

1-1-2015

Visibility And Confidence Estimation Of An Onboard-Camera Image For An Intelligent Vehicle

Minglei Huang
Wayne State University,

Follow this and additional works at: http://digitalcommons.wayne.edu/oa_theses



Part of the [Engineering Commons](#)

Recommended Citation

Huang, Minglei, "Visibility And Confidence Estimation Of An Onboard-Camera Image For An Intelligent Vehicle" (2015). *Wayne State University Theses*. Paper 425.

This Open Access Thesis is brought to you for free and open access by DigitalCommons@WayneState. It has been accepted for inclusion in Wayne State University Theses by an authorized administrator of DigitalCommons@WayneState.

**VISIBILITY AND CONFIDENCE ESTIMATION OF AN
ONBOARD-CAMERA IMAGE FOR AN INTELLIGENT
VEHICLE**

by

MINGLEI HUANG

THESIS

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirement

for the degree of

MASTER OF SCIENCE

2015

MAJOR: ELECTRICAL ENGINEERING

Approved by:

Advisor

Date

ACKNOWLEDGEMENTS

First of all, I would like to acknowledge my advisor, Dr. Abhilash Pandya. You gave me the opportunity to work, study and develop research ability in the lab, which I really think will benefit not only my career but my future life. I really appreciate your encouragement, fund and teaching me the how to think like an engineer. And it is exactly the starting point of this thesis.

I want to say thanks to Ali, who helps me a lot. You gave me a lot advices about how to pick up a topic, how to think on a higher level and how to keep focus on a research area.

I want to thank all the members from my thesis committee. Dr. Xiaoyan Han, and Dr. Le Yi Wang. Thank you so much for your time and help during the way to finish my thesis.

I also want to thank the CARESer. Alireza, Tony, Luke, Brady, Prem, Shahab and all the other members. Thank you so much about your countless help. I really grateful and I am so lucky to work with you.

Finally, I want to thank my family, who support me and encourage me during these two years. Without your support, it is impossible to finish this thesis.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	ii
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
1 INTRODUCTION.....	1
1.1 Motivation.....	2
1.2 Background.....	4
2 BACKGROUND AND SIGNIFICANCE.....	5
2.1 Monocular and Binocular Camera.....	6
2.2 Monocular Camera Assisted by Radar.....	8
2.3 Other Solutions.....	9
3 CREATION OF SIMULATION ENVIRONMENT.....	11
3.1 Introduction of PreScan.....	11
3.2 Environment Modeling.....	13
3.2.1 Flexible Road.....	14
3.2.2 Lane Adapter.....	14
3.2.3 Ramp Segment.....	17
3.3 Vehicles and Sensors Integration.....	17
3.3.1 Trajectory.....	17
3.4 Some limitations of PreScan.....	19
3.5 Sensors.....	21
3.6 Driver in the loop.....	22
3.6.1 Vehicle Mode.....	22
3.6.2 Eye Tribe and Oculus Rift.....	23

4	VISIBILITY ESTIMATION AND CONFIDENCE ANALYSIS.....	25
4.1	Assumptions and Architecture.....	25
4.1.1	Flat-World Assumption.....	26
4.1.2	Koschmieder’s Law.....	27
4.1.3	Detection System Architecture.....	29
4.2	Rough Visibility Estimation.....	30
4.2.1	Methodology.....	30
4.2.2	Detection of inflection point.....	31
4.2.3	Detection of Horizon.....	36
4.2.4	Optimization.....	38
4.3	Accurate Visibility Estimation.....	40
4.3.1	Methodology.....	40
4.3.2	Confidence Analysis.....	42
4.3.3	Optimization.....	44
5	CONCLUSIONS AND DISCUSSIONS.....	47
5.1	Factors that may Impact Estimation Accuracy.....	47
5.1.1	Tracking Distance.....	48
5.1.2	Image Capture Rate and Initial Position of Reference Image.....	51
5.2	Combination of Single Measurements.....	55
5.3	Conclusion and Future Work.....	55
	REFERENCES	57
	ABSTRACT.....	61
	AUTOBIOGRAPHIC STATEMENT.....	63

LIST OF TABLES

Table 4.1: Visibility Estimation Result Comparing with The Ground Truth.....	39
Table 4.2: Variance Ratio of Vehicle Image in the Different Position under the 150 Meters Ground Visible Range.....	43
Table 4.3: Visibility Estimation Result by Using Different Approaches.....	46
Table 5.1: Tracking Distance VS. Visibility Estimation Accuracy.....	50
Table 5.2: Image Acquisition Speed VS. Visibility Estimation Accuracy.....	52
Table 5.3: Position of Reference Image VS. Visibility Estimation Accuracy.....	54

LIST OF FIGURES

Figure 1.1: Part of the functions currently available on the vehicle.....	4
Figure 2.1: Estimated Visibility by finding the Horizon Point and the Inflection Point.....	7
Figure 3.1: Comparison between PreScan and VISSIM.....	12
Figure 3.2: Flexible Road are more skewed with too many definition points.....	15
Figure 3.3: Entrance Lane Road vs. Lane Adapter Road.....	16
Figure 3.4 : Many Options are available in the Drafting Mode	18
Figure 3.5: Vehicle Can Change Trajectories between Defined Paths Freely.....	18
Figure 3.6: Limitations of PreScan.....	20
Figure 3.7: System Appearance.....	22
Figure 3.8: Human-in-the-loop	24
Figure 4.1: Trigonometry of a pin hole camera model and ground.....	26
Figure 4.2: Trigonometry of a titled pinhole camera model and ground.....	27
Figure 4.3: Detection System Architecture.....	30
Figure 4.4: How Image be Deteriorated by the Fog	31
Figure 4.5: Original Image and the Growing Region	34
Figure 4.6: Estimated Position of the Inflection Point	36
Figure 4.7: Original Image and the Lane detected by Hough Line Detection.....	37
Figure 4.8: Comparison between Theoretical Curve and Actual Curve of How the Image Corrupted by the Fog	39
Figure 4.9: Image Confidence Curve	45

Figure 5.1: Reference Image Position VS. Estimated Visibility Result.....	49
Figure 5.2: Image Acquisition Speed VS. Estimated Visibility.....	52
Figure 5.3: Image Confidence Curve.....	53

CHAPTER 1

INTRODUCTION

The development of the automated vehicle and the advanced driver assistance system (ADAS) unobtrusively relieve the driver from a lot of tedious tasks. With the help of ADAS, drivers can pay more attention and have more time to handle higher-level tasks. In order to accomplish this task, many sensors are integrated into the ADAS system, including RADAR, LIDAR, ultrasound and cameras. However, almost every kind of sensor has its own limitations, like the limited azimuth measurement of RADAR, and short detection range under rain or snow of LIDAR [1]. In addition to these limitations, the cost of sensors is another big issue. Based on this consideration, many researchers focus on the single vision-based system. However, the quality of information provided by the vision sensors is still very sensitive under different weather conditions – the same as the human eyes. Thus, it is very important to report the quality of an image captured by an on-board camera under bad weather is.

In our approach, we estimate the confidence of an image based on the visible range which we analyzed from the same image. Actually, the visibility estimated can not only be used to deduce confidence, but is also an important indicator to help a driver drive safely. According to literature, the human eyes estimate 60 % further about the position of vehicles in front under foggy weather than fair weather [2]. Nowadays, most sensors used to detect visual range are quite expensive to operate, install, and calibrate like the scatterometer and the transmissometer [3]. Therefore, many have proposed their own ways to estimate visibility range based on cheap

sensors, like a camera. This research will focus on the camera approach and the need and solutions to better algorithms.

1.1 MOTIVATION

On-road visibility detection is a problem researched and discussed world-wide in the last decade. Some focused on the standstill cameras installed along the road at first [4]. Affected by the inconvenience and the infrastructure cost, many later moved to the on-board camera systems. However, since the detection algorithms based on such systems are always used under many assumptions, they can only be used under specific terrain conditions, like straight flat road segments.

The method proposed here follows the Koschmieder's law, which is primarily used to calculate how visibility is blurred by fog. Then, it estimates the visibility by detecting the horizon, the inflection point and the variance of the image of the preceding vehicle. The algorithm contributes in three ways: first, regardless of the terrain, the method can always maintain a relatively accurate result as long as the vehicle is on a structured road. Second, this method only requires monocular camera, while many systems require both a LIDAR and a camera in order to make an accurate prediction. Third, the confidence of image will be analyzed during this procedure.

This thesis is organized as follows: in Chapter 2, several perspectives and general achievements about autonomous vehicle will be presented. Second, related works will be summarized based on the different kinds of sensors the visual detection systems used and their limitations. Chapter 3 will briefly introduce the platform we used to create the simulation environment and how we built it. Also, some other components that were added into this system will be mentioned. Chapter 4 will describe how we

analyze the images, the general process of the algorithm and the result under a simulation environment. Finally, we conclude the thesis, analyze the experiment, and introduce the future work.

1.2 BACKGROUND

As many OEMs are ambitious to set their goal to release their conditional automated vehicles (Level three) in 2020 including Ford, GM. One biggest problem now is how to judge whether their system is reliable under a specific environment. As what specified by SAE, a conditional automated vehicle shall be able to reminder a driver to take it over whenever system is not confident. As Fig 1.1 shows, almost half of the ADAS functions are based on the camera solely or partially. For example, lane keeping, surround view, traffic sign recognition, and pedestrian detection. That is the reason why the visibility detection is important since image-based algorithms heavily rely on the visibility. Whenever the visibility falls under a threshold, a detector is needed to warn the user to take over the control.

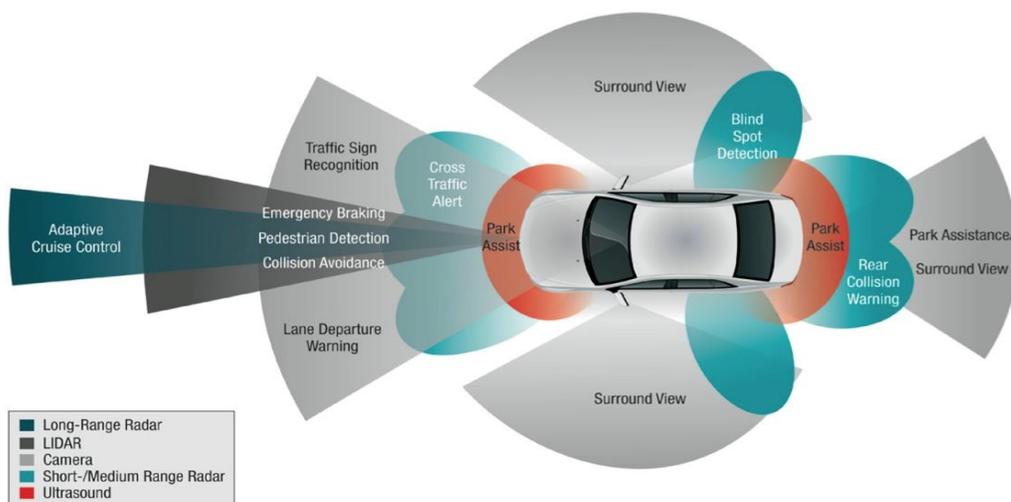


Figure 1.1: Part of the functions currently available on the vehicle [27].

