global sharpness as well as the local sharpness within the image and compare the results to analyze the images. The output of this process is the final output of the sensor. This scalar value is what describes the density of fog present in the atmosphere. Given a reference scale the system can compare the output of the sensor to the scale and determine the density of fog present on the highway. This output makes it possible to make the sensor an automated system, eliminating the need for any manual interaction to alert travelers of the weather conditions.

2.3.1 Global Sharpness

Global sharpness computes the directional contrast change of the entire image, as mentioned above. This process uses the size of the input image as the window size (in our case image size = window size = 480×640). The directional change in contrast of the image is more commonly known as the gradient. Figure 2.12 shows an example of the directional gradient of an image. As it

Figure 2.12: Gradient Image Example: The left image is the actual image. The image in the middle is the horizontal gradient image, and the image on the right is the vertical gradient image [22].

can be seen in the image, the gradient is high when moving from black-to-white or white-to-black pixels, while it is constant and flat when the image pixel intensity is the same. We see this in the top-left blob of black pixels, we can see that both directional gradients represent this blob by flat gray pixel. However there are changes in the directional gradients when the image contains a high contrast. This is the result of computing the gradient of an image. These gradients correspond to a specific pixel, an image returns a matrix of such values from which the sharpness value is

computed using (2.17) and (2.18). We further investigate sharpness by computing the sharpness of the image on a local scale. We do this so that the global and local sharpness can be compared to determine which sharpness method will give a more precise measurement metric that can be used to determine the existence of fog in an image.

2.3.2 Local Sharpness

Local Sharpness uses the same set of equations, however, instead of computing the gradient of the entire image at once, the image is broken up into smaller sub-images. This is done by defining a window size that represents how big each sub-image will be. Another way to think about this is taking the sharpness measure of several sub-images that are pieces of the bigger image and then combining the results to obtain a final sharpness value. Selecting windows to compute the sharpness is important since different window sizes and shapes make a difference in how the sharpness is computed. We further discuss this to provide clarity of how to determine window sizes.

Window Sizes. Figure 2.13 shows the different window sizes and shapes used in our analysis. The sizes used for our purposes are $window = [2, 4, 8, 16, 32]$, the largest window size used is 32 since this is the largest power of 2 value that evenly divides the 480×640 image that we use. This limitation is due to the square shaped window that we chose to use for our analysis, however, other shapes such as rectangle, circle, or cross would not have this restriction. The square shaped window is used for simplicity. A window size of 2 and 4 are also known as edge detection filters. This is due to the fact that these filters look at the image at a very zoomed in level detecting changes at a pixel level rather than a window level.

Figure 2.13: Square shaped window size filters used to compute the local sharpness of the images (a) window size = 2 (considered an edge detector filter) (b) window size = 4 (sometimes considered an edge detector filter as well) (c) window size = 8 (d) window size = 16 (e) window size = 32

Similar to the global sharpness, we are still measuring the directional change of intensity in the image. The difference between the two is that when computing the local sharpness the overall sharpness is affected by the minor changes, as well as the major changes. This in effect changes the sharpness value of the image. This is because the local sharpness looks at a more zoomed in view of the image and any minor changes in the intensity of the image affect the overall sharpness more than it did when computing the global sharpness. Theoretically, the smaller the window size the higher the sharpness value. After selecting the window size/sizes and shape the sharpness is computed for each of the smaller images, once all these local values are computed they are summed together to obtain a single scalar value that represents the local sharpness of the image.

2.4 Calibration Problem

As mentioned, our goal is to design a sensor that will integrate with the existing infrastructure in aiding the detection of fog and alerting commuters of the weather condition. A vital part of autonomous fog detection is being able to reliably detect fog at any given time. Since we are only using a single-camera image with no reference image, we need to be able to reliably detect fog in an image regardless of the time of day. This means we need to be able to detect fog during the day time as well as night time. Fog detection in day time and night time images is possible, however, they require two separate measurement metrics to determine fog. This is due to the fact that the pixel intensity values in the day image are much brighter than the night images. This in terms of sharpness translates to day time clear image will have a higher sharpness value than the same image in night time, which implies the need for having two separate reference scales in order to determine the density of fog. This also then requires to have some sense of what time of day it is so that the correct reference scale is used in order to determine the density of fog. It is to eliminate this additional requirement we intend on investigating methods to obtain an intensity independent reference scale that can be used for both day images as well as night images. We define this issue as the calibration problem here forth. In Chapter 3 we display the results and discuss our findings for the techniques discussed in this chapter.

CHAPTER 3

STUDY & RESULTS

The preprocessing was performed on the images at which point the global and local sharpness of each image was computed including the global and local sharpness of the original set of images. A 5-scale DWT was then performed on the original and the preprocessed images to obtain 5 sets of images corresponding to the sizes shown in Table 3.1 on page 40. Global and local sharpness is computed again on each of the images. In this chapter we discuss these results and explain the process for computing the sharpness.

