DOCTORAL THESIS

Study on Evolutionary Approach for Fast Horizon Detection and Tracking in Images

Graduate School of Science and Engineering, Iwate University
Doctoral Course, Design & Media Technology
Ganbold Uuganbayar

March 2022

Acknowledgement

I want to express my gratitude to all those who helped me during this research.

First of all, I would like to extend my sincere gratitude to my supervisor Prof. Takuya Akashi for his patient academic guidance and great support in life. I am deeply grateful to Prof. Kouichi Konno, Prof. Tadahiro Fujimoto, Prof. Takamitsu Tanaka, and Assoc. Prof. Katsutsugu Matsuyama, who discussed in revising this thesis.

Additionally, I would like to thank Mrs. Hikaru Kaketa, who helped improve the research environment. I sincerely appreciate the help from Assoc. Prof. Junya Sato of the Gifu University. I want to thank every member of the Smart Computer Vision Lab. of Iwate University for their help and patience.

At last, gratitude also extends to my parents, wife Shurentsetseg, children, who have supported and cared for me all of my life.

ABSTRACT

In this research, horizon line detection and tracking using the evolutionary method in images are addressed. Due to the rapid development of computer vision and the increase in camera resolution, information from cameras is used to solve various problems, such as object detection, recognition, and tracking. Videos captured by optical systems are valuable for autonomous vehicles to perceive surrounding information for obstacle detection, remote control, and estimation of spatial orientation. Video processing in maritime and land scenarios is quite challenging because of random camera shaking and the processing time of high-resolution images.

A horizon line on maritime images is used to address the above challenges. For example, it is used for the following purposes: estimation of the spatial orientation of a camera/ship and image registration, which aligns consecutive frames into a one-coordinate system for object detection and object tracking. On the other hand, the performance of state-of-the-art horizon line detection methods is still limited in terms of both speed and accuracy. The focus of this thesis is to produce a fast and accurate method of horizon line detection. Especially, a horizon line in maritime images is mainly focused in this study, and horizon line detection and tracking are addressed. Also, as applications of proposed techniques, horizon detection in road

scenarios using genetic algorithms are proposed.

In order to achieve this goal, we searched for and analyzed state-of-art methods in horizon line detection. Then, we proposed our novel method, which efficiently integrates the advantages of existing methods. We formulated the horizon detection problem as a global optimization problem, and optimization is done by genetic algorithm (GA). The GA provides fast optimization utilizing fewer combinations of parameters compared to exhaustive search. We also introduced a fast estimation of global and local feature estimations for fast evaluation of fitness function. In addition, we adopted a coarse-to-fine approach to meet real-time processing requirements. We verified the performance of the proposed method using the Singapore Maritime Dataset (SMD) and Buoy-Dataset (BD), which are publicly available. The experimental results indicated that the proposed method could detect the horizon line more accurately than the compared methods. In particular, the median positional error and median orientation error of the proposed method were relatively smaller than those of all the compared methods in all datasets. The processing speed of our method was approximately 20fps for high-resolution images.

Additionally, we applied the proposed method for two applications. Those are detection of the horizon line in road images and detection of borderline detection for mowing machines. The experimental results show that the evolutionary approach can be used for other prominent line detection problems.

Contents

1 Introduction **1**

1.1 General Introduction 1

1.2 Goals 4

1.3 Challenges and Contributions 5

1.4 Thesis Structure 6

2 Background **8**

2.1 Evolutionary Algorithms 8

2.1.1 Historicial Context and Advantages of Genetic Algorithm . 8

2.1.2 Methodology of GA 10

2.1.3 Applications of GA 11

2.2 Horizon Line 13

3 Fast horizon detection in maritime images using genetic algorithm **16**

3.1 Introduction 17

3.2 Related Work 19

3.3 Proposed Method 25

3.3.1 Overview 25

3.3.2 Optimization by Genetic Algorithm 26
3.3.3 Global Feature 31
3.3.4 Local Features 33
3.4 Experimental Results 36
3.4.1 Dataset and Evaluation Criteria 36
3.4.2 Parameter Setting 38
3.4.3 Effectiveness of EVP and Parameter Analysis 39
3.4.4 Comparison Results and Consideration 41
3.5 Conclusion 47

4 Possibility of practical application **54**

4.1 Horizon line detection in road images using genetic algorithm . . . 55
4.1.1 Introduction 55
4.1.2 Methodology 57
4.1.3 Experimental Result 62
4.1.4 Conclusion 64
4.2 Local texture based borderline detection of mowing using genetic algorithm 65
4.2.1 Introduction 66
4.2.2 Related Work 67
4.2.3 Methodology 68
4.2.4 Experimental result 73
4.2.5 Conclusion 74

5 Conclusion **76**

A Details of the SMD and Comparison Result **77**

B Dataset Captured by Fisheye Camera **84**

Chapter 1

Introduction

1.1 General Introduction

In this research, horizon line detection and tracking using the evolutionary method in images are addressed. Due to the rapid development of computer vision and the increase in camera resolution, information from cameras is used to solve various problems, such as object detection, recognition, and tracking. Videos captured by optical systems are valuable for autonomous vehicles to perceive surrounding information for obstacle detection, remote control, and estimation of spatial orientation. Video processing in maritime and land scenarios is quite challenging because of random camera shaking and the processing time of high-resolution images. To solve the these challenges, a horizon line is used for the following purposes: estimation of the spatial orientation of a camera/vehicle and image registration, which aligns consecutive frames into a one-coordinate system for object detection and object tracking. On the other hand, the performance of state-of-the-art horizon line detection methods is still limited in terms of both speed and accuracy. The focus

of this thesis is to produce the fast and accurate method of horizon line detection. Especially, a horizon line in maritime images is mainly focused in this study, and horizon line detection and tracking are addressed. Also, as applications of proposed techniques, horizon detection in road scenarios using genetic algorithms are proposed.