CHAPTER 2

BACKGROUND AND SIGNIFICANCE

In the past 10 years, many researchers already have studied ways to detect fog using a binocular camera, monocular camera or RADAR. Many different solutions have been proposed by researchers to estimate visible range under the fog. First, researchers like Bush et al. [8] and Hasegawa [4] utilized a fixed camera mounted on the road or overhead structure to detect fog. This method exploits some advantages like foreground detection, which can be used to improve accuracy of object contour detection. The general idea of Bush's method is to detect the furthest pixel, whose contrast is greater than 5 % in the contour of any object in the captured image. Later, by estimating the distance of the furthest visible pixel to host-vehicle, visible range is deduced. It was a very popular idea at first. However, since it is a static application, it may involve lot infrastructure problems. Thus, many researchers later shifted their focus to the on-board sensors application. In the following several sections, we introduce several methods to estimate visibility by using different on-board sensors.

2.1 MONOCULAR AND BINOCULAR CAMERA

Most researchers who estimate visible range by monocular camera have to make two assumptions beforehand. The first is the flat world assumption which is used to estimate how far a certain pixel is in the image, from the camera. The other one is Koschmieder's model which is used to estimate the inflection point in the image. Similar solutions and techniques can be found in [5], [9], [10], [14]. The general procedure of these methods is to find inflection points, which represent the division points separating the inside and the outside of the fog, by using region growing algorithm. In addition, the authors need to find out the horizon line in the image. Generally, the Hough line detection algorithm is used to find the horizon. Finally, based on the ordinate of horizon in the image, inflection point, flat-world assumption and Koschmieder's model, visible range in the real world coordinate is deduced. The difference between these solutions mainly involves the kind of region growing algorithms and the kind optimization algorithms that are used to improve the accuracy. Additionally, different methods are utilized to find the inflection point, like the second derivative of the Koschmieder's model, while some others try to find the furthest visible pixel whose contrast is bigger than 5 % in the image to represent it [16].

In order to avoid the flat-world assumption, some try to estimate range with the aid of binocular cameras. Hautiere et al. [16] built a depth map of the vehicle environment and then estimated range based on a v-disparity method. But in general, the main solution is similar.

The biggest problem of such solutions is that they only maintain a relatively

accurate estimation result within 400 meters. In the following picture, the purple point represents the inflection point and the yellow point represents the horizon point. When the actual visibility is too far, the inflection point will be very close to the horizon. Since the image is always somehow deteriorated by noise, the calculated point fluctuates among the real inflection point. This is not a problem when it is very far from the horizon. However, when it is very close to the horizon, a one- or two-pixel distance error in the image will cause a huge error in the final estimated visible range based on Koschmieder's model.

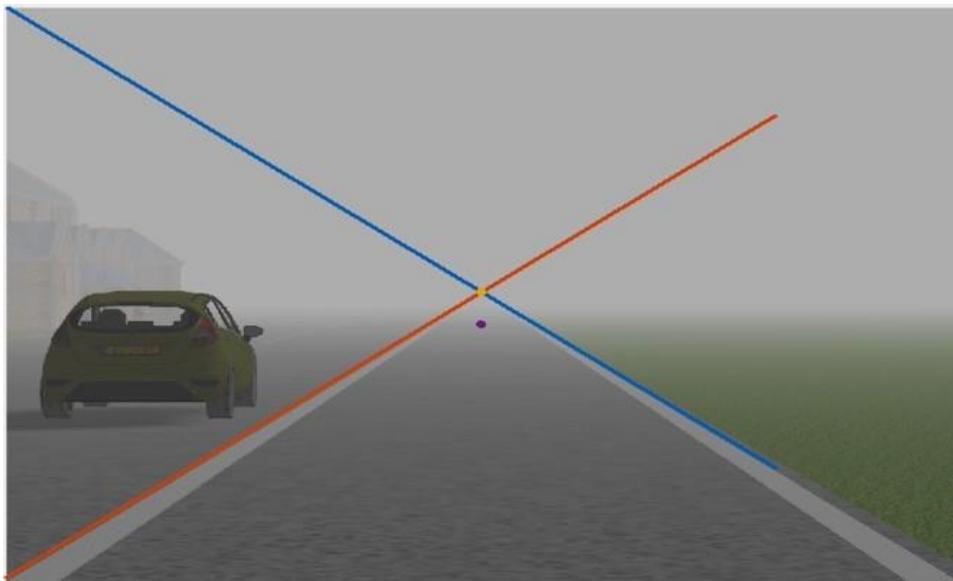


Figure 2.1: Estimated Visibility by finding the Horizon Point and the Inflection Point: the purple point is the Inflection Point and the yellow point is the Horizon Point

Limited by this constraint, many authors use this method to classify weather

conditions rather than the exact determination of the visibility range. The input images are classified into coarse weather categories like dense fog, moderate fog and low fog.

2.2 MONOCULAR CAMERA ASSISTED BY RADAR

Since a monocular camera itself is not enough to make an accurate estimation, many later try to use radar to detect range. Michael Gabb et al. [11] proposed a solution to estimate visibility with aid of a state-of-art vehicle detection and tracking algorithm. The time at which each vehicle disappears in the video and how far they go captured by the on-board camera will be recorded, which represents the visible range. Since there are many vehicles within each frame, there could be many visibility inputs. These inputs are filtered to deduce a final visibility estimation.

However, this algorithm heavily relies on the presence of a number of vehicles to get a correct result and it requires a relatively long time to be initialized. In an extreme case, for instance, when there is no incoming or preceding vehicles in front of the host vehicle, the algorithm cannot work properly. Another problem is the cost. RADAR is much more expensive than a camera. In addition, RADAR usually has a more limited azimuth angle than the camera. The typical detection range for a radar is $\pm 30^\circ$.

Another solution proposed by Kenji et al. [12] [13] is to classify weather categories by evaluating variance or Discrete Cosine Transform (DCT) of vehicle image cropped from images captured on the road. A vehicle template image was used to detect vehicles in the image. RADAR here is also used as a tool to detect a more

accurate range between a front vehicle and the host vehicle. A part of my solution is inspired by their algorithm.

This algorithm also has its own problem. Using a template vehicle image to find vehicles will highly reduce accurate rates of vehicle detection. In addition, to detect a relatively accurate numerical visible distance is more desired than just classify weather conditions into several coarse categories.

2.3 OTHER SOLUTIONS

The issue of finding ways to remove fog effects in the image or restore the image is also encountered frequently in the literature review. Since it is not a main topic of this thesis, however, we will only lightly cover this area.

Most researchers try to recover the image based on an image-deterioration model, which is deduced from Koschmieder's model like Nicolas Hautiere et al. [14] and Narashimhan et al. [15]. What they did was to find the atmosphere coefficient k , assisted by the same method introduced in the previous section. By inserting this coefficient into the Koschmieder's model, an image can be thereby restored. However, due to vertical objects in the image, the intensity of the restored image is discontinuous around the boundary of vertical objects on the road. Thus, by extracting these areas and then trying to saturate the intensity of these areas with surrounding road intensities, a final restored image is presented.

Other solutions which do not rely on the Koschmieder's model are always based on some other findings like Tan et al. [17] and He et al. [18]. He proposed a method to restore an image based on an interesting finding, the black channel. They found that for most haze-free outdoor images in most of the non-sky patches,

there is at least one color channel, the black channel, that has very low intensity at some pixels. More interesting is that 75 % of these low intensity pixels have zero values. Based on this finding, the author took a min operation among all the local path and then deduced the fog transmission equation by calculating the difference between minimum intensity value of local path with value zero. Tan [17] tried to recover the deteriorated image by maximizing the contrast of the image; however, this may bring many artificial particles into the image.

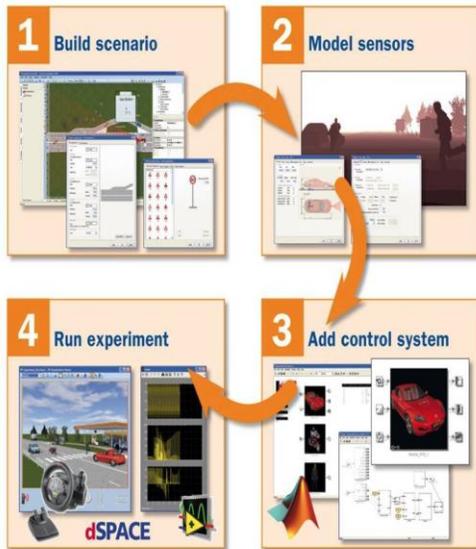
CHAPTER 3

CREATION OF SIMULATION ENVIRONMENT

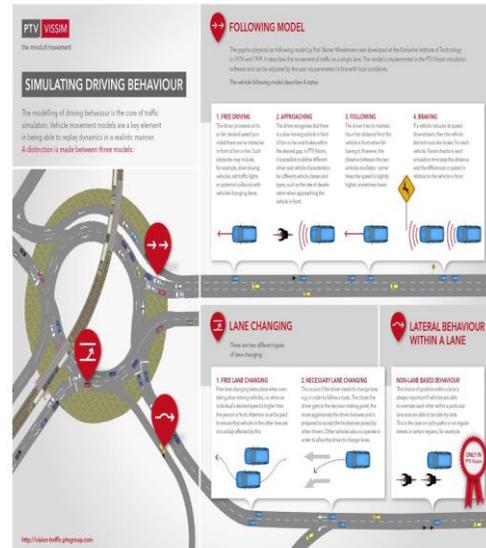
3.1 INTRODUCTION OF PRESCAN

PreScan is a platform used to build simulation environments which can be used to develop and test different kinds of ADAS functions. The biggest advantage of this platform is the availability of multiple types of sensors. PreScan has a big virtual sensor library including camera, LIDAR, RADAR, GPS, and Vehicle-to-Vehicle (V2V) communication protocols. Equipped with these sensors, one can create blocks in the Simulink to develop ADAS functions. In addition, one can add noise and predefine the drift of sensors to simulate possible scenarios, or test the robustness of one's algorithm.

Another simulation platform used ubiquitously is VisSim, which focuses on traffic patterns. Many, therefore, use it to plan the best route, analyze traffic flow or reschedule the timetable of a bus. One advantage of VisSim is that one does not need to set a trajectory for every vehicle. What one needs to do is to define a number for traffic flow and the average speed for different types of vehicles that one wants to



(a) Main Features of PreScan [20]



(b) Main Features of VISSIM [21]

Figure 3.1: Comparison between PreScan and VISSIM

add in the experiment, and VisSim simulates the drivers' behavior automatically. Vehicles in the VisSim will try to avoid collision automatically if possible, while they will only follow predefined trajectories in the PreScan. Thus, the experiments built in the PreScan are all under micro-scenarios with limited numbers of vehicles and distance.

3.2 ENVIRONMENT MODELING

The first task is to create road networks. The information to create a road network is gathered from three parts, a simplified map from OpenStreetMap, a satellite map screenshot from Google Maps, and real world-data collected on the road. The whole process is actually to correct the simplified map manually based on information from satellite map and real-world data.