3.1 Preprocessing

Using the preprocessing techniques explained in Section 2.1 the four input images are processed to obtain all the images that will be analyzed. The processing applied to the images are negative transformation, logarithmic transformation, power law transformation, and contrast stretching transformation. This is done to analyze images to see which process gives the most desirable results. An important thing to note is that due to the nature of the power law transform we do not apply the power law to either of the night images for any γ -values since we are only using positive values for γ . The use of γ -values greater than 1 means that a narrow range of bright pixel input values will be mapped into a wider range of output values to create a higher contrast, since night images already contain dark pixels this process is skipped. This means that the night images for the power law transform for $\gamma = 2 : 7$ are the same as the original night images. The output of all the preprocessed images is shown below in Figures 3.1 to 3.10. When using the Logarithmic transformation, as well as Power Law transformation the constant, c , is always set to 1 for simplicity but it doesn't have to be 1. Setting c to something other than 1 simply increases the magnitude of the transform however the trend of the sharpness doesn't get affected for linear-values of c . In all the preprocessing functions the input is defined by matrix r , the output is defined by matrix s and both are of size 480×640 . Appendix A contains all the code that is implemented for the different transformation algorithms. The figures below are shown to display the visual effects of

the different preprocessing techniques. Although these help give a good visual understanding of the different techniques, they are not of much use in terms of sharpness. These images can be interpreted differently by different people, some might say the image is blurry while other might say that the images are sharp. To eliminate this opinionated ambiguity of the human vision, a more helpful visual is looking at the image and it's histogram. Figures 3.11 through 3.20 display this information for the day clear image for all the preprocessed techniques.

Figure 3.1: Original Images

Figure 3.2: Negative Transform

Figure 3.3: Logarithmic Transformation

Figure 3.4: Power Law $\gamma = 2$

Figure 3.5: Power Law $\gamma = 3$

Figure 3.6: Power Law $\gamma = 4$

Figure 3.7: Power Law $\gamma = 5$

Figure 3.8: Power Law $\gamma = 6$

Figure 3.9: Power Law $\gamma = 7$

Figure 3.10: Contrast Stretched Images

As mentioned previously histograms are more informative than an image. Figure 3.11 through 3.20 show the day clear image and it's histogram for each of the different preprocessed images. Looking at the histogram is helpful as it provides a distribution of the pixels of an image. Histograms are generally compared to see differences in image contrast. If we compare the histogram of the power law, day clear image where $\gamma = 7$ (Fig. 3.19(b)) with the histogram of the contrast stretched, day clear image (Fig. 3.20(b)) we can see that there is a significant difference in the two histograms. The power law image has a very narrow distribution towards the lower pixel values (these are the darker colors on the gray-scale spectrum), while the contrast stretched image has a much more distributed histogram, this in terms of the histogram means a better contrast image. The computed global sharpness of the two images are 0.0147 and 0.0455, respectively. Looking at the images it would be hard to predict correctly which image is sharper, however given the histogram this becomes a bit simpler to determine, this is backed by the results of the sharpness measure which numerically computes the sharpness of the two images.

Figure 3.11: Original Image and it's histogram

Figure 3.12: Negative Transform Image and it's histogram

(a)

(b)

Figure 3.13: Logarithmic Transform Image and it's histogram

(a)

(b)

Figure 3.14: Power Law Transform Image and it's histogram for $\gamma = 2$

(a)

(b)

Figure 3.15: Power Law Transform Image and it's histogram for $\gamma = 3$

(a)

(b)

Figure 3.16: Power Law Transform Image and it's histogram for $\gamma = 4$

Figure 3.17: Power Law Transform Image and its histogram for $\gamma = 5$

Figure 3.18: Power Law Transform Image and its histogram for $\gamma = 6$

Figure 3.19: Power Law Transform Image and its histogram for $\gamma = 7$

Figure 3.20: Contrast stretched Image and its histogram

3.2 Discrete Wavelet Transform

This function iterates through the preprocessed images (40 total images) one at a time and computes the 5-scale DWT for each image. DWT is computed based on the explanation provided in Section 2.2. Although this process is not vital to the detection of fog, we perform this step to see how sharpness measurements are affected with loss in resolution of the image. If the results are useful it will be important, using lower resolution images means needing less computing power to obtain the same/similar performance. In terms of the sensor design this is a vital step, potentially saving on resources as well as hardware required to perform automatic fog detection. The images used are of size 480×640 which is reduced to half its size with the computation of each scale. Table 3.1 shows the image size reduction as the scale increases, for an input image of 480×640 the 5-scale DWT reduces down to a 15×20 image. A 5-scale DWT is performed for completeness, however past the 3^{rd} scale the image is practically useless since too much spatial information has been lost.

Table 3.1: DWT Image scales along with image sizes for the respective scale.

Stage	Image Size
Input Image	480×640
1	240×320
2	120×160
3	60×80
4	30×40
5	15×20

We start the DWT process by selecting a filter. Our approach uses the **Haar** filter, however, many other filters are available such as the Daubechies filters. We then compute a multi-scale 2-D wavelet decomposition of the image. The decomposition handles boundary extensions by using the symmetric boundary value replication. This type of extension mode mirrors the samples. Although this method has the disadvantage of artificially creating discontinuities of the first derivative at the border, it is commonly used in general for images. We then extract the approximation coefficients at each scale. The function computes the approximation coefficients at the scale N using the wavelet structure along with obtaining the 2-D detail coefficients as explained in Section 2.2. The DWT output plots show the four components that make up the image, the top left image is the approximation or also known as the blur (it is called the blur since there is a loss in resolution) of

the original image, and the other three components are the directional detail matrices of the image (top-right, bottom-left, bottom-right) which are the horizontal, vertical and diagonal directions, respectively. The below image shows the 5-stage DWT of the original day clear image.