In order to achieve this goal, we searched for and analyzed state-of-art methods in horizon line detection. Then, we proposed our novel method, which efficiently integrates the advantages of existing methods. We formulated the horizon detection problem as a global optimization problem, and optimization is done by genetic algorithm (GA). The GA provides fast optimization utilizing fewer combinations of parameters compared to exhaustive search. Simple GA requires the generation of an initial population for each frame of a sequence, and it is time-consuming. Therefore, we used evolutionary video processing (EVP). The EVP generates the initial population once at the initial frame of a sequence and inherits a population of the last generation into an initial generation of the next frame. As a result, the EVP improved optimization accuracy and reduced processing time. In an EVP, a fitness function is used to guide the simulation toward an optimal solution, and it evaluates the goodness of each individual. Therefore, designing the fitness function is very important for quick convergence on an appropriate solution, and it has a significant impact on computational time. The fitness function should precisely evaluate how to fit a given solution and should be fast to compute. Therefore, we also introduced a fast estimation of global and local feature estimations for the fast evaluation of fitness function. In addition, we adopted a coarse-to-fine approach to meet real-time processing requirements.

The local feature-based extract the horizon candidates using the edge information and use consecutive filtering to find the final solution. A limitation of these methods is that if the candidates cannot be extracted from edge information in the previous stage, they are not considered in the next stage, even though these candidates are survival candidates in the filtering of the next stages. Unlike these methods, our method is a metaheuristic optimization method, and local and global features are concurrently utilized to evaluate each candidate. The proposed method does not extract edge information from multi-scale images, and even for blurred input images, it can detect the horizon line.

On the other hand, the existing global feature-based methods used a global feature as an optimization criterion to optimize the horizon line parameters. These methods are not dependent on edge information, and they can detect the horizon line on blurred and noisy images. But these methods require computational cost because they calculate the global feature for all candidates, and they use exhaustive search to optimize the parameters. Thus, we created a probability map of the horizon line in the pre-processing stage and used it as a global feature factor in the fitness function of the coarse optimization. In coarse optimization, local and global features are concurrently utilized to evaluate each candidate. In the fine-tuning step, the global feature factor effects were weaker than those of the local feature factor for the optimization because the position of the horizon line was roughly determined in the coarse optimization. Therefore, we assumed only the local feature factor in the fine optimization for fast evaluation of fitness function.

We verified the performance of the proposed method using the Singapore Maritime Dataset (SMD) and Buoy-Dataset (BD), which are publicly available. The

SMD consists of onboard and onshore videos. The onboard videos were captured by a camera mounted on a moving board, and the onshore videos were captured by a static camera installed onshore. The videos contain complex maritime scenes that have strong noise caused by wakes and waves and color changes in the sea. The resolution of the SMD videos was 1920×1080 pixels. The BD consists of videos captured by a camera mounted on a floating buoy with a resolution of 800×600 pixels. A challenge for the onboard videos of SMD and BD is the large variation in the orientation and position of the horizon line between adjacent frames. In addition, we compared the performance of the proposed method with that state-of-the-art method, which used the same datasets. The experimental results indicated that the proposed method could detect the horizon line more accurately than the compared methods. In particular, the median positional error and median orientation error of the proposed method were relatively smaller than those of all the compared methods in all datasets. The processing speed of our method was approximately 20fps for high-resolution images.

Additionally, we tested the proposed method to detect the horizon line in road images captured by a fish-eye camera. In the experimental result, the absolute error of position and absolute error of orientation was low and stable. It indicates that the proposed method can detect the horizon line in road images.

1.2 Goals

The goal of this research is to explore methods that to detect the horizon from video captured by a vehicle-mounted camera. In order to achieve this goal, we

done following sub research and works.

- To clarify what is the horizon and to study how it is defined in recent researches;
- To analyze the state-of-art methods in horizon line detection;
- To determine the model to provide consistent and robust identification of the horizon line;
- To make the reliable method to detect the horizon line and test it.

1.3 Challenges and Contributions

Although several horizon detection methods were proposed, each presents certain challenges. A horizon detection methods that edge extraction-based (local feature-based) are fast processing, but these are weak in noises and blurring images. Contrary, horizon detection methods, which are optimization-based (global feature-based) methods, are relatively robust for noise and blurring of input images, but these methods are computationally expensive and not available for real-time processing. The machine learning-based techniques are highly dependent on the training sets and computationally expensive. Therefore, making a fast and reliable detection method of the horizon is a challenging task. In this work, we propose a fast and reliable method, and our contributions are following:

- A metaheuristic approach for borderline detection, which is independent from edge-extraction unlike existing deterministic methods;

- A combination of the GA and coarse-to-fine approach for fast processing;
- Integration of local and global features in the optimization criteria for high performance;
- A quick estimation of the local and global features for fast processing.