We can first directly download simplified maps from OpenStreetMap and then underlay a satellite map with it. It is not very easy to align these two maps. In order to align them closely, we need to find a point on the satellite map with its longitude and latitude. That is needed when you import the satellite map. Since every point on the OpenStreetMap is labelled with its GPS data, we can align the maps by overlaying the points. Second, we need to find out the real length of a segment of a recognizable line. It will be used as a scalar to match the two maps. A higher resolution satellite map is recommended. It consumes a lot of time to adjust the alignment if you use a rough map.

We can roughly build the network now, but we still need images and inclination data to estimate heights of rail guards, slopes of the roads, and correct roads.

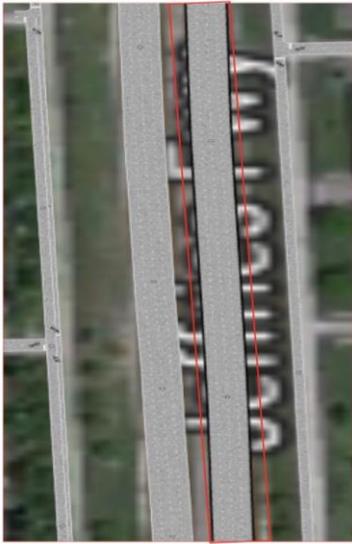
I will next emphasize the segments which I use to build network and their limitations

3.2.1 Flexible Road

The road segment which I used most frequently is flexible road, which is what 90 % of all segments in the road network consist of. PreScan introduces the flexible road as "basically the same as a curved road, but the user has the option to add any number of definition points between start point and end point" [26]. It actually means every definition point you add in the flexible road creates a new segment and thus you can adjust every segment subtly to overlay the satellite map. As we know, roads are not exactly straight in reality. Thus, it offers us this flexibility to create these variations. Besides that, I use the flexible road as long as possible. The longer the road, the less unexpected are the variations that exist, as demonstrated in Fig 3.2. In addition, you can create a sharper curve as the road grows longer. Therefore, it is recommended to connect all flexible roads as one.

3.2.2 Lane Adapter

Lane adapters are used frequently when building freeways. PreScan says, "The Lane Adapter Road is a road segment connecting two roads with a different number of lanes, different widths of lanes or a combination of both." Many may use an entrance lane road or exit lane road to merge roads. However, as I have looked at the satellite map from Google Earth, I found almost all the merge lanes in the freeway merged gradually instead of abruptly. Thus, a lane adapter actually is much more appropriate to use as a road merger, as seen in Fig 3.3 below.

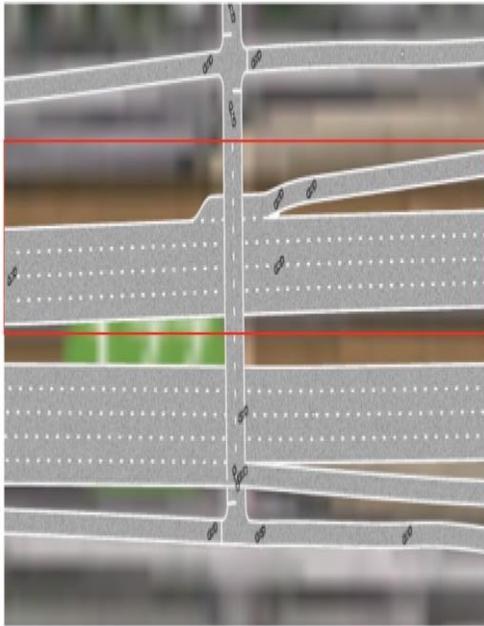


(a) Flexible Road with Limited Number of Definition Points



(b) Flexible Road with Many Definition Points

Figure 3.2: Flexible Roads are more skewed with too many definition points



(a) Built with Entrance Lane Road



(b) Built with Lane Adapter



(c) Road Surface from Satellite Image

Figure 3.3: Entrance Lane Road vs. Lane Adapter Road

Another feature of a lane adapter is that you can enable the Positive / Negative x-dir lane offset option to determine which lane you want to merge. However, only the first lane can be merged if you choose entrance lane road or exit lane road to merge new lanes.

3.2.3 Ramp Segment

A ramp segment is the only road segment that has the option to define a title angle. Unfortunately, it is a straight road. Thus, we use small segments of flexible roads to link ramp segments when building an overpass. Unfortunately, the result of doing so is to make roads look very wavy. However, I still have not found a better solution till now to make it much smoother.

3.3 VEHICLE AND SENSOR INTEGRATION

After building the road network, many other components can be added on. The first one is the vehicle. After that, as mentioned before, each vehicle's trajectory needs to be set in the PreScan.

3.3.1 Trajectory

The trajectory consists of two parts: speed profile and path. You can choose either an inherit mode, or a free drafting mode to plot paths. However, you can only change the elevation of a path in the drafting mode. Thus, you can only use this mode when you have ramp segments in your road.

	Type	Length	Radius at Start	Radius at End	$\Delta \varphi$	Height or Offset	End Pos X	End Pos Y	End Pos Z	End Angle
17 - 18	Polyline	340.00	[-]	[-]	0.30	4.50	3,509.75	-935.96	4.50	4.88
18 - 19	Straight	40.00	[-]	[-]	[-]	[-]	3,549.61	-932.56	4.50	4.88
19 - 20	Arc	[-]	20.01	20.01	2.00	[-]	3,550.30	-932.49	4.50	6.88
20 - 21	Polyline	8.50	[-]	[-]	6.00	0.00	3,558.59	-930.60	4.50	12.88
21 - 22	Polyline	36.00	[-]	[-]	0.00	2.25	3,593.68	-922.57	6.75	12.88
22 - 23	Arc	[-]	90.00	90.00	17.00	[-]	3,618.46	-912.87	6.75	29.88
23 - 24	Polyline	12.00	[-]	[-]	0.00	0.00	3,628.86	-906.90	6.75	29.88
24 - 25	Polyline	30.00	[-]	[-]	10.00	2.25	3,651.88	-887.66	9.00	39.88

Figure 3.4: Drafting Mode Provides More Tunable Parameters

Additionally, drifting mode offers you different types of path segment like arc, Bezier curve, lane change and so on. It is much more useful when used to create different traffic scenarios. In the demo experiments, PreScan also provides a case where a vehicle can change from one path to the other. Fig 3.5 shows how a black vehicle changes its predefined path to a new one.

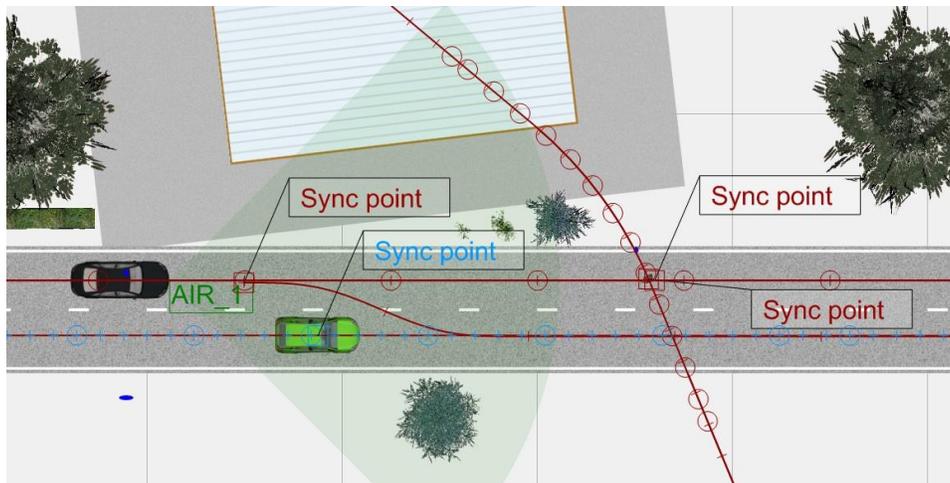


Figure 3.5: Vehicle Can Change Trajectories between Defined Paths Freely

In the speed profile, you can add as many time slots, which can be used to describe the car behavior within the amount of time associated with the slot, as you want. For every slot, we can define a type from wait, sudden throttle release, to acceleration. One can also set the initial position of a vehicle in case one wants to share a path with more than one vehicle. However, please save your file first before you modify your speed profile. It is very likely that your file will crash during modification.

3.4 SOME LIMITATIONS OF PRESCAN

PreScan still has some limitations for now, regarding the road networks. You may notice some work cannot be done easily with the current implementation. Thus, the section here is to remind a new user that some parts of the road cannot be built easily without workarounds.

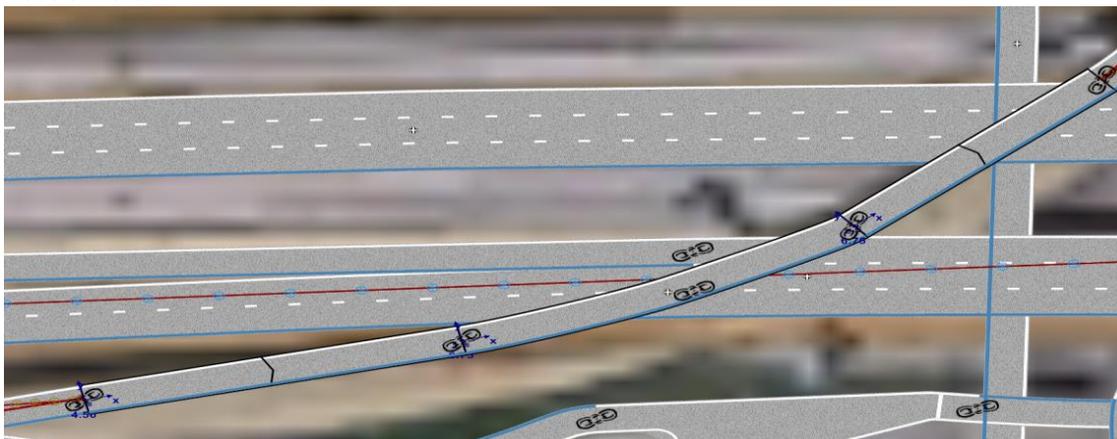
1. PreScan does offer the option to set the width of the shoulder. However, in case the width of right or left shoulder of a road is different, you cannot set width for the shoulder individually.
2. Another case is Fig.3.6 (a). When two lanes of the same road are merged to different lanes simultaneously, Lane Adapter does not offer you the option to imitate this scenario.
3. Although we can use a ramp road to create overpass, we cannot create road underneath the ground. There are also some special patterns which I have no idea to simulate it in the PreScan, as can be seen below.



(a) User cannot set widths for the road individually when the size of shoulders are different



(b) Lanes can not be merged into an arbitrary Lane in the successor road



(c) The wavy overpass

Figure 3.6: Limitations of PreScan

3.5 SENSORS

There are three types of sensors in the PreScan: Idealized Sensor, Detailed Sensor and Ground Truth Sensor. All the information of sensors can be found in the manual. Thus, I try to focus on several important, but may easily missed points.

Firstly, it will consume a lot of time to output a camera image in the Simulink. Thus, if you do not need camera images for real-time processing, it is better to disable the “Output Simulink Image” option in the Camera Sensor configuration. However, in case you need the image for further real-time processing, you have to check this option. Even when you do not need to plot this image, you still need to check it. What the “Output Simulink Image” does is nothing to do with plotting but produces an RGB matrix for the user. Again, it is not recommended to plot image synchronously in the simulation.