Figure 3.21: DWT images for original day clear (a) 1-Stage DWT (b) 2-Stage DWT (c) 3-Stage DWT (d) 4-Stage DWT (e) 5-Stage DWT

Next we show the 5-scale DWT output for each image (Figs. 3.22 to 3.61). These are the 5-scale DWT images of all the images. Image list consists of the original set of images as well as all the preprocessed images. All images are organized similarly to the layout of Figure 3.21.

(a)

(b)

(c)

(d)

(e)

Figure 3.22: DWT for Original Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.23: DWT for Original Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.24: DWT for Original Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.25: DWT for Original Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.26: DWT for Negative transform Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.27: DWT for Negative transform Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.28: DWT for Negative transform Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.29: DWT for Negative transform Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.30: DWT for Logarithmic transform Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.31: DWT for Logarithmic transform Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.32: DWT for Logarithmic transform Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.33: DWT for Logarithmic transform Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.34: DWT for Power Law transform ($\gamma = 2$) Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.35: DWT for Power Law transform ($\gamma = 2$) Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.36: DWT for Power Law transform ($\gamma = 2$) Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.37: DWT for Power Law transform ($\gamma = 2$) Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.38: DWT for Power Law transform ($\gamma = 3$) Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.39: DWT for Power Law transform ($\gamma = 3$) Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.40: DWT for Power Law transform ($\gamma = 3$) Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.41: DWT for Power Law transform ($\gamma = 3$) Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.42: DWT for Power Law transform ($\gamma = 4$) Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.43: DWT for Power Law transform ($\gamma = 4$) Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.44: DWT for Power Law transform ($\gamma = 4$) Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.45: DWT for Power Law transform ($\gamma = 4$) Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.46: DWT for Power Law transform ($\gamma = 5$) Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.47: DWT for Power Law transform ($\gamma = 5$) Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.48: DWT for Power Law transform ($\gamma = 5$) Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.49: DWT for Power Law transform ($\gamma = 5$) Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.50: DWT for Power Law transform ($\gamma = 6$) Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.51: DWT for Power Law transform ($\gamma = 6$) Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.52: DWT for Power Law transform ($\gamma = 6$) Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.53: DWT for Power Law transform ($\gamma = 6$) Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.54: DWT for Power Law transform ($\gamma = 7$) Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.55: DWT for Power Law transform ($\gamma = 7$) Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.56: DWT for Power Law transform ($\gamma = 7$) Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.57: DWT for Power Law transform ($\gamma = 7$) Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.58: DWT for Contrast Stretched Day Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.59: DWT for Contrast Stretched Day Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.60: DWT for Contrast Stretched Night Clear Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

(a)

(b)

(c)

(d)

(e)

Figure 3.61: DWT for Contrast Stretched Night Foggy Image (a) Stage 1 DWT (b) Stage 2 DWT (c) Stage 3 DWT (d) Stage 4 DWT (e) Stage 5 DWT

As it is clear from the images that at each stage of the DWT the image loses resolution, the images are half the size of the previous stage and thus by the 4th – scale the image is of no use since the dimensions on the image are 30×40 . Such a small image has no useable information to determine the existence of fog.

3.3 Sharpness Analysis

Sharpness analysis are done using two different methods. The first method of computing sharpness is using a global sharpness measure where the entire image is analyzed at once. A more detailed sharpness analysis is computed where the local sharpness is measured to determine the sharpness of the image.

This section compares the preprocessed image sharpness alongside the sharpness measures after applying DWT. The plots are different due to the reduction in size of the images which is a direct effect of the DWT. The preprocessed images are all of size $[480 \times 640]$ whereas the output of the scale-1 DWT are of size $[240 \times 320]$. The size is reduced to half the size of the previous scale for each subsequent scale (see Table 3.1 for size of images for each scale). This reduction in size causes a loss in data hence the difference in magnitude of the computed sharpness measurement values. Although there is a loss in the data, the important fact remains that the sharpness trends do not change for the preprocessing techniques. Values have changed but they still follow the general expected trend, with the exception of the logarithmic transform. We are not concerned with the precise computed values, however, we are more interested in comparing the effect that scaling of an image has in terms of the general trends. Also note, for the power law transformations (i.e. Pow2 - Pow7) the night clear and the night foggy images have not been modified from the original images, hence the original, and power law transformations have the same values for the night images. The other transformations are applied on all images hence those sharpness values are different from the original image's sharpness values for the day clear/foggy images and the night clear/foggy images. Notice that the axis limits of the plots in Figures 3.62 to 3.67 are not uniform, this is intentional since we are trying to see the trend based on the scaling factor. We have not made the axis uniform since scaling causes the magnitude of the sharpness measures to drop exponentially and for the lower scale plots the trends would look close to a straight line, hence the zoomed in view is

displayed so that the trend can be seen. Again, our goal is to observe the trends and we are not as concerned with the magnitude of the sharpness values.

3.3.1 Global Sharpness

Global sharpness is done using the gradient function that is available in *MATLAB*. The *MATLAB gradient* function uses the forward difference algorithm at the edges and uses the central difference algorithm for the interior values. The formulas for the two algorithms are shown below in (3.1) and (3.2), respectively. In the equations G is a matrix that represents the gradient image, and f is the input image, h in the equations is the spacing between the pixels, which in our case is $h = 1$.

$$G(x) = f(x) - f(x - h) \quad (3.1)$$

$$G(x) = \frac{f(x + h) - f(x - h)}{2h} \quad (3.2)$$

The code for this implementation can be seen in appendix A.3. Using the *gradient* function in *MATLAB* we obtain the directional intensity changes individually. We then use these values to find the square root of the Sum Squared error (SSE) to get a scalar value that we use as the sharpness measurement value. The plots denoting the sharpness measurements are shown in Figures 3.62 to 3.67. We compute the global sharpness by computing the directional gradient of the entire image at once. For a more in-depth explanation refer to Chapter 2.3. Each of the sharpness figures shows the Global sharpness followed by the local sharpness for each scale starting with the Original set of images all the way down to the scale-5 DWT. Figure 3.62 shows the global and local sharpness for the preprocessed images. Figures 3.62 to 3.67 shows the DWT for each scale 1-5, respectively. The description of what each of the lines represents in all of the plots is described in Table 3.2.