Our main contribution is the metaheuristic method GA based, which is applied for borderline detection. It is independent of edge-extraction, unlike existing deterministic methods such us Hough and Radon transform-based methods. Thus, we tested the effectiveness of the metaheuristic method on two applications of borderline (single line) detection. Those are the detection of the horizon line in road scenarios images and the boundary line for a mower. In detecting the horizon line in road scenarios images, GA is utilised to optimise line parameters, and local features are used for the objective function. The local features are derived from vanishing line characteristics, same as the horizon detection method in maritime. In the boundary line detection for a mower, GA is also used to optimise line parameters, and grey level co-occurrence (GLCM) is used to distinguish the cut grass and the uncut grass. The result shows that the evolutionary method can be use in borderline line detection applications.

1.4 Thesis Structure

Chapter 2 provides a brief background in the evolutionary algorithm, which is used for optimization in our proposed method, and also discusses the horizon. Our proposed method, which is the" fast horizon detection method in maritime images us-

ing genetic algorithm," is introduced in chapter 3.

In chapter 4, some practical applications of the proposed method are introduced. We improved and tested our proposed method for horizon detection in images that are captured in road scenarios. It is introduced in section 4.1. Also, we proposed the evolutionary method for detecting the boundary line for mower. It is introduced in detail in the section 4.2. Finally, we conclude with an overview of this study in chapter 5.

Chapter 2

Background

2.1 Evolutionary Algorithms

An evolutionary algorithm (EA) is a generic population-based metaheuristic optimization algorithm. It is inspired by the process of the natural selection process. Since the genetic algorithm was proposed by John Holland [1], the study of the evolutionary algorithm has emerged as a popular research field [2]. In this thesis, we used the genetic algorithm, which is the most classic evolutionary algorithm. Thus, to introduce the working mechanism of an EA, we introduce, illustrate, and discuss genetic algorithms to understand EAs. We also discuss the history of genetic algorithms, current applications, and especially useful in computer vision.

2.1.1 Historicial Context and Advantages of Genetic Algorithm

The GAs are a type of optimization algorithm, meaning they are used to find the optimal solution(s) to a given combinatorial problem that maximizes or minimizes a particular function. Since genetic algorithms are designed to simulate an evolution

process, it involved a simulation of Darwinian survival of the fittest and crossover, recombination, mutation, and inversion that occur in genetics. Mitchell [3] stated that this population-based method was a massive innovation because previous genetic algorithms only used mutation to drive evolution. In 1975, Holland presented the first comprehensive study of genetic algorithms to solve optimization problems as his doctoral dissertation. In the 1970s and 1980s, research on genetic algorithms rapidly increased due to computer science technology development.

Computer scientists also realized the limitations of conventional programming and traditional optimization methods for solving complex problems. Researchers found that genetic algorithms can find solutions to a wide range of search problems. Genetic algorithms can simultaneously test many points from all over the solution space, optimize with either discrete or continuous parameters. Especially, Genetic algorithm provides several optimum parameters instead of a single solution and works with many different kinds of data. These advantages allow genetic algorithms to produce stunning results when traditional optimization methods fail miserably. GA has many advantages over these traditional methods. It is efficient for global optimization problems because it consists of candidates' solutions. Also, GA does not require additional information that is not related to the meaning of the possible solutions. The only mechanism that guides their search is the numerical value of the candidate's solution fitness based on the definition of fitness. This allows the search space to be noisy, non-linear, and free of derivatives. It means that GA can be used in more situations than traditional algorithms.

2.1.2 Methodology of GA

Here, I explain an algorithm of a simple GA. Since GA is derived from a biological process, much of the terminology is related to biology. The basic components of GA are following:

- a population of candidate (each candidate consists a chromosome)
- a fitness function for optimization
- selection of which chromosome will reproduce
- a genetic operators(crossover, mutation) to produce the next generation

To solve the optimization problem with GA, a representation of a solution to the problem as chromosomes and a formulation of the fitness function must be determined, which are essential for optimization accuracy and speed. The candidates (individual) are search points in search spaces and each individual has chromosomes, which are optimized parameters. The chromosomes also consist of a binary string and users can determine the length. A fitness function is utilized to guide the simulation toward an optimal solution, and it evaluates the goodness of each individual. Therefore, designing the fitness function is very important for quick convergence on an appropriate solution, and it has a significant impact on computational time. The fitness function should precisely evaluate how to fit a given solution and should be fast to compute.

Once the genetic representation of inidividual and the fitness function are defined, GA proceeds to initialize a population of solutions. Typically, the initial population is generated randomly, allowing the entire range of possible solutions. Then,

the fitness function evaluates the fitness of the candidates. After the evaluation of the fitness, genetic operations, selection, crossover, and mutation are performed to produce the next generation.

Selection tries to apply pressure upon the population like that of natural selection found in biological systems. Candidates with poorer fitness are weeded out. Contrary, and candidates with higher fitness have a greater chance of promoting genetic information within the next generation. To implement this idea in an algorithm, a roulette wheel selection is used to select the individuals whose fitness is high with high probability.

Crossover allows solutions to exchange genetic information in a way similar to that used by natural organisms undergoing sexual reproduction. One method is a uniform crossover which changes the genes of the two selected individuals. The positions of the changed genes are randomly determined.