Secondly, try not to set the update frequency of the sensor too high, especially when you need to visualize data like LIDAR or RADAR output in real-time. Furthermore, the compilation sheet rate must be higher than the update frequency of sensors. The higher the sheet rate is, the slower the performance of the outputs is. This is because computers need to process more frames when rate is higher.

3.6 DRIVER IN THE LOOP

In order to analyze how realistic the environment we built is, we added the human-in-the-loop feature into the system. A brief explanation of how it was built follows.



Figure 3.7: System Appearance

3.6.1 Vehicle Mode

There are three options that can be chosen from the driver model: Man-in-the-loop, Path Follower, and Path Follower with Preview. A steering wheel is required

to enable Man-in-the-loop mode. The one we use in the experiment is Logitech G27. Once the model is chosen and created in the PreScan, you can have several built-in blocks which are created automatically by PreScan in the SimuLink. From these basic built-in blocks, we can enable manual shift or automatic shift, light the fog light or activate haptic feedback. Also, we can add ADAS function blocks based on sensors to take over the control when emergency is detected.

3.6.2 Eye Tribe and Oculus Rift

Since Man-in-the-loop function is already provided by PreScan, it is easy to reproduce it by following instructions. The problem is how we can integrate Oculus Rift, or Eye Tribe into this system. The first thought is to add a camera view in the driver seat, and then change its direction and view according to the movement of the eyeball or head. Unfortunately, the position and direction of camera is fixed once we predefined them in the PreScan GUI. Only the direction of the vehicle is changeable. An alternative idea is to add another vehicle beneath the ground and then add a camera on it with the same height of the driver's viewer in the host vehicle. We keep the second vehicle in the same position of the host vehicle, but change its direction or view according to the movement of the eyeball or head detected by Eye Tribe or Oculus Rift. In this way, we can simulate how view changes when a driver looks around. Oculus Rift here is used as a head-mounted display and tracking tool – the same as Eye Tribe. The degree of movement of head is detected by these sensors, and sent to SimuLink.

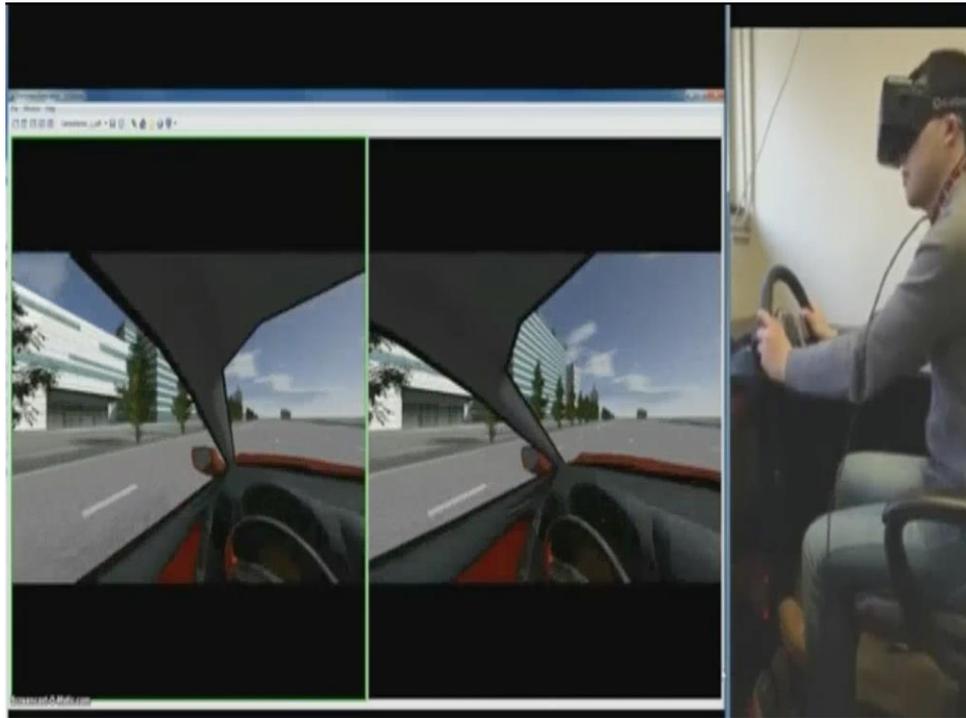


Figure 3.8: Human-in-the-loop

CHAPTER 4

VISIBILITY ESTIMATION AND CONFIDENCE ANALYSIS

In this chapter, I will go through the procedure of the fog detection algorithm. First, several underlying assumptions will be introduced and then the detailed explanation of the algorithm. The algorithm states two parts and then these two ways of measurements will be combined based on a probabilistic model.

4.1 ASSUMPTIONS AND ARCHITECTURE

The first problem of single camera system is distance. In order to approximate the real distance of each row in the image, we need to analyze the image under the flat-world assumption first. Thus, we constrain our method to a structured road which we assume is planar. Secondly, the model used to simulate fog is Koschmieder's Law [6], which is a very popular model used to imitate how fog attenuates the luminance.

4.1.1 Flat-World Assumption

Figure 1.1 below shows us an on-board camera model which we use to calculate distance of each row in the image. Coordinate $S(X,Y,Z)$ is centered at one point on the road, while $C(x,y,z)$ is another coordinate centered at the pinhole of this camera. For an image captured by this camera, we create the third and planar coordinate (u,v) with its center (u_0,v_0) . Assuming f is focal length of the camera, t_{pu} and t_{pv} stand for size of a pixel in millimeter horizontally and vertically (we assume $t_p \cong t_{pu} \cong t_{pv}$), then we say $\alpha = f/t_p$ denotes the focal length of camera represented by the number of pixels. Thus, any points (x,y,z) in the coordinate C can be expressed by the Equation (4.1) below derived from trigonometry of the camera model on the image coordinate.

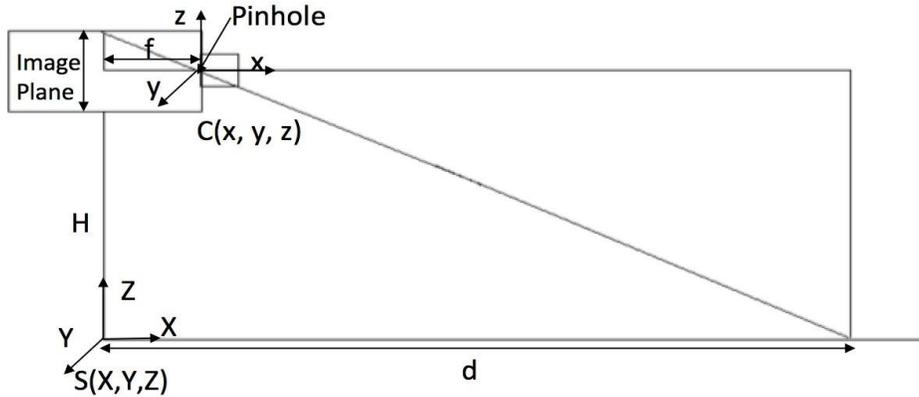


Figure 4.1: Trigonometry of a pinhole camera model

$$\begin{cases} u = u_0 + \alpha^y/x \\ v = v_0 + \alpha^z/x \end{cases} \quad (4.1)$$

As Figure 4.1 shows, we can find two similar triangles. H here in the figure represents the height of the camera based on the coordinate S , if we denote v_h as ordinate of horizon that pass through the image. For any point $M(d,y,0)$ on the ground relative to the $S(X, Y, Z)$ with what it shows on the image plane $m(u,v)$, we have the equation below.

$$\frac{v-v_h}{\alpha} = \frac{H}{d} \quad (4.2)$$

When camera is tilted by degree θ towards the horizon, we can still find a similar equation, in the Figure 4.2. ($V_h C = \frac{f}{\alpha}$)

$$\frac{\frac{v-v_h}{\cos \theta}}{\alpha} = \frac{H}{d} \quad (4.3)$$

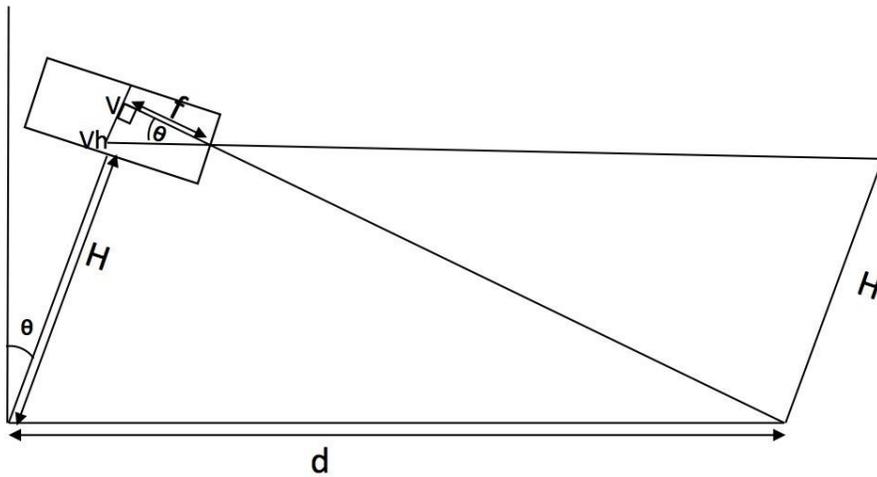


Figure 4.2: Trigonometry of a titled pinhole camera model and ground

4.1.2 Koschmieder's Law

One of the most important and popular models used to study fog effect is Koschmieder's model. Koschmieder [6] studied and revealed his model of how luminance attenuated through atmosphere in 1924. In this equation, he successfully linked the degradation of luminance with the distance of an object.

$$L = L_0e^{-kd} + L_f(1 - e^{-kd}) \quad (4.4)$$

What this model shows us is luminance of an object will be attenuated by coefficient e^{-kd} , and gradually deteriorated by luminance of sky at rate of $L_f(1 - e^{-kd})$. In this equation, L stands for the perceived luminance of an object at distance d , L_0 is the intrinsic luminance of the object, L_f is the background luminance, k is the extinction coefficient. Later, this equation, rewritten by Duntley [7], unveiled the relationship between an object with contrast C_0 against background and observed the contrast C at distance d .

$$C = [(L_0 - L_f)/L_f]e^{-kd} = C_0e^{-kd} \quad (4.5)$$

In 1987, the International Commission on Illumination set a contrast threshold [19], which is 5 %, for the barely visible object, in order to define "meteorological visibility distance" V_{met} . Thus, for a black object, having contrast $C_0=1$, the greatest distance it can be seen is defined as V_{met} . According to (1.4), V_{met} is derived below.

$$V_{met} = -\frac{1}{K} \log(0.05) \cong \frac{3}{K} \quad (4.6)$$

4.1.3 Detection System Architecture

The process of visibility and image confidence estimation, as Figure 1.3 demonstrates, can be separated in two.

In the first part, Sobel and Canny filters are applied to the image in order to find the lane markings. Later, based on the lane markings gotten, we can easily calculate the horizon by employing the Hough line detector. In the meantime, a region growing algorithm is used to find the inflection point. According to Equation (4.3), we can find the distance of inflection point, relative to camera, from ordinates of inflection point and horizon. Then, a rough extinction coefficient k can be deduced. Finally, least square optimization is utilized to get a more accurate coefficient k . This coefficient will be used as base estimation for part two of the algorithm.