Figure 3.62 shows the global sharpness of the preprocessed images of size 480×640 for the day clear, day foggy, night clear, and night foggy images. The plot shows the sharpness values of the 10 different preprocessing techniques applied to each of the images. Looking at the plot, the original images and the negative transformed images have the same sharpness values for each of the images, this is expected since the negative transformation is a linear transform that only inverts the image intensity but doesn't hinder the image in any other way. The logarithmic transformation sharpness values are consistently below that of the original sharpness values, this trend is seen in the lower resolution images (Figs. 3.62 to 3.67) as well. This indicates that the logarithmic transform may

Table 3.2: Legend information for Figures 3.62 to 3.67

Legend Entry	Description	Line & Marker Type
Orig	Original Images
Neg	Negative Transformation
Log	Logarithmic Transform
Pow2	Power Law Transform $\gamma = 2$
Pow3	Power Law Transform $\gamma = 3$
Pow4	Power Law Transform $\gamma = 4$
Pow5	Power Law Transform $\gamma = 5$
Pow6	Power Law Transform $\gamma = 6$
Pow7	Power Law Transform $\gamma = 7$
Cont	Contrast Stretching Transform

not be the best choice for fog detection. The power law transformation plots ($\gamma = 2...7$) result in a lower sharpness value than the original images for the day images, while having the same sharpness value as the original image sharpness for the night images. The low values for the day images make the power law transformation unattractive for our purposes. Lastly, the contrast stretching transformation produces sharpness values that are consistently higher than the sharpness values of the original images. Along with yielding higher sharpness values, they also follow the trend of the original images which makes them more preferable when computing sharpness of an image.

Computing the 5-scale DWT of the images was performed to determine how much image resolution is needed at a minimum to obtain reasonable results for fog detection. As it can be seen in Figures 3.65 to 3.67 that past a certain point the loss in information causes a change in the sharpness value trend making the respective scale unusable for fog detection. More specifically, past the 2nd scale the loss in information causes unreliable data. In Figure 3.65 the night clear image's sharpness value is higher than the sharpness value of the day foggy images, this is unexpected since calibration has not been applied to the images. It is for this reason that the data in the scale-3 DWT and the remaining subsequent stages has been determined unreliable. We would expect that for any given set of images the day clear image would have the highest sharpness value, followed by the day foggy, then by night clear and finally the night foggy image which would have the lowest sharpness value. Any plot that doesn't meet this expectation is considered unreliable data.

3.3.2 Local Sharpness

Computing the local sharpness is a bit more resource and time intensive than computing the global sharpness since this involves computing the sharpness for a small window size and iteratively navigating through the entire image to obtain the sharpness measure of each of the different windows, and finally summing up all these values together to obtain one scalar value. At the basis this function also uses the same function that is used in the computation of the global sharpness measurement. However, in order to compute the local sharpness we first select a window size and iteratively pass in the sub-image to get a local sharpness value which is then summed together once the entire image has been traversed through. Once we have the sub-image sharpness measurement values we do a 2-D summation to obtain a scalar value. For our purposes we chose the window sizes to be 2, 4, 8, 16, 32 pixels. This process is done to obtain a sharpness measure that is weighted more by the local changes and details rather than the image as a whole. The window sizes were selected based on the input image size of 480×640 , since we decided to use only powers of two that evenly divide our original image's input dimensions we get 32, which is the largest power of two value that evenly divides the original input image dimensions. Similar to the global sharpness we compute the local sharpness of the original set of images as well as the 5-scale DWT images. As the image sizes reduce the window size that we use may not be suitable for the sharpness analysis. At this point the sharpness value of that image is designated as N/A . Since we know the input image size and the window sizes we can determine which DWT scales and window sizes will result in a sharpness value that cannot be computed and hence will be designated as N/A . This information is shown in a tabular form in Table 3.3. In the table *value* implies that the sharpness value of the image for the specified window size can be computed. For example, for an image size of 120×160 and a window size of 8, a value can be computed since $120/8 = 15$ and $160/8 = 20$. This means that local sharpness can be computed for this image since dividing by 8 of both dimension results in an integer value. However, for the same image size and a window size of 16 this is not possible since $120/16 = 7.5$ and $160/16 = 10$. Although 16 divides evenly into 160 it doesn't evenly divide into 120, hence the sharpness value cannot be computed and is assigned to N/A . Local sharpness, just like global sharpness shares similarities in the trends. All the preprocessed images follow the same trends before and after the DWT with the exception of the Logarithmic Transform. Images for the different local transforms are shown below in Figures 3.62, 3.63, 3.64, and 3.66. As discussed

previously, for window size of $W = 32$ sharpness is only computed for the original image sizes, sharpness has not been computed for the DWT scaled images since the window size is too large and does not divide evenly with the dimensions of the images.

Table 3.3: Output values for Sharpness analysis based on image size and window size.