The standard mutation inverts genes with a low probability of maintaining diversity. In other words, this processing is to preserve premature convergence to local optima. By iterating, these operations parameters are optimized, and an elite individual(it has the highest fitness) is obtained as a global optimum. These genetic operations are repeated until a termination condition has been reached. A common terminating conditions are sufficient solutions are found and fixed number of generations.

2.1.3 Applications of GA

The advantages of genetic algorithms as problem-solving strategies are quickly gaining recognition among researchers of diverse areas of study. Some advantages

of GA are the following. First, GA can optimize various problems such as discrete functions, multi-objective problems, and continuous functions. Second, it does not need derivative information. Genetic algorithms know nothing about the problems they are being applied to. Instead of relying on specific information about a problem, as do many other search techniques, a fitness function is employed to ascertain whether the random changes resulting from crossover and mutation have made improvements to the overall fitness with data to a theoretical model. Also, GA has excellent parallel capabilities. Due to the above advantages, GAs are applied in many fields. Several fields are following:

- Transportation[4, 5];
- Electronis(VLSI)[6, 7];
- Trainging and designing artificial inteligence systems[8, 9];
- Bioinformatics[10];
- Robotics[11, 12];
- Computer vision[13, 14, 15];

This thesis will focus more on the GA application of computer vision. The main tasks of computer vision are to obtain and understand meaningful information from images and videos. The various processes that perform these tasks are often confronted with situations that require optimization. Especially, the solution spaces are a vast and complex landscape. In that situation, GA is used powerful tool for solving the optimization problems occurring in computer vision. For example, GA is involved with respect to different computer vision tasks: image segmentation.

[16, 17], feature detection and selection [18, 19], template matching [20, 21], visual tracking [15, 22], face recognition [23, 24, 25].

2.2 Horizon Line

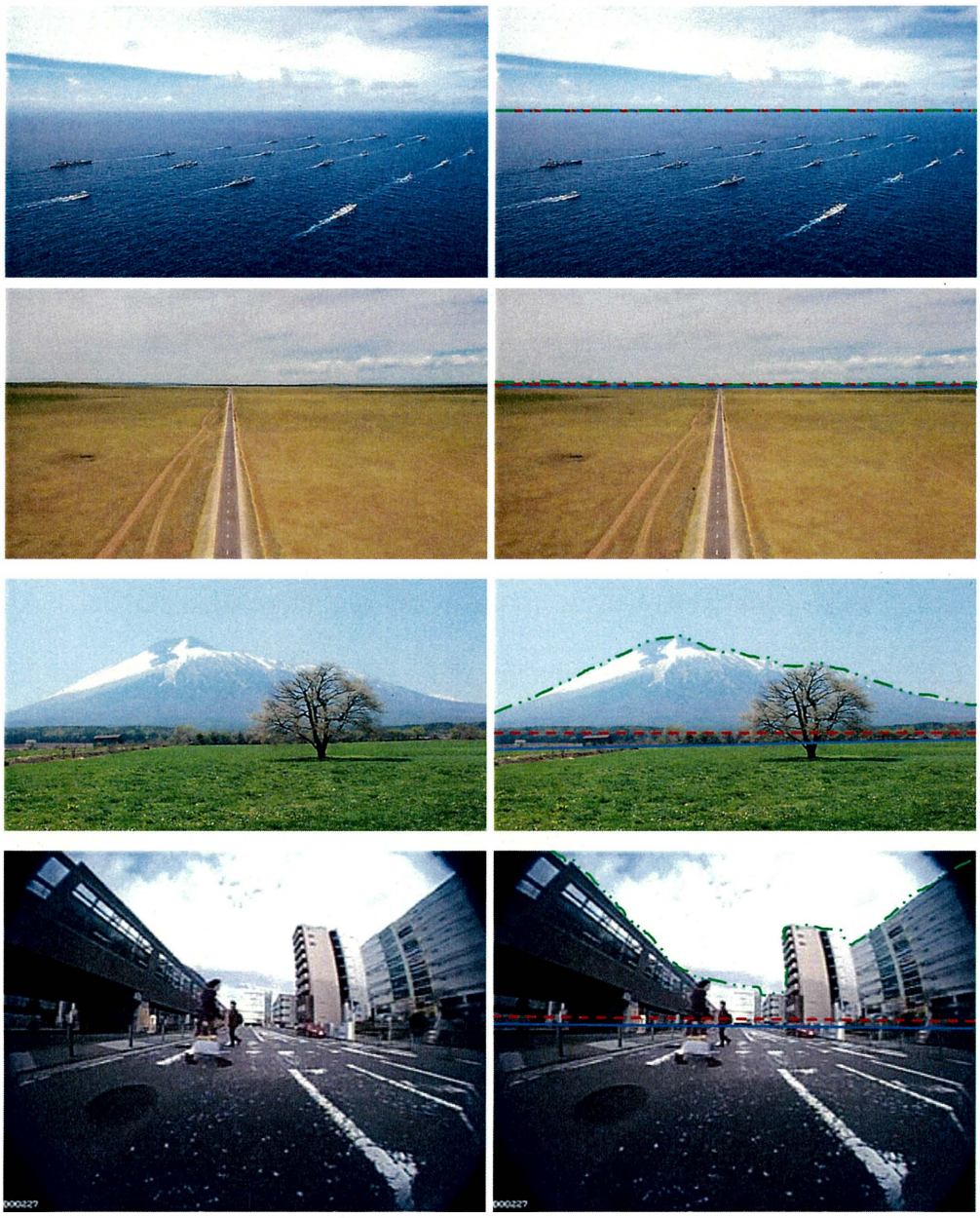
The horizon is important for space perception and self-orientation [26, 27]. Gibson's ground theory of space perception identified that the vanishing line of ground (horizon) helps to solve problems often encountered in such applications, like the lack of initial estimates for the depth of structures and height of objects. The horizon is important for space perception and self-orientation as well as in computer vision.