What this first step does is to give us general information of the image. In part two, we need to verify both the coefficient k and confidence information we deduced from part one. By tracking vehicles, clipping each vehicle and calculating its variance, we can get another extinction coefficient k_v . By comparing both k_v and k , we can confirm how the result concluded trustworthy in part one. Thus, a more accurate evaluation of an image can be made and image confidence can be deduced.

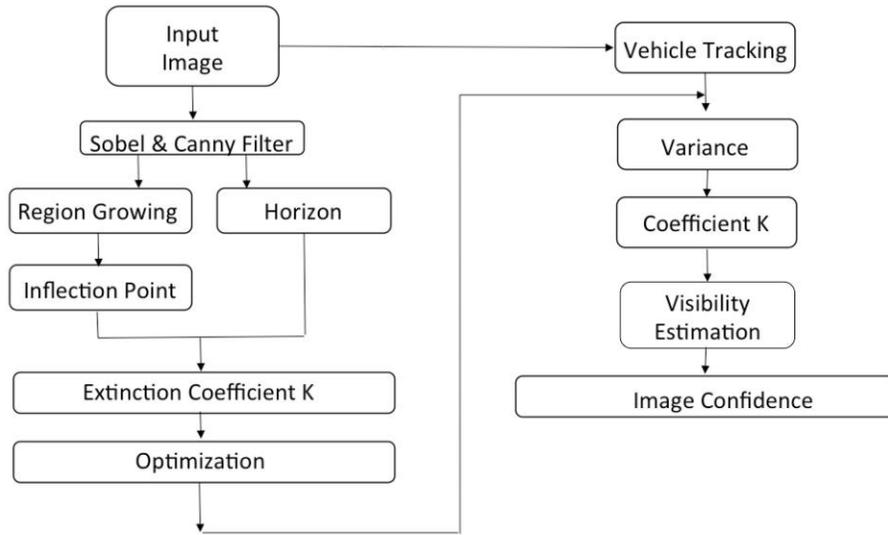


Figure 4.3: Detection System Architecture

4.2 ROUGH VISIBILITY ESTIMATION

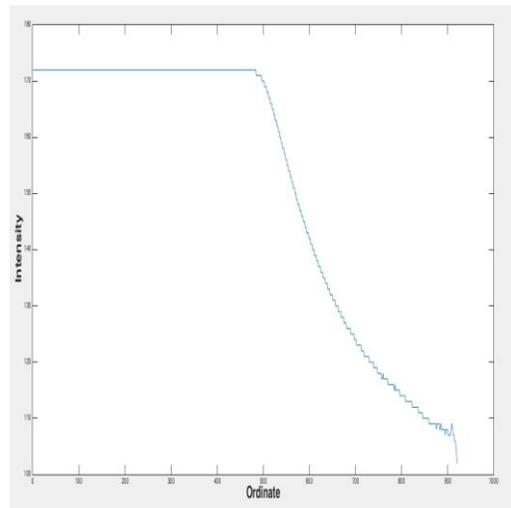
4.2.1 Methodology

To any fog deteriorated image, we can randomly pick a vertical line of image from the top to the bottom. As seen in Figure 4.1 (b), at the position around ordinate 550, the intensity drops dramatically. Intuitively, an object tends to be obscured quickest by fog around the furthest visible range. However, a much more explicit explanation needs a mathematical point of view, which I will explain below.

Since we have already deduced the Equation (4.3) and (4.4), we can merge them together to express Koschmieder's model in a new equation,



(a) Original Image



(b) Intensity of image increases very quickly when it approaches to the furthest visible range

Figure 4.4: How an image can be deteriorated by fog

$$L = L_0 e^{-\frac{\kappa\lambda}{v-v_h}} + L_f \left(1 - e^{-\frac{\kappa\lambda}{v-v_h}} \right) \quad (4.7)$$

Where the $\lambda = \alpha H / \cos \theta$ in the Equation (4.7). As mentioned before, an object tends to be obscured quickest by fog around the furthest visible range. Therefore, we can find the inflection point when the second derivative of Equation (4.7) with respect to image ordinate v equals to zero, which is where the intensity drops quickest in the image.

$$\frac{\partial^2 L}{\partial v^2} = \frac{\kappa\lambda(R-L_f)}{(v-v_h)^3} e^{-\frac{\kappa\lambda}{v-v_h}} \left[\frac{\kappa\lambda}{v-v_h} - 2 \right] = 0 \quad (4.8)$$

If we take the right part of equation (4.8) equal to zero, then we can have two solutions; $k_1 = 0$ and

$$k_2 = \frac{2(v_i - v_h)}{\lambda} = \frac{2}{d_i} \quad (4.9)$$

Where v_i represents inflection point in the image and d_i represents its distance to the camera.

Based on the Equation (4.8) and Equation (4.6), we can find the relation between visibility with inflection point and horizon.

$$d_{met} = 3\lambda / 2(v_i - v_h) \quad (4.10)$$

Thus, the problem now becomes how to find out the inflection point and the horizon.

4.2.2 Detection of inflection point

Region growing algorithms are very popularly used to find out inflection points. In this way, only free space is added into checking the region. Thus, we can avoid the influence of volatile pixels of any object on the road. It will be much more accurate for us to find where and how stable pixels of road surface are deteriorated by the fog. For the sake of efficiency and purpose of a rough result, we choose a similar region growing algorithm listed in [5] with fewer constraints. Generally, the bottom row of the image is always a part of the road surface. Thus, we choose the bottom row as seed pixels for the algorithm. The algorithm is designed to grow upward, since the road surface extends vertically to the top in the image. With these prerequisites, several constraints are added.

For any seed pixel, the algorithm will check the right, left and middle pixels above it. These points will be aggregated into a growing region only if they meet these constraints:

1. $|p_{seed} - p_{new}| \leq 3$

The difference of intensity between seed pixel and new pixel has to be within 3

2. $|p_{medianbottom} - p_{new}| \leq 3n$

$p_{medianbottom}$ represents the median intensity value of pixels in the bottom row, n represents difference of row number between bottom row and p_{new} . This constraint is to keep new aggregated pixels similar with the bottom row.

3. $p \notin R$

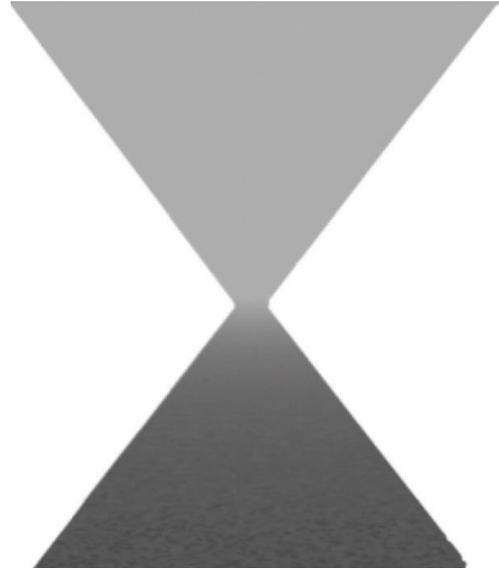
The pixel should not be in the edge of an image.

4. $|p_{median} - p_{new}| \leq 10$

All pixels that meet the requirements above will be reordered to calculate their median intensity value. Any pixel too far from the median will be discarded.



(a) Original Image



(b) Result of the Growing Region

Figure 4.5: Original Image and the Growing Region

Once we get the growing region, we need to check whether it contains both sky and ground surface – which is used as a criteria to check whether fog exists in the image, which is the case in Figure 4.5 above.

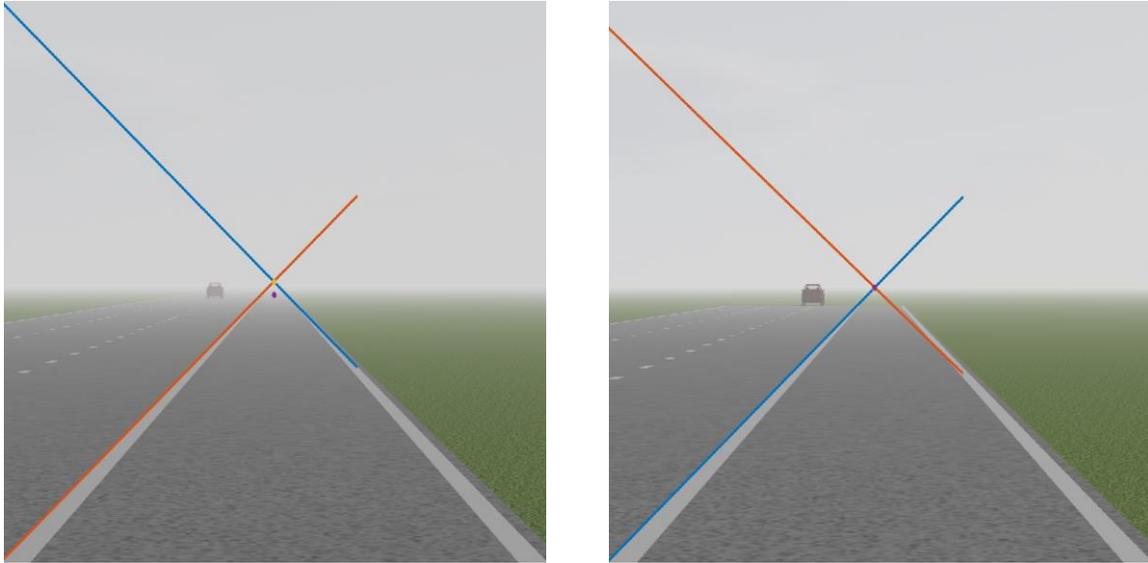
If this is so, we will calculate the median for each row in the region. The result of Fig 4.4(a) is shown in Fig 4.4(b). In order to find the global maxima in the region, we will calculate the median difference for every 20 rows. The row, which

has biggest difference, will be recorded as an inflection row and then we choose the center point in this row as the inflection point. We have tried to calculate median difference every 10 rows or every 15 rows, however the noise introduced is still too large to get a reasonable result. It is actually another reason why we cannot get a rough result from this solution and why the estimation of visibility beyond 400 or 500 meters is inaccurate. It is because we calculate the median difference for every 20 rows. An example is shown here. According to Equation (4.10), if we install a camera which is parallel with ground, with focal length 1.4mm, $t_p = 400$ pixels/mm and 1 meter away from surface. Then λ should be equal to 5600. However the ground truth visibility is 500 meters, then we can get:

$$(v_i - v_h) = \frac{3\lambda}{2d_{met}} = \frac{3 \times 5600}{2 \times 500} = 16.8 \quad (4.11)$$

The equation above tells us that when the visibility range is 500 meters, the inflection point should be 16.8 pixel units away from the horizon point.

As we already have around 10 pixels inaccuracy in calculating the inflection point, you can imagine the result will be very inaccurate when the visibility is beyond 500 meters. Below is an example showing the inflection point estimated from this algorithm when ground truth visibility is 400 or 500 meters. The inflection point (purple) is aligned with horizon point (orange) in Fig 4.6(b) below.