	Original Images (480×640)	Scale-1 DWT Images (240×320)	Scale-2 DWT Images (120×160)	Scale-3 DWT Images (60×80)	Scale-4 DWT Images (30×60)	Scale-5 DWT Images (15×30)
2	value	value	value	value	value	N/A
4	value	value	value	value	N/A	N/A
8	value	value	value	N/A	N/A	N/A
16	value	value	N/A	N/A	N/A	N/A
32	value	N/A	N/A	N/A	N/A	N/A

As shown in Table 3.3 for a window size of $W = 4$ only 4 sharpness sets are computed, the original, scale-1 DWT, scale-2 DWT and scale-3 DWT. The rest of the images are not considered due to the window size not evenly dividing into at least one of the image dimensions. The trends here are similar to the previously discussed global and local sharpness values with a reduction in the magnitude of the sharpness value. This is expected since data is lost with each scale of the DWT. Figures 3.65, 3.66 and 3.67 show similar trends. It is evident from these analysis that using the highest quality image in conjunction with the smallest window size produces the most reliable sharpness evaluation. However, it is also possible to use a scale-1 DWT with a small window size to yield similar results, this reduction in size relates to using fewer hardware resources and thus making it possible for the sensor to be a smaller and more energy efficient device.

In Figure 3.62 we can see that the sharpness value is highest for the local sharpness computation for a $win = 2$, and subsequently reduces as the image size reduces. The reduction ratio in sharpness value is approximately 4. As with the global sharpness, the local sharpness trends also match up, wherein any images past the 2^{nd} – scale DWT produce data that is unreliable and hence useless. The 3^{rd} – scale DWT is arguably passable since the trend is still followed, however, we disregard this plot as well since the borderline characteristic of the Day foggy and Night clear images sharpness value could cause ambiguity in our fog detection decision.

Figure 3.62a

Figure 3.62b: Global & Local Sharpness image comparison for the preprocessed images (Image size 480×640) (a) Global Sharpness (b) Local Sharpness ($win = 2$) (c) Local Sharpness ($win = 4$) (d) Local Sharpness ($win = 8$) (e) Local Sharpness ($win = 16$) (f) Local Sharpness ($win = 32$).
NOTE: Refer to Table 3.2 to understand the legend

Figure 3.62c: Continued

Figure 3.62d: Continued

Figure 3.62e: Continued

Figure 3.62f: Continued

Figure 3.63a

Figure 3.63b: Global & Local Sharpness image comparison for Scale-1 DWT (Image size 240×320)
(a) Global Sharpness (b) Local Sharpness ($win = 2$) (c) Local Sharpness ($win = 4$) (d) Local Sharpness ($win = 8$) (e) Local Sharpness ($win = 16$)

Figure 3.63c: Continued

Figure 3.63d: Continued

Figure 3.63e: Continued

Figure 3.64a: Global & Local Sharpness image comparison for Scale-2 DWT (Image size 120×160)
(a) Global Sharpness (b) Local Sharpness ($win = 2$) (c) Local Sharpness ($win = 4$) (d) Local Sharpness ($win = 8$)

Figure 3.64b: Continued

Figure 3.64c: Continued

Figure 3.64d: Continued

Figure 3.65a: Global & Local Sharpness image comparison for Scale-3 DWT (Image size 60×80)
(a) Global Sharpness (b) Local Sharpness ($win = 2$) (c) Local Sharpness ($win = 4$)

Figure 3.65b: Continued

Figure 3.65c: Continued

Figure 3.66a

Figure 3.66b: Global & Local Sharpness image comparison for Scale-4 DWT (Image size 30×40)
(a) Global Sharpness (b) Local Sharpness ($win = 2$)

Figure 3.67a: Global & Local Sharpness image comparison for Scale-5 DWT (Image size 15×20)
(a) Global Sharpness

CHAPTER 4

CONCLUSION & FUTURE WORK

Preliminary results show that sharpness (global and local) depends on the average image intensity, as was expected. However, within day and night sets, foggy images are noticeably less sharp than their clear counterparts. The results discussed in Chapter 3 imply that applying Contrast Stretching to the image yields the highest sharpness values compared to the other images while keeping the trends consistent to that of the original image, be it day or night image. The application of the DWT suggests that images smaller than $1/8$ the original size are not useful in fog detection since a lot of the pertinent spatial information has been lost.

The suggested approach clearly shows a basis for a near-automated image analysis scheme that accurately detects fog from a single camera. We are currently working on a calibration procedure that will allow for images to be analyzed on the same scale regardless of the image intensity. This will allow for a single fog measurement scale for both day and night images. Since fog can exist at any given time of the day/night it is preferable to design a system that analyzes fog regardless of the ambient lighting, having a generalized scale makes it universal so as to not have different sets of scales based on lighting conditions. As mentioned previously, our goal is to design a sensor that performs fog detection, solving the calibration problem will result in the use of a sensor that requires fewer hardware and computing resources.

This thesis sets a basis for further work that would require implementation of this system in a real-world setting performing in real-time to see how it performs and make changes accordingly. The sensor will be integrated into the existing infrastructure and perform analysis on the camera image to provide a sharpness value which will provide information on the density of the fog based on a predefined measurement scale.