Humans have no problem completing this task easily. But, machines still have a big problem finding images given only such a semantic description. This is not unintentional, given that almost every image needs to be interpreted. Find and identify objects (trees, cars, and humans), estimate perspectives, etc. We extract all the information in just a few hundred milliseconds per image. The analysis is not very thorough, but it is detailed enough to form a superficial representation of the image content in our minds. The term "horizon" is widely used by "where the sky meets the ground". However, it becomes more complex caused by image scenarios and purpose to estimate. In previous studies, three different clues are assumed as "horizon," and these are explained in detail here.

The first cue is an astronomical horizon. It is defined by a horizon plane that is perpendicular to gravity and located at the same height as the observer. It is not dependent on the slant of the ground surface nor on the presence of occluding

objects, and it is orthogonal to gravity [28]. Second is a visible horizon, which is the boundary line above which separates the regions, sky and not sky. It is usually not straight-line [29, 30, 31]. The third is the horizon line which is the vanishing line of the ground plane. Most of these researchers assume that the ground/sea plane is almost plate [14, 32, 33, 34]. As shown in Fig.2.1(top row), these cues are overlapped when the ground plane is almost flat and there are no occluding objects. However, there exists plants, mountains, and buildings, or ground surface is slanted, estimation of horizon line becomes complex as shown in Fig.2.1(3-rd and 4-th rows).

For ground and sea surface vehicle navigation, the monocular vision systems have been proposed to investigate the environmental information [35, 36, 33, 34]. Like the above works, the vanishing line of the ground/sea surface is referred to as the horizon on the rest of this work. This work concentrates on horizon estimation from videos captured by a camera in vehicles.



(a) (b)

Figure 2.1: Scenarios to show horizon cues. (a) Original images. (b) The variation of horizon estimation are shown as colored lines. The red and green dashed lines indicate the visible horizon and astronomical horizon, respectively, and the blue line indicates the horizon line which vanishing line of ground plane.

Chapter 3

Fast horizon detection in maritime images using genetic algorithm

In this chapter, I explain a fast horizon detection in maritime images using genetic algorithm. Horizon detection is useful in maritime image processing for various purposes, such as estimation of camera orientation, registration of consecutive frames, and restriction of the object search region. Existing horizon detection methods are based on edge extraction. For accuracy, they use multiple images, which are filtered with different filter sizes. However, this increases the processing time. In addition, these methods are not robust to blurting. Therefore, we developed a horizon detection method without extracting the candidates from the edge information by formulating the horizon detection problem as a global optimization problem. A horizon line in an image plane was represented by two parameters, which were optimized by an evolutionary algorithm (genetic algorithm). Thus, the local and global features of a horizon were concurrently utilized in the optimization process, which was accelerated by applying a coarse-to-fine strategy. As a result, we could

detect the horizon line on high-resolution maritime images in about $50ms$. The performance of the proposed method was tested on 49 videos of the Singapore marine dataset and the Buoy dataset, which contain over 16000 frames under different scenarios. Experimental results show that the proposed method can achieve higher accuracy than state-of-the-art methods.

3.1 Introduction

Autonomous surface vehicles have been developed for applications such as environmental protection and coastal guard [37]. These vehicles usually utilize radar, light detection and ranging, inertial systems, and GPS for navigation and obstacle detection [38]. With the rapid development of computer vision and the increase in camera resolution, information from cameras is used to solve various problems, such as object detection, recognition, and tracking. Videos captured by optical systems are valuable for autonomous surface vehicles to perceive surrounding information for obstacle detection, remote control, and estimation of the spatial orientation. Video processing in maritime scenarios is quite challenging because of random camera shaking caused by waves and the processing time of high-resolution images.

To address the above challenges, a horizon on maritime images is used for the following purposes: estimation of the spatial orientation of a camera/ship and image registration, which aligns consecutive frames into a one-coordinate system for object detection and object tracking [39, 40, 41]. In addition, the horizon is used to determine the region of interest to reduce the processing time and false detection [41, 42].

Therefore, the accurate detection of the horizon line is critically important for maritime image processing as an initial step. However, the detection of the horizon line faces several issues caused by complex maritime environments, such as waves, ocean color, light changing, and partial occlusions by maritime objects. Another challenge is that the pixels of the horizon line features are fewer than those of the entire image[43]. Thus, an accurate extraction of the horizon line features is required.