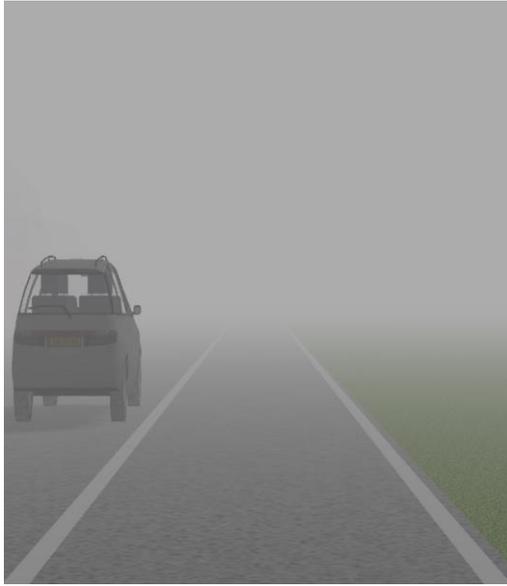


- a) Inflection Point (Purple) Estimated when ground truth visibility is 400 meters
- b) Inflection Point Estimated when ground truth visibility is 500 meters, the inflection point is so close to the horizon point so as to almost the same one

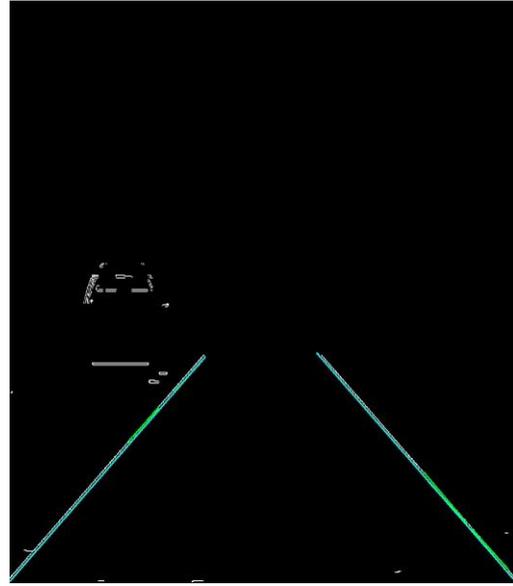
Figure 4.6: Estimated Position of the Inflection Point

4.2.3 Detection of Horizon

The way we use to estimate the horizon is based on the Hough line detection algorithm. The very first step is to find edges in the image by employing the Canny and Sobel detectors. Secondly, the two longest lines, which are actually lane markings, detected by Hough line detection algorithm are extracted after the first step (see Figure 4.7).



(a) Original Image



(b) Result of Hough Line Detection

Figure 4.7: Original Image and the Lane detected by Hough Line Detection

In order to improve the efficiency of the algorithm, Hough Line detection algorithm is modified to fit the scenario. To detect two lane markings, Hough line detector will search for the longest lines in the range from 0 to $\pi/2$ and from $\pi/2$ to π individually. Since there is a second way to calculate the position of the horizon based on the pitch of camera, we can always use it as a way to check consistency between these two methods and also a threshold is set for disagreement of these two methods. Thus, we can reduce the possibility of mistakes in finding the horizon.

4.2.4 Optimization

Based on the position of horizon and inflection point, a visibility range and atmosphere extinction coefficient k_0 can be deduced. However, since some errors are introduced either by the methods or by the device, an optimization is needed for a more accurate result. According to Equation (4.4), if we know the original intensity, fog intensity and coefficient k , then we can approximate the intensity of an object perceived by CCD at different location. Again, we need the growing region calculated in the first step. If we take the median intensity of the bottom row of the region as the original intensity of the ground surface and the median intensity of the top row as intensity of fog, then for any k and distance d , we can get an intensity L . For the efficiency, k will be assigned any value from $k_0-0.01$ to $k_0+0.01$ and square error will be calculated between median intensity from the growing region and theoretical intensity from Equation (4.4). Finally, minimum square error will be found and the calculated new coefficient $k_{optimized}$ will be used to deduce visibility. Figure 4.8 shows a theoretical curve (blue) calculated based on Equation (4.4) and an actual curve calculated based on the growing region (orange).

As you can see from the Figure 4.8, only a part of these two curves aligned together which is from ordinate of the horizon to 50 pixels below the inflection point. Part of that may be caused by the big variance in intensity of the ground surface in the bottom row. In order to avoid such noise, only the rows range from the horizon row to the row 50 pixels below the inflection point will be counted into the optimization algorithm. A table below shows how the optimization algorithm can improve the accuracy.

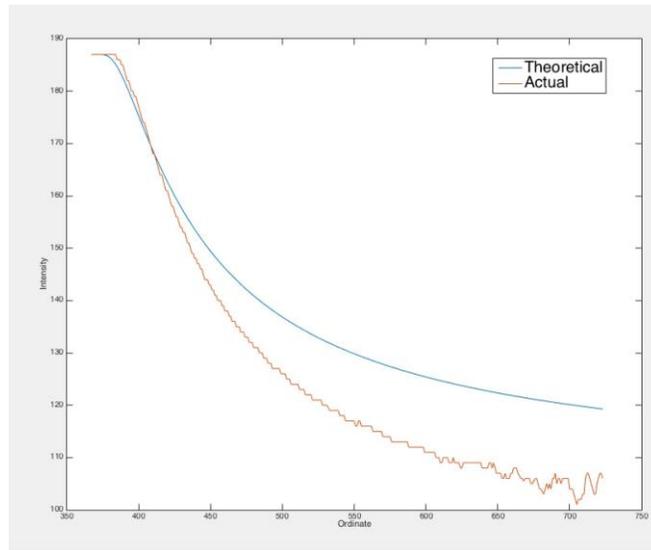


Figure 4.8: Comparison between Theoretical Curve and Actual Curve of How the Image Corrupted by the Fog

Ground Truth	Estimation without Optimization	Estimation with Optimization
50m	46m	59m
100m	88m	103m
200m	140m	222m
350m	193m	279m
400m	255m	288m
500m	Infinity	Infinity

Table 4.1: Visibility Estimation Result Comparing with the Ground Truth

4.3 ACCURATE VISIBILITY ESTIMATION

In this section, I will demonstrate how a new method is introduced to estimate visibility. Later, in Chapter 5 I will show how this method and the one introduced in the last section can work together and the necessity of using this method.

4.3.1 Methodology

The equation listed below shows how we calculate the variance of an image.

$$\sigma^2 = \frac{1}{N} \sum_{j=1}^N (L_j - \frac{1}{N} \sum_{j=1}^N L_i) \quad (4.12)$$

L_i and L_j here represent pixels captured by the CCD of the camera. Based on the Equation (4.4) and (4.12), we can express the variance σ of image by original intensity value of an object which is shown by Equation (4.13). Notice that the L_{j0} here is the original intensity of the object not the intensity captured by the camera. Therefore, σ in the equation (4.12) and σ_i in the (4.13) are the same.

$$\sigma_1^2 = \frac{e^{-2kd_1}}{N} \sum_{j=1}^N (L_{j0} - \frac{1}{N} \sum_{j=1}^N L_{i0}) \quad (4.13)$$

If the image we discussed above is an image of a vehicle, and we have another image of the same vehicle under the same weather condition at a different distance when captured, then we can express the variance of this vehicle image by using same equation above.

$$\sigma_2^2 = \frac{e^{-2kd_1}}{N} \sum_{j=1}^N (L_{j0} - \frac{1}{N} \sum_{j=1}^N L_{i0}) \quad (4.14)$$

In this way, we can express the atmosphere coefficient k as equation (4.15) by (4.13) and (4.14)

$$k = \frac{1}{2(d_2 - d_1)} \ln \left(\frac{\sigma_1^2}{\sigma_2^2} \right) \quad (4.15)$$

So, σ_1 and σ_2 above are the variances of two images and d_1 and d_2 are the distance in the real world between the vehicle captured and the host vehicle. Once we get the coefficient k , based on the Equation (4.9), we can calculate the visibility range.

One step of this method involves how to detect and trim vehicle image from the background. It is actually another topic, which we will only lightly cover here. Many methods and review articles are published in this area including knowledge based methods, stereo-vision-based methods, and motion-based methods. A detailed introduction can be found in [23] and [24]. Actually, since we can detect a vehicle with stereo-vision methods, this is another reason why many researchers will choose stereo-camera to estimate visibility. All the methods listed above are called Hypothesis Generation (HG) methods which is used to find out possible locations of vehicles. After that, Hypothesis Verification (HV) is employed to verify all these possible locations based on machine learning or other techniques. In order to improve the efficiency and precision rate of the detection algorithm, many now consider to integrate the tracking algorithms into it. Since a valid detection result will trigger the process of the tracking and then possible location of the vehicle; the next image frame will feed into the detection algorithm. Thus, a Bayesian approach is

built for detection. However, for the sake of simplicity, we skip all these approaches in this work and just manually select and clip all these vehicle figures.

4.3.2 Confidence Analysis

As stated before, the confidence of the image will also be produced of this method. Confidence itself is not just the only indicator, we are interested about, it is also an intermediate value we need for optimization.

The confidence is based on the variance we calculated in the last step. Consider a case in which we followed a vehicle for 20 meters. The preceding vehicle is 5 m/s faster than the host-vehicle and the on-board camera will capture an image every 1 second. If the preceding vehicle captured is 25 meters away from the host-car in the first image, then we have a series images of this vehicle ranging from 25 to 45 meters. Based on the Equation (4.9), we can get an estimation of the atmosphere coefficient k by any two variances of the vehicle images with different distances between it and the host-car when captured. If we only calculate the ratio of each two variances by Equation (4.16) and for all these five images ranging from 25 meters to 45 meters, then we can actually make a table showing variance ratios (see Table 4.2). The variance ratios calculated below is under the 150 meters visibility of the fog.

$$Ratio = 100 \frac{\sigma_1^2}{\sigma_2^2} \quad (4.16)$$

Ratio	25m	30m	35m	40m	45m
0m	100	100	100	100	100
5m		84.9044	92.1565	77.1763	92.0751
10m			78.2449	71.1230	71.0602
15m				60.3865	65.4866
20m					55.6010

Table 4.2: Variance Ratio of Vehicle Image in the Different Position under the 150 Meters Ground Visible Range

We can just take the last column of Table 4.2 as an example to interpret this table. For the variance σ_{45} of vehicle image we captured at 45 meters away from the host- vehicle, we will definitely get 100 if we replace both the σ_1 and σ_2 from Equation 4.10 by σ_{45} . The ratio, 92.0751, in the next row is based on the ratio between σ_{45} and σ_{40} , which is the variance of vehicle image we captured at 40 meters away from the host-vehicle. There is five meters difference among each image we captured on the road. As you can see, for five images, we can get 15 ratios. Thus, some kind optimization is important in order to increase accuracy and reduce redundancy for these numbers. This kind optimization will be explained in the next section. After such optimization, we can draw a confidence curve to visualize how visibility is attenuated by the fog under different kind thickness of fog.