APPENDIX A

MATLAB CODE

A.1 Preprocessing Functions

A.1.1 Negative Transform

```
1 function s = pp_inverse(img)
2 s = imcomplement(img);
```

A.1.2 Logarithmic Transform

```
1 function s = pp_log_transform(r)
2 c = 1;
3 s = c.*log(1 + r);
```

A.1.3 Power Law Transform

```
1 function s = pp_power_law(r, gamma)
2 c = 1;
3
4 if (nargin < 2)
5     gamma = 1;
6     disp('For Power Law gamma = 1');
7 end
8
9 % s is the output image
10 % r is the input image
11 s = c*r.^gamma;
```

A.1.4 Contrast Stretching Transform

```
1 function s = pp_contrast_stretch(img)
2 s = imadjust(img, stretchlim(img), []);
```

A.2 DWT Code

```
1 % 2D image analysis
2 function [cod_a1 ,cod_a2 ,cod_a3 ,cod_a4 ,cod_a5] = compute_DWT(X, titles ,
    type)
3 run_local = 'n';
4 global wname ncolors;
5 if run_local == 'y'
6     path = '/Volumes/Storage_USB/School/Master''s Thesis (Fog Detection
        )/Images/Company_images/';
7     name = [path 'Day_Clear.jpg'];
8     X = imread(name);
9 end
10 if size(X,3) == 3
11     X = mat2gray(rgb2gray(X));
12 else
13     X = mat2gray(X);
14 end
15
16 % We will use filters with symmetric extension at the boundaries.
17 dwtmode('sym')
18
19 % Compute a 5-level decomposition of the image
20 stages = 5;
21 wname = 'Haar'; % Filter type to use
22 ncolors = size(gray(256),1); % Number of colors.
23 [wc,s] = wavedec2(X,stages,wname);
24
25 sz = size(X);
26
27 % Extract the level 1 coefficients.
28 a1 = appcoef2(wc,s,wname,1); h1 = detcoef2('h',wc,s,1);
29 v1 = detcoef2('v',wc,s,1); d1 = detcoef2('d',wc,s,1);
30
31 % Extract the level 2 coefficients.
32 a2 = appcoef2(wc,s,wname,2); h2 = detcoef2('h',wc,s,2);
33 v2 = detcoef2('v',wc,s,2); d2 = detcoef2('d',wc,s,2);
34
35 % Extract the level 3 coefficients.
36 a3 = appcoef2(wc,s,wname,3); h3 = detcoef2('h',wc,s,3);
37 v3 = detcoef2('v',wc,s,3); d3 = detcoef2('d',wc,s,3);
38
39 % Extract the level 4 coefficients.
40 a4 = appcoef2(wc,s,wname,4); h4 = detcoef2('h',wc,s,4);
41 v4 = detcoef2('v',wc,s,4); d4 = detcoef2('d',wc,s,4);
```

```

42
43 % Extract the level 5 coefficients.
44 a5 = appcoef2(wc,s,wname,5); h5 = detcoef2('h',wc,s,5);
45 v5 = detcoef2('v',wc,s,5); d5 = detcoef2('d',wc,s,5);
46
47 cod_a1 = wcodemat(a1,ncolors); cod_a1 = wkeep(cod_a1, sz/2);
48 cod_h1 = wcodemat(h1,ncolors); cod_h1 = wkeep(cod_h1, sz/2);
49 cod_v1 = wcodemat(v1,ncolors); cod_v1 = wkeep(cod_v1, sz/2);
50 cod_d1 = wcodemat(d1,ncolors); cod_d1 = wkeep(cod_d1, sz/2);
51 figure,
52 image([cod_a1,cod_h1;cod_v1,cod_d1]);
53 box off; axis image; set(gca,'XTick',[],'YTick',[]);
54 colormap(gray(256))
55 str = ['./THEESIS_imgs/DWT_L1-', titles, '_-', type];
56 saveas(gca,str,'jpg');
57 close;
58
59 % Display the entire decomposition upto level 2.
60 cod_a2 = wcodemat(a2,ncolors); cod_a2 = wkeep(cod_a2, sz/4);
61 cod_h2 = wcodemat(h2,ncolors); cod_h2 = wkeep(cod_h2, sz/4);
62 cod_v2 = wcodemat(v2,ncolors); cod_v2 = wkeep(cod_v2, sz/4);
63 cod_d2 = wcodemat(d2,ncolors); cod_d2 = wkeep(cod_d2, sz/4);
64 figure,
65 image([[cod_a2,cod_h2;cod_v2,cod_d2],cod_h1;cod_v1,cod_d1]);
66 box off; axis image; set(gca,'XTick',[],'YTick',[]);
67 colormap(gray(256))
68 str = ['./THEESIS_imgs/DWT_L2-', titles, '_-', type];
69 saveas(gca,str,'jpg');
70 close;
71
72 % Display the entire decomposition upto level 3.
73 cod_a3 = wcodemat(a3,ncolors); cod_a3 = wkeep(cod_a3, sz/8);
74 cod_h3 = wcodemat(h3,ncolors); cod_h3 = wkeep(cod_h3, sz/8);
75 cod_v3 = wcodemat(v3,ncolors); cod_v3 = wkeep(cod_v3, sz/8);
76 cod_d3 = wcodemat(d3,ncolors); cod_d3 = wkeep(cod_d3, sz/8);
77 figure,
78 image([[[cod_a3, cod_h3; cod_v3, cod_d3],cod_h2;cod_v2,cod_d2],cod_h1
       ;...
       cod_v1,cod_d1]);
79
80 box off; axis image; set(gca,'XTick',[],'YTick',[]);
81 colormap(gray(256))
82 str = ['./THEESIS_imgs/DWT_L3-', titles, '_-', type];
83 saveas(gca,str,'jpg');
84 close;
85