In the last two decades, several approaches have been proposed for detecting the horizon line in maritime environments. In the maritime scenario, the horizon line is generally represented as a straight line because the sea surface can be assumed to be flat and vanishes into a line on the image plane. Related works can be divided into local feature-based methods[44, 45], global feature-based methods[46, 47] and hybrid methods[32, 33, 34, 41]. The local feature-based methods extract a line segment of the horizon from local features such as edge information using line detection techniques such as Hough, and Radon. Global feature-based methods optimize the horizon line parameters using the horizon features on the entire image. Although hybrid methods, which utilize both global and local features, can achieve higher accuracy, the estimation of these features for all horizon candidates requires considerable computation time.

In this work, we formulate the horizon line detection as an optimization problem and propose a new method called coarse-to-fine evolutionary method, abbreviated as CFEM. As suggested by the name, we adopted the genetic algorithm (GA) to optimize the parameters of the horizon line. The GA is an evolutionary algorithm that can optimize the parameters using a criterion, which concurrently assumes global

and local features. In addition, a coarse-to-fine approach was adopted to accelerate the processing. First, the GA optimizes the parameters of the horizon line on down-sampled image of input image by using an optimization criterion that utilizes both global and local features. Then, the coarsely estimated parameters from the previous step are fine-tuned on a higher resolution image within a narrow range of line parameters with the GA using local feature estimation.

Our main contributions are:

- A metaheuristic approach, which is independent from edge-extraction unlike existing deterministic methods;
- A combination of the GA and coarse-to-fine approach for fast processing;
- Integration of local and global features in the optimization criteria for high performance;
- A quick estimation of the local and global features for fast processing.

In Section 3.2, we discuss related works on horizon detection and the GA. Section 3.3 outlines the proposed method for horizon detection. The experimental results and details of the parameters are discussed in Section 3.4. Finally, we conclude with an overview of this study in Section 3.5.

3.2 Related Work

For most horizon detection methods in the maritime scene, the horizon is considered a straight line. Related works can be classified as local and global feature-based methods. The local feature-based methods [44, 45] identify a prominent line as a

(a)

(b)

(c)

Figure 3.1: Result of transformation-based method [44]. (a) An input image. (b) edge map extracted by Canny edge method on the grayscale image of an input image. (c) red lines show the top 100 candidates lines with largest strengths in Hough spaces and a green line shows ground truth of the horizon line.

horizon line using line segment extraction methods from an edge information of the input image. Hough transform[44], Radon transform[32], and line segmentation algorithms are mostly used to extract line features from an edge information.

Although the horizon line can be detected in real-time [44, 45], it cannot be established by a prominent line owing to oceanic color differences and noise caused by waves and blurring of an input image shown in Fig. 3.1. Another limitation of this approach is that it is difficult to distinguish the horizon line from the extracted lines[43].

Several methods that improve the local feature-based method have been introduced. Fefilatyev et al. [41] introduced a candidate-first approach. First, a few candidate lines are selected by the Hough transform based on an edge map. Then, a global feature of the horizon line is used to find an optimal solution from candidates, and calculates the difference of the color distributions in two regions divided by the candidate line. A similar method was proposed by Lipschutz et al.[48], in which a color histogram was used to model the color-space distribution of two regions to reduce the processing time. Prasad et al.[49, 32] used multi-scale edge extraction approaches to extract edges from multiple images filtered with different filter sizes and accurately extract the edge information of the horizon line. In MSCM-LiFe[49], Canny edge detection and Hough transform are used to select the first modal candidates on multiple images filtered with different filter sizes and the maximum intensity variation is calculated to select the second modal candidates. Then, to select the final solution from the candidates, the goodness of the two modals and the geometric proximity of the pair of modals are measured. In MusCoWERT[32], a weighted edge map is computed for each image filtered with a multi-scale filter and candidates are selected by the Radon transform from the weighted edge map. Then a voting system is used for the final solution from all the candidates. Jeong at el.[50] combined multi-scale edge detection and convolutional neural network

for reliable edge extraction. Then, they used linear curve fitting along with median filtering to find an optimal horizon line. In experimental results of [50], the above multi-scale approaches achieved the highest accuracy. But, they required expensive computation for real-time processing in high-resolution images for detecting the optimal horizon line.

In addition, global feature-based methods [47, 46, 44, 48] have also been proposed. The global feature is used as an optimization criterion to optimize the horizon line parameters. The horizon line can be represented by two parameters, orientation and position. Ettinger et al. [46, 47] considered that the horizon line divides an image into two different regions, namely sky and sea, thus the difference of the two regions was used as an optimization criteria. To find the optimal parameters of the horizon line, they calculated the statistical distance metrics of distributions in the two regions for all combinations of the horizon line parameters. These methods are not dependent on edge information and they can detect the horizon line on blurred and noisy images. They achieved real-time processing on a low-resolution image using a coarse-to-fine approach. However, the result of [32] shows that this method requires tens of seconds to detect the horizon line on the high-resolution image because it requires calculation of the statistical distribution of the two regions for all candidates and uses exhaustive search to optimize the parameters.

Recently, sky-sea region extraction methods have been proposed to reduce the processing time by restricting the search region [33, 34]. Liang et al.[34] extracted the sky-sea region using probabilities that were distributed on vertically divided regions by weighted textures. Then, candidate lines were extracted from the sky-sea region using an edge detector and Hough transform. Finally, a voting method was

applied to obtain the final solution. The extraction of the sea-sky regions reduced the processing time and false detection. However, only part of the horizon line is obtained when there are occlusions near the horizon and a large angle gradient along the horizontal axis. Jeong et al.[33] also vertically divided an image into several regions and extracted the sky-sea region using the difference between the color distributions of the consecutive regions. The difference between the two regions was calculated by the Bhattacharyya distance, and a region with the largest distance was selected as the sky-sea region. Then, multi-scale edge detection was applied to the sky-sea region and merged into one edge map. Finally, the Hough and least-squares methods were sequentially used to find the horizon line.