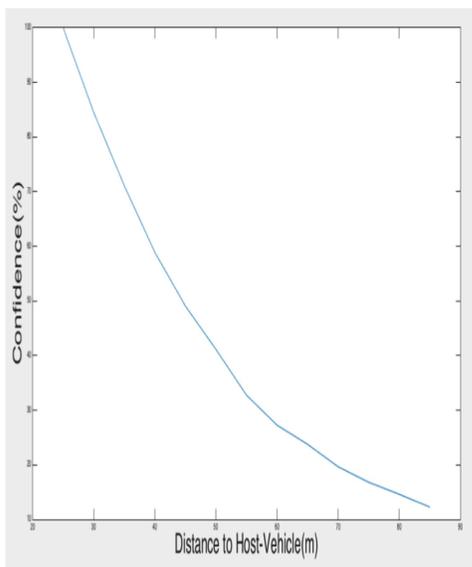
4.3.3 Optimization

Visibility estimated directly from such method can achieve a better result than what we get from the rough visibility estimation method. However, in order to eliminate the effects of noise further, a similar approach employed in the rough visibility method is utilized. As mentioned before, the confidence ratio will be used as an intermediate value for optimize. Actually, after all the ratio numbers are calculated as what shows in the Table 4.2, we will calculate the median of each row in the table. Any ratio value in the table beyond the $\pm 50\%$ of the median will be discarded. Finally, the remaining number will be averaged as the final ratio for calculating visibility. According to Equation 4.13 and 4.14, we can derive the equation below.

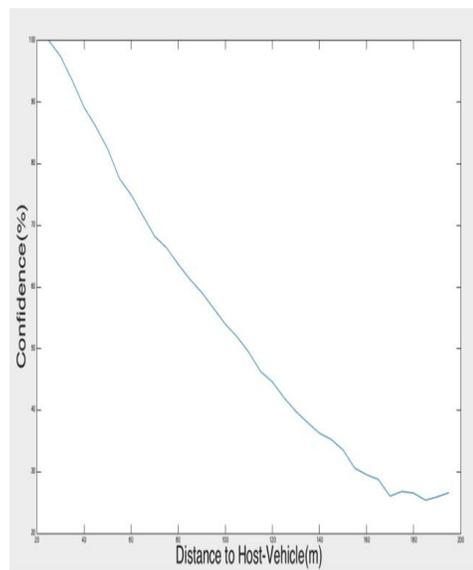
$$\frac{\sigma_1^2}{\sigma_2^2} = \frac{e^{-2kd_1}}{e^{-2kd_2}} \quad (4.17)$$

Similarly, k will be assigned any value from 0 to 0.09, and distant d_1 and d_2 are based on the distance when vehicle images are captured. Later, square error will be calculated between average ratios and theoretical ratios. At last, the coefficient k with least error will be taken to calculate visibility based on Equation (4.9).

The image below shows how image confidence declines with range after optimization. Table 4.3 below shows how estimation of visibility is improved by the new approach after optimization.



(a) Image Confidence Curve Under 150 meters Visibility



(b) Image Confidence Curve Under 500 meters Visibility

Figure 4.9: Image Confidence Curve

Ground Truth	New Approach of Estimation	Old Approach of Estimation
50m	60m	59m
100m	95m	103m
200m	180m	222m
350m	340m	279m
400m	380m	288m
500m	480m	Infinity

Table 4.3: Visibility Estimation Result by Using Different Approaches

CHAPTER 5

DISCUSSIONS AND CONCLUSIONS

In this chapter, the algorithm for distance estimation described in the previous chapter will be analyzed and discussed here, we first describe how several factors may impact the visibility estimation result in the new approach. Next, a way to combine a single measurement into a final estimation will be presented. Third, we will talk about the conclusions and future work.

5.1 FACTORS THAT MAY IMPACT ESTIMATION ACCURACY

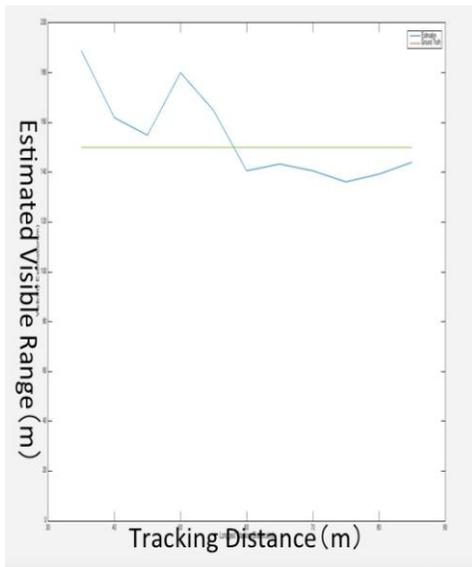
Based on the algorithm we proposed, we can easily find three factors, tracking distance, image acquisition speed and reference image position, may affect the final estimation result, as follows.

5.1.1 Tracking Distance

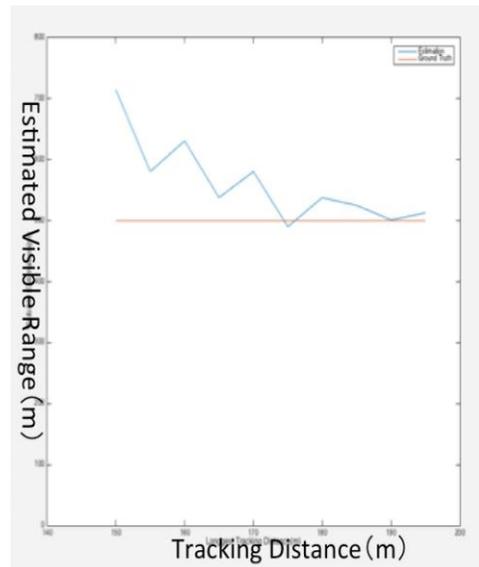
As we see in the algorithm, in order to calculate a relative accurate variance ratio, we need to track a vehicle for a certain distance. We may intuitively assume that the longer we track, the better the estimation result we will get. However, we still want to know whether there is a tracking range threshold where it is hard for us to improve accuracy dramatically even when we track longer distance for this method.

In order to test the algorithm, we set the different ground-truth visible ranges from 50 meters to 500 meters in the simulation environment for this experiment. Then, under a specific environment, we track a vehicle from initial distance of 25 meters to 90 meters and clip a vehicle image every five meters. Thus, we have a series of 13 images of the vehicle. To accomplish this experiment, we reduce number of images one by one from the last image, and thus we can calculate variance ratios for 75 meter tracking distance to 10 meters tracking distance.

Figure 5.1 presents how the tracking range under the ground-truth 150 meters and 500 meters affects visibility estimation results. The green and the orange line in the Fig 5.1 (a) and (b) are the ground-truth of 150 meters and 500 meters. As you can see, the longer we track a vehicle, the better result we can estimate. From the table below you may find we can get a relatively accurate visible range estimation result after tracking 40 % of ground-truth visible range.



(a) Tracking Distance vs. Estimated Visibility Result under Ground-Truth 150 meters



(b) Tracking Distance vs. Estimated Visibility Result under Ground-Truth 500 meters

Figure 5.1: Tracking Distance vs. Estimated Visibility Result

Tracking Distance of Ground Truth Visible Range	100m	200m	300m	400m	500m
10 %	106	247	417	INF	INF
20 %	99	205	402	895	INF
30 %	103	189	364	691	714
40 %	95	205	342	459	512
50 %	91	201	298	388	523
60 %	94			392	

Table 5.1: Tracking Distance vs. Visibility Estimation Accuracy

From this table, you can easily find that the algorithm will overestimate the visibility when tracking a short distance. This result coincidentally matches with article [2]. As it says, human eyes will estimate the position of front vehicles under foggy weather as further away than the fair weather. Though I did not prove this algorithm works as same as the way how human eyes work, it may be an interesting topic which we can work on in the future.

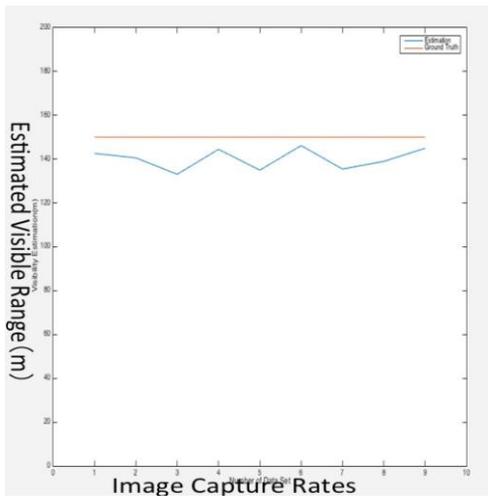
This result or experiment also explains why we need a rough result from the old visible range approach. This is because we need to track 40 % of the ground-truth visible range in order to get a good estimation.

5.1.2 Image Capture Rate and Initial Position of Reference Image

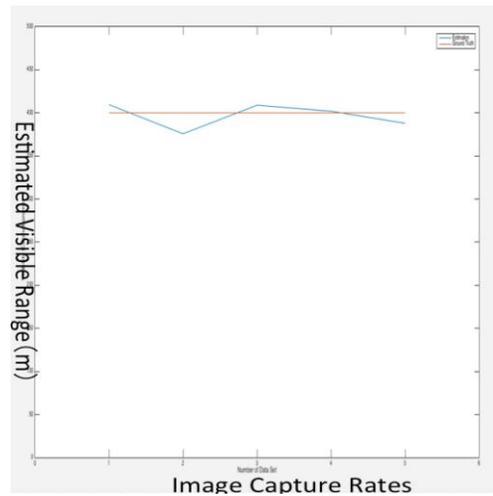
In this section, we will analyze whether the other two effects, image acquisition speed and reference image position, impact the result. Similarly, we set ground-truth visibility from 50 meters to 400 meters in the simulation environment and then we captured the vehicle image every 1.25 meters to 30 meters in the image capture rate experiment.

Attached below is the result of ground-truth visibility of 200 meters and 400 meters for different image acquisition speed. We can find that for different image acquisition speed, the visible estimation results only vary slightly. Thus a higher image captured rate does not contribute to the higher accuracy of visibility estimation.

It is very likely that since images are captured too closely, we only get a lot of redundant information which may not contribute a lot to the final estimation result. For this experiment, we calculated under other different ground-truth visibility environments as seen in the table listed below. Distance Difference here refers to the difference of the distance between host-vehicle and tracking vehicle in each two continuous images.