```

```

86 % Display the entire decomposition upto level 4.
87 cod_a4 = wcodemat(a4,ncolors); cod_a4 = wkeep(cod_a4 , sz/16);
88 cod_h4 = wcodemat(h4,ncolors); cod_h4 = wkeep(cod_h4 , sz/16);
89 cod_v4 = wcodemat(v4,ncolors); cod_v4 = wkeep(cod_v4 , sz/16);
90 cod_d4 = wcodemat(d4,ncolors); cod_d4 = wkeep(cod_d4 , sz/16);
91 figure ,
92 image([[[[cod_a4 , cod_h4; cod_v4 , cod_d4] , cod_h3; cod_v3 , cod_d3] ,...
93         cod_h2;cod_v2 ,cod_d2] ,cod_h1;cod_v1 ,cod_d1] ) ;
94 box off;
95 axis image; set(gca,'XTick',[], 'YTick',[]);
96 colormap(gray(256))
97 str = ['./THEESIS_imgs/DWT_L4-', titles , '- ', type];
98 saveas(gca,str , 'jpg');
99 close;
100
101 % Display the entire decomposition upto level 5.
102 cod_a5 = wcodemat(a5,ncolors); cod_a5 = wkeep(cod_a5 , sz/32);
103 cod_h5 = wcodemat(h5,ncolors); cod_h5 = wkeep(cod_h5 , sz/32);
104 cod_v5 = wcodemat(v5,ncolors); cod_v5 = wkeep(cod_v5 , sz/32);
105 cod_d5 = wcodemat(d5,ncolors); cod_d5 = wkeep(cod_d5 , sz/32);
106 figure ,
107 image([[[[[cod_a5 , cod_h5; cod_v5 , cod_d5] , cod_h4; cod_v4 , cod_d4] ,...
108         cod_h3; cod_v3 , cod_d3] ,cod_h2;cod_v2 ,cod_d2] ,cod_h1;cod_v1 ,cod_d1
109         ] ) ;
110 axis image; set(gca,'XTick',[], 'YTick',[]);
111 colormap(gray(256))
112 str = ['./THEESIS_imgs/DWT_L5-', titles , '- ', type];
113 saveas(gca,str , 'jpg');
114 close;

```

A.3 Sharpness Function

```

1 %% Sharpness Estimation From Image Gradients
2 % Estimate sharpness using the gradient magnitude.
3 % sum of all gradient norms / number of pixels give us the sharpness
4 % metric.
5 function [sharpness]=get_sharpness(G)
6
7 [Gx, Gy]=gradient(double(G));
8
9 S=sqrt(Gx.*Gx+Gy.*Gy);
10
11 sharpness=sum(sum(S))./(480*640);

```

A.3.1 Local Sharpness Function

```
1 function l_sharp = local_sharpness(n, img)
2 % n is the window size to compute sharpness
3
4 rows = size(img,1)/n;
5 cols = size(img,2)/n;
6
7 if n == 32
8     if ~isinteger(rows) || ~isinteger(cols)
9         rows = rows*2;
10        cols = cols*2;
11    end
12 end
13
14 sharp = zeros(rows,cols);
15
16 r_cnt = 1;
17
18 for i = 1:rows
19     c_cnt = 1;
20     for j = 1 : cols
21         a = img(r_cnt:r_cnt-1+n, c_cnt:c_cnt-1+n);
22         sharp(i,j) = get_sharpness(a);
23         c_cnt = c_cnt + n;
24     end
25     r_cnt = r_cnt + n;
26 end
27
28 l_sharp = mean2(sharp);
```

BIBLIOGRAPHY

- [1] Federal Highway Administration. How do weather events impact roads, Nov. 2014. Document available at http://www.ops.fhwa.dot.gov/weather/q1_roadimpact.htm.
- [2] D Aubert, V Boucher, R Brémond, P Charbonnier, A Cord, E Dumont, P Foucher, F Fournela, F Greffier, D Gruyer, N Hautière, V Muzet, P Nicolle, and J Tarel. Digital imaging for assessing and improving highway visibility. 2014.
- [3] S. Bronte, L. M. Bergasa, and P. F. Alcantarilla. Fog detection system based on computer vision techniques. *2009 12th International IEEE Conference on Intelligent Transportation Systems*, pages 1–6, 2009.
- [4] Jurek Grabowski Bruce Hamilton, Brian Tefft, Lindsay Arnold. Hidden Highways : Fog and Traffic Crashes on America’s Roads. (November), 2014. Document found at <https://www.aaafoundation.org/sites/default/files/2014FogAndCrashesReport.pdf>.
- [5] Laurent Caraffa and Jean-philippe Tarel. Daytime Fog Detection and Density Estimation with Entropy Minimization. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences (PCV’14)*, II(1):25–31, 2014.
- [6] Cristian Ordoyo Casado and Antoaneta Popova. IMAGE CONTRAST ENHANCEMENT METHODS. 2010.
- [7] Jorge Caviedes and Franco Oberti. A new sharpness metric based on local kurtosis, edge and energy information. *Signal Processing: Image Communication*, 19(2):147–161, 2004.
- [8] Kanjar De and V. Masilamani. Image sharpness measure for blurred images in frequency domain. *Procedia Engineering*, 64:149–158, 2013.
- [9] Judith Dijk, Michael van Ginkel, RutgerJ. van Asselt, LucasJ. van Vliet, PietW. Verbeek, and Michael Van Ginkel. A new sharpness measure based on Gaussian lines and edges. *Computer Analysis of Images and Patterns*, 2756:149–156, 2003.
- [10] Konstantina Fotiadou, Grigorios Tsagkatakis, and Panagiotis Tsakalides. *Low Light Image Enhancement via Sparse Representations*, pages 84–93. Springer International Publishing, Cham, 2014.
- [11] Rafael Gonzalez and Richard Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