Except for Ettinger's methods [46, 47], the above methods extract the candidates of horizon using local features as edge information and use consecutive filtering on several stages, which are based on features of the horizon. One limitation of this approach is that filtered candidates in the previous stage cannot be considered in the next stage, even though these candidates have survival candidates in the filtering of the next stages. Thus, multi-scale approaches have been proposed to extract sufficient candidates in an early stage. However, these methods require along processing time. In addition, Jeong et al.[33] stated that the methods that depend on edge information cannot detect the horizon line when the input image is blurred or the boundary between the sky and sea region is gradually changed.

Therefore, in our previous study[51], we proposed a novel method that optimizes the parameters of the horizon line. For the fast and accurate detection of the horizon, we considered several improvements. First, we used global optimization algorithms to solve the horizon detection problem, and applied GA for efficient op-

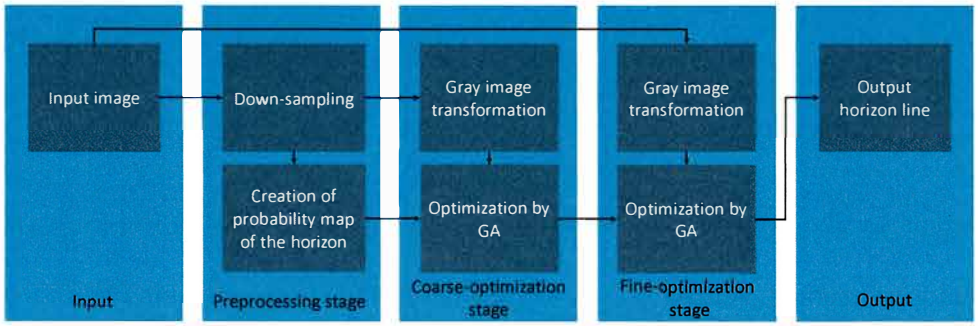


Figure 3.2: Diagram of the proposed method.

timization to reduce the processing time. Second, instead of considering all pixels of an image for each combination of the horizon line parameters, such as in [46], we defined the local features of the horizon line using a vanishing line characteristic. The result of [51] shows that utilization of the local features of the horizon for optimization criteria can reduce the processing time and can increase an accuracy. However, this method has limitation in certain scenarios such as for mostly occluded horizon and drastic changes in the color of the sea.

Our study extends the previous method [51] for improving accuracy by adding the factor of global feature to the optimization criterion. Consequently, the accuracy of the proposed method was improved in frames, whose sea color drastically changes. One advantage of optimization-based approaches is that several optimization criterion can be used as fitness functions. In our case, local and global features were utilized concurrently in the fitness functions.

3.3 Proposed Method

3.3.1 Overview

In this section, we introduce a horizon detection method called coarse-to-fine evolutionary method. A diagram of the proposed method is presented in Fig. 3.3. The proposed method consists of three steps. First, an input image is down-sampled and a probability map of the horizon is created during the pre-processing stage. Before down-sampling, the input image is filtered by a Gaussian filter. The probability map of the horizon is used for optimization criterion as a factor of the global feature of the horizon in the next step. Subsequently, coarse-to-fine optimization is performed for detect the horizon line. Coarse-to-fine approaches are widely used in computer vision to improve the efficiency[52, 53]. In the coarse-optimization stage, the parameters of the horizon are roughly optimized on the gray-scale image of the down-sampled image. The global and local features are concurrently utilized in the optimization criterion. Finally, the fine-optimization of the parameters is performed at high-resolution to improve the accuracy. The fine-optimization stage is performed in a narrow region close to the parameters that are roughly optimized by coarse optimization. The results of each stages are shown in Fig. 3.3.

For quick optimization of the horizon line parameters, GA is used, which provides optimization utilizing fewer combinations of parameters compared to exhaustive search. The GA is broadly applied to efficiently solve combinatorial optimization problems in computer vision such as template matching and object detection[23, 14, 13].

For all optimization methods, an optimization criterion significantly affects the

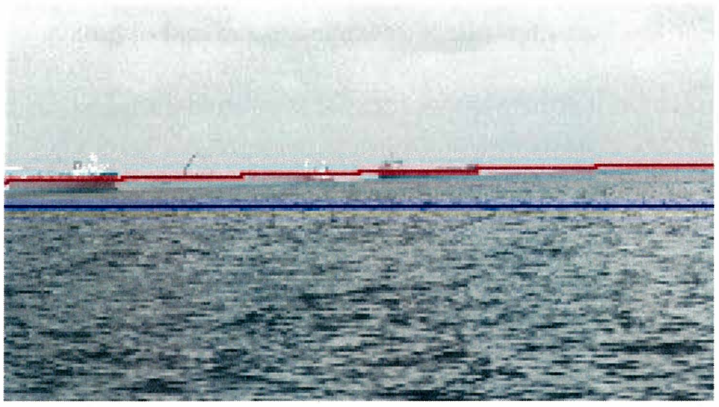
processing time and accuracy. We introduced the fast estimation of local and global features for the optimization criteria. In the next subsection, for horizon detection, a GA and utilization of local and global features are presented.