(a) Image Acquisition Speed vs. Estimated Visibility Result under Ground-Truth 200 meters



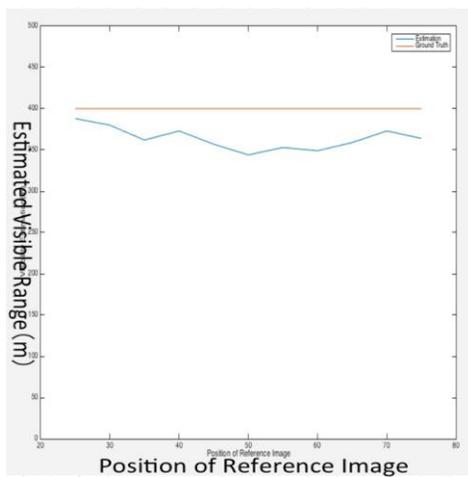
(b) Image Acquisition Speed vs. Estimated Visibility Result under Ground-Truth 400 meters

Figure 5.2: Image Acquisition Speed vs. Estimated Visibility

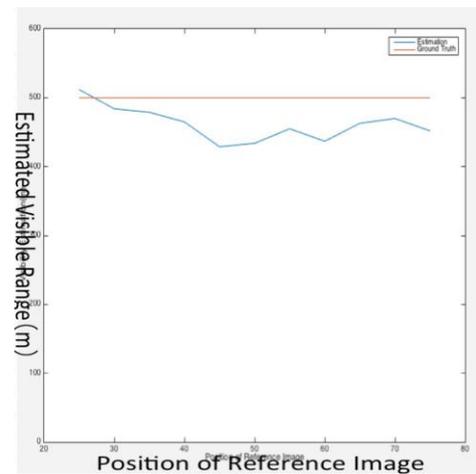
Distance Difference	50m	100m	200m	400m
1.25m	61	94.9	202	
2.50m	59	93	186	
3.75m	58	94	178	
5.00m	61	96	200	409
7.50m	57	91	181	
10.00m	59	95	180	376
15.00m	62	93	181	
20.00m	55	94	188	409
30.00m		95		
40.00m				402

Table 5.2: Image Acquisition Speed vs. Visibility Estimation Accuracy

What the next experiment did is to check the effect of the position of the reference image captured. Reference image here refers to the first image which contains a specific vehicle. In the experiment, different positions of reference images are checked and recorded under the same environment. Figure 5.3 below shows how position of reference image affects the estimation.



(c) Reference Image Position vs. Estimated Visibility under 400 meters Visibility



(b) Reference Image Position vs. Estimated Visibility under 500 meters Visibility

Figure 5.3: Reference Image Position vs. Estimated Visibility

Position of Reference Image	100m	200m	300m	400m	500m
25m	97	200	298	388	512
30m	95	198	296	380	484
35m	95	203	281	362	479
40m	91	203	290	373	429
50m	97	203	267	344	434
60m	102	228	316	349	437
70m	112	310	323	373	470

Table 5.3: Position of Reference Image vs. Visibility Estimation Accuracy

From Table 5.3 above, we can find that the position of the reference image has very little effect on the final result. However, an interesting pattern has been found where the estimation result generally falls down between 35 meters to 50 meters and then increases. This is caused by the factor of tracking range. As mentioned before, a limited tracking range may lead the algorithm to overestimate the visibility. Since, the distance for which we will track the vehicle in this experiment is fixed, then our actual tracking distance becomes less and less when we prolong the position of reference image.

5.2 COMBINATION OF SINGLE MEASUREMENTS

For each tracked vehicle, we can get a single measurement result. In order to combine all these single measurements to derive a final result, a Gaussian Mixture Model is chosen to combine the measurements. According to [25], the expression of model can be listed below.

$$\mu = \frac{1}{N} \sum_{j=1}^N z_j \quad (5.1)$$

$$\sigma = \frac{1}{N} \sum_{j=1}^N (z_j - \mu) \quad (5.2)$$

z_j here represents every single measurement which is treated equally, and σ , μ are variance and mean of the final estimation result.

Note that the weight of visibility we estimated in the specific positions and times will drop by rate $e^{-\frac{d_s}{t_s}}$ and $e^{-\frac{d_t}{t_t}}$ [11], where d_s and d_t are the spatial and temporal distances, and t_s and t_t are parameters controlling how fast the weight of a certain visibility estimation drops.

5.3 Conclusion and Future Work

In this thesis, we briefly introduce how we build simulation environments by PreScan, a new algorithm how to estimate the visibility under the bad weather. The algorithm we proposed here increases the accuracy of the visibility estimation by only using a monocular camera. Also, an optimization approach is proposed with the algorithm to further improve the performance of the algorithm.

There are several parts which we can improve in the future. First, after testing the algorithm in the simulation environment, we are planning to test the algorithm in the real environment. It will be important not only to validate how good the algorithm performs in the environment, but also for validating how good PreScan can simulate foggy environments. It may be used in the future to improve the algorithm which PreScan used to simulate fog.

Secondly, another interesting topic which we can study in the future is to analyze whether the variance comparison is similar to that used by the human eyes to recognize how visibility is corrupted by fog. Furthermore, the confidence curve deduced from the second step also can be used to improve algorithms like vehicle recognitions, based on machine learning or other approaches. Some algorithms cannot detect vehicles not due to bad performance itself, but possibly because the images has already been corrupted by fog. Thus, we can test how good our algorithm makes estimation by comparing it with the precision and recall rate of vehicle recognition algorithms for the same set of images.

The third part which we can work on is the incorporation of the vehicle recognition and tracking algorithms to make my method work automatically. At the moment, we just clip the vehicle images from background manually.

REFERENCES

- [1] R. H. Rasshofer, and K. Gresser, “Automotive Radar and Lidar Systems for Next Generation Driver Assistance Functions,” *Advances in Radio Science*, vol. 3, pp. 205–209, 2005.
- [2] V. Cavallo, M. Colomb, and J. Dore, “Distance Perception of Vehicle Rear Lights in Fog,” *Human Factors*, vol. 43, pp. 442-451, Fall 2001.
- [3] Nicolas Hautire, Jean-Philippe Tarel, and Didier Aubert, “Road Segmentation Supervised by an Extended V-Disparity Algorithm for Autonomous Navigation,” *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, vol. 11, no. 2, Jun. 2010.
- [4] T. Hagiwara, “Visibility Assessment Methods on Road-Development of Visibility Assessment Methods using Digital Images under Daytime Fog Conditions” (in Japanese), Tech. Rep. IEICE, 2004-PRMU 31, June 2004.
- [5] 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*, vol. 17, issue 1, pp. 8-20, April 2006.
- [6] W. Middleton, *Vision Through the Atmosphere*. University of Toronto Press, 1952.
- [7] E. Dumont and V. Cavallo, “Extended photometric model of fog effects on road vision,” *Transp. Res. Rec.: J. Transp. Res. Board*, no. 1862, pp. 77-81, 2004.
- [8] C Bush and E. Debes, “Wavelet transform for analyzing fog visibility,” *IEEE Intelligent Systems*, vol. 13, no. 6, pp. 66-71, 1998.

- [9] Mihai Negru and Sergiu Nedevschi, "Image Based Fog Detection and Visibility Estimation for Driving Assistance Systems," Intelligent Computer Communication and Processing (ICCP) 2013 IEEE International Conference.
- [10] S. Bronte, L. M. Bergasa, P. F. Alcantarilla, "Fog Detection System Based on Computer Vision Techniques," Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, MO, USA, October 3-7, 2009.
- [11] Michael Gabb, Sebastian Krebs, Otto Lohlein, and Martin Fritzsche, "Probabilistic Inference of Visibility Conditions by Means of Sensor Fusion," 2014 IEEE Intelligent Vehicles Symposium (IV), Dearborn, Michigan, USA, June 8-11, 2014.
- [12] Kenji Mori, Terutoshi Kato, Tomokazu Takahashi, Ichiro Ide, Hiroshi Murase, Takayuki Miyahara, Yukimasa Tamatsu, "Visibility Estimation in Foggy Conditions by In-Vehicle Camera and Radar," Proceedings of the First International Conference on Innovative Computing Information and Control (ICICIC'06).
- [13] Kenji Mori, Tomokazu Takahashi, Ichiro Ide, Hiroshi, Murase, Takayuki Miyahara and Yukimasa Tamatsu, "Recognition of Foggy Conditions by In-Vehicle Camera and Millimeter Wave Radar," Proceedings of the 2007 IEEE Intelligent Vehicles Symposium Istanbul, Turkey, June 13-15, 2007.
- [14] Nicolas Hautire, Jean-Philippe Tarel, and Didier Aubert, "Mitigation of Visibility Loss for Advanced Camera-Based Driver Assistance," IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, vol. 11, no. 2, Jun. 2010.

- [15] S. G. Narashimhan and S. K. Nayar, "Contrast Restoration of Weather De-graded Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 6, pp. 713-723, Jun. 2003.
- [16] Nicolas Hautire, Raphal Labayrade, and Didier Aubert, "Real-Time Disparity Contrast Combination for Onboard Estimation of the Visibility Distance," IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, vol. 7, no. 2, Jun. 2006.
- [17] Tan, R.T., "Visibility in Bad Weather from a Single Image," IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008, pp. 1-8, 23-28 Jun. 2008.
- [18] Kaiming He, Jian Sun, Xiaoou Tang, "Single Image Haze Removal Using Dark Channel Prior," IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, pp. 1956-1963, 20-25 Jun. 2009.
- [19] International Lighting Vocabulary, Central Bureau CIE, Vienna, Austria, 1987, no. 17.4 Commission Internationale de l'eclairage.
- [20] <https://www.tassinternational.com/prescan-overvie>
- [21] <http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/>
- [22] R. H. Rasshofer and K. Gresser, "Automotive Radar and Lidar Systems for Next Generation Driver Assistance Functions," Advances in Radio Science 3, 205-209, 2005 SRef-ID: 1684-9973/ars/2005-3-205
- [23] Zehang Sun, George Bebis, and Ronald Miller, "On-Road Vehicle Detection: A Review," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 28, no. 5, May 2006.
- [24] Zhou Junjing, Duan Jianmin and Yu Hongxiao, "Automotive Radar and Lidar Systems for Next Generation Driver Assistance Functions,"

Proceedings of the 10th World Congress on Intelligent Control and Automation, July 6-8, Beijing, China.

[25] C. M. Bishop, Pattern Recognition and Machine Learning, 2nd ed. Springer, 2006.

[26] PreScan Manual: A Simulation and Verification Environment for Intelligent Vehicle Systems.

[27] Vision-Based Advanced Driver Assistance: TI Hopes You'll Give Its Latest SoCs a Chance.

ABSTRACT**VISIBILITY AND CONFIDENCE ESTIMATION OF AN ONBOARD-CAMERA IMAGE FOR AN INTELLIGENT VEHICLE**

by

MINGLEI HUANG**August 2015****Advisor:** Dr. Abhilash Pandya**Major:** Electrical Engineering**Degree:** Master of Science

More and more drivers nowadays enjoy the convenience brought by advanced driver assistances system (ADAS) including collision detection, lane keeping and ACC. However, many assistant functions are still constrained by weather and terrain. In the way towards automated driving, the need of an automatic condition detector is inevitable, since many solutions only work for certain conditions. When it comes to camera, which is most commonly used tool in lane

detection, obstacle detection, visibility estimation is one of such important parameters we need to analyze.

Although many papers have proposed their own ways to estimate visibility range, there is little research on the question of how to estimate the confidence of an image. In this thesis, we introduce a new way to detect visual distance based on a monocular camera, and thereby we calculate the overall image confidence.

Much progress has been achieved in the past ten years from restoration of foggy images, real-time fog detection to weather classification. However, each method has its own drawbacks, ranging from complexity, cost, and inaccuracy. According to these considerations, the new way we proposed to estimate visibility range is based on a single vision system. In addition, this method can maintain a relatively robust estimation and produce a more accurate result.

AUTOBIOGRAPHICAL STATEMENT

Education:

- B.S. MSE, Zhejiang University of Technology, Hangzhou, China, 2015

Working Experience:

- Research Engineer, 2015 to current, Denso International America
- Student Research Assistant, 2014-2015, Wayne State University