- [12] Nicolas Hautiere, Jean-Philippe Tarel, Jean Lavenant, and Didier Aubert. Automatic fog detection and estimation of visibility distance through use of an onboard camera. *Machine Vision and Applications*, 17(1):8–20, 2006.
- [13] Nicolas Hautiere, Jean-Philippe Tarel, Jean Lavenant, and Didier Aubert. Enhanced fog detection and free-space segmentation for car navigation. *Machine Vision and Applications*, 25(3):667–679, 2014.
- [14] David Doermann Jayant Kumar, Francine Chen. sharpness-estimation-for-document-and-scene-images.pdf.
- [15] Bernard O Keefe. Fog Sensor / ITS Integration. (March), 2003. Document available at <https://cait.rutgers.edu/files/FHWA-NJ-2003-031.pdf>.
- [16] David J. Ketcham. Real-time image enhancement techniques. *Proc. SPIE*, Vol. 0074:120–125, 1976.
- [17] Lark Kwon, Jaehee You, and Alan C Bovik. Referenceless Perceptual Fog Density Prediction Model. *IEEE Trans. Image Proc.*, 9014:1–12, 2014.
- [18] Jung-rim Lee, Chu-yong Chung, and Mi-lim Ou. Automatic Fog Detection Algorithm With Application of Dynamic Thresholds. pages 8–13, 2007.
- [19] X. Marichal, Wei-Ying Ma Wei-Ying Ma, and HongJiang Zhang HongJiang Zhang. Blur determination in the compressed domain using DCT information. *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, 2:386–390 vol.2, 1999.
- [20] Avinash Unnikrishnan Mehrdad Shahabi, Aung Hlaing, David R. Martinelli. Fog Detection for Interstate and State Highways. <http://www.mautc.psu.edu/docs/WVU-2010-01.pdf>, 2012.
- [21] Mihai Negru and Sergiu Nedevschi. Assisting Navigation in Homogenous Fog. *Proceedings of the 9th International Conference on Computer Vision Theory and Applications*, pages 619–626, 2014.
- [22] Njw000. Intensity image with gradient images, 2010. Image available at https://en.wikipedia.org/wiki/Image_gradient#/media/File:Intensity_image_with_gradient_images.png.
- [23] Mario Pavlić, Heidrun Belzner, Gerhard Rigoll, and Slobodan Ilić. Image based fog detection in vehicles. *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 1132–1137, 2012.
- [24] Paul a. Pisano, Lynette C. Goodwin, and Michael a. Rossetti. U.S. highway crashes in adverse road weather conditions. *24th Conference on Institutional Information Processing System*, pages 1–15, 2008.

- [25] Doron Shaked and Ingeborg Tastl. Sharpness measure: Towards automatic image enhancement. *Proceedings - International Conference on Image Processing, ICIP*, 1(September):937–940, 2005.
- [26] Garima Sharma and Andy Thé. Automating Image Registration with MATLAB. *Mathworks.Com*, (Figure 1):1–7, 2013.
- [27] Jean-Luc Starck, Fionn Murtagh, Emmanuel J Candès, and David L Donoho. Gray and color image contrast enhancement by the curvelet transform. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 12(6):706–717, 2003.
- [28] N Treil, S Mallat, and R Bajcsy. Image wavelet decomposition and applications. *International Journal*, (April):1657–1660, 1989.
- [29] C. T. Vu and D. M. Chandler. S3: A spectral and spatial sharpness measure. In *Advances in Multimedia, 2009. MMEDIA '09. First International Conference on*, pages 37–43, July 2009.
- [30] Chong-Yaw Wee and Raveendran Paramesran. Measure of image sharpness using eigenvalues. *Information Sciences*, 177(12):2533–2552, 2007.
- [31] Medha V Wyawahare, Pradeep M Patil, and Hemant K Abhyankar. Image Registration Techniques : An overview. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 2(3):11–28, 2009.
- [32] Akito Yamasaki, Hidenori Takauji, and Shun Kaneko. Denighting: Enhancement of nighttime images for a surveillance camera. *2008 19th International Conference on Pattern Recognition*, pages 1–4, 2008.
- [33] Barbara Zitová and Jan Flusser. Image registration methods: A survey. *Image and Vision Computing*, 21(11):977–1000, 2003.

BIOGRAPHICAL SKETCH

Jigar Patel was born in Mumbai, India on October 29th, 1987. He moved to the United States of America in 2001. He received a Bachelor of Science in Computer & Electrical Engineering from Florida State University in Spring of 2014 and a Masters of Science in Electrical Engineering in Fall 2016. During his graduate studies at Florida State University he focused in the field of Digital Signal Processing more specifically, Image Processing which was a direct result of an internship that he did with Northrop Grumman during the Summer of 2015. Jigar's Masters Thesis research, presented in this thesis, focused on the development of an automatic fog detection system using only preinstalled cameras on the Florida Highways to provide a new approach to fog detection in highway driving to improve driving safety.