3.3.2 Optimization by Genetic Algorithm

The horizon is projected onto a single line in an image plane. Therefore, the problem of the horizon detection can be regarded as a global optimization problem. GA is a popular evolutionary algorithm for global optimization and has been applied to various combinatorial optimization problems in computer vision [23, 13] and simple GA algorithm is shown in Algorithm 1. Thus, we used the GA to optimize the parameters of horizon line in both coarse and fine optimization. A simple GA requires the generation of an initial population of individuals for every frame of a sequence. Each individual within the population represents a possible solution, a so-called candidate. For every iteration, the individuals of the population are evaluated by a fitness function, and then updated by genetic operations such as selection, crossover, and mutation.

The generation of an initial population for each frame of a sequence is time-consuming. Therefore, we used evolutionary video processing (EVP) [15]. It generates the initial population once at the initial frame of a sequence and inherits a population of the last generation into an initial generation of the next frame. Akashi et al. [15] stated that evolutionary video processing can improve optimization accuracy and reduce processing time. In addition, we used an elite saving strategy to improve the efficiency of GA, which is the process of preserving previous high-performance solutions from the current generation to the next. To solve the combinatorial prob-

(a)



(b)

(c)

Figure 3.3: The result of the coarse and fine stage optimization (a) input image (b) result of coarse-optimization (c) result of fine-optimization. The elite candidate line is shown by the red, line of initial parameter value on each optimization step is shown by blue and the ground truth of the horizon line is shown by the green.

lem with GA, a representation of a solution to the problem as chromosomes and a formulation of the fitness function must be determined, which are essential for optimization accuracy and speed, and will be explained in the following sections.

Algorithm 1: GA(N_g,N_p,c,m)

//initialization:

$m = 0$;

P_m = N_p number of randomly generated individuals;

//evaluate P_m**:** compute $fitness(i)$ for each $i \in P_m$;

while $N_g > m$ **do**

//1.Selection and copy:

select the individuals of P_m by roulete selection;

insert the selected individuals into P_{m+1};

//2.Crossover:

select $N_p \times c$ individuals of P_{m+1}, pair them up;

produce new individuals from the pairs;

insert the new individuals into P_{m+1};

//3.Mutation:

select $N_p \times m$ individuals of P_{m+1},;

invert a randomly selected bit in selected individuals;

//4.Evaluate P_{m+1} **and increase**

compute $fitness(i)$ for each $i \in P_{m+1}$;

$m = m + 1$;

end

Result: the fittest individual from P_{Ng}

Representation of Parameters in Chromosomes

The chromosome of an individual is often represented by bit strings because of it is faster than the real coded GA in the processing of crossover and mutation

operations[23]. The chromosome contains a set of parameters, that are necessary to solve a problem. In our case, the horizon can be a straight line, and the ground truth of the horizon line is given by a straight line on the datasets [40, 54]. Hence, a horizon can be represented by two parameters of the straight line: vertical position Y and orientation angle θ. An adjustment values for the two parameters decoded as a chromosome of individuals. These were the orientation ρ and the height adjustments of the horizon line λ. Y and θ were calculated as follows,

$$Y = (Y_0 + \lambda), \tag{3.1}$$

$$\theta = (\theta_0 + \rho), \tag{3.2}$$

where Y_0 is the initial vertical position, and θ_0 is the initial orientation angle. In the coarse-step optimization, the initial value of the vertical position $Y_0 = H/2$ was located in the center of an image, and the initial value of orientation $\theta_0 = 0$ was set parallel to the horizontal edge of the image. In the fine-step optimization, the initial parameters of the horizon line were the elite-candidate-line parameters of the coarse-optimization stage as follows,

$$Y_0 = Y_e \times S, \tag{3.3}$$

$$\theta_0 = \theta_e, \tag{3.4}$$

where Y_e and θ_e are the elite-candidate-line parameters of the coarse-optimization stage, and S is a scale used to downsample an input image into a low-resolution image.

Designing of the Fitness Function

In a GA, a fitness function is used to guide the simulation toward an optimal solution, and it evaluates the goodness of each individual. Therefore, designing the fitness function is very important for quick convergence on an appropriate solution, and it has a significant impact on computational time. The fitness function should precisely evaluate how to fit a given solution and should be fast to compute. Existing optimization-based[47, 48] methods for horizon detection use an optimization criterion, which calculates the color distribution across all pixels of an image. Thus, these methods require a significant amount of time to achieve an accurate detection of the horizon line. In our previous work [51], we determined the local feature of the horizon line, which was used in the fitness function. As a result, the processing speed and accuracy of horizon detection was improved. However, the method in [51] failed in certain scenarios, such as changes in the color of the sea and the mostly occluded horizon line. To improve the accuracy in the above scenarios, we assumed a global feature factor in the fitness function. As mentioned before, the global feature estimation for each candidate is time-consuming because it covers a wide area of the input image. Thus, we created a probability map of the horizon line in the pre-processing stage and used it as a global feature factor in the fitness function. In the coarse step, the fitness function F was designed with the global feature factor G and local feature factor L as follows,

$$F^c(Y, \theta) = G(Y) \times L(Y, \theta). \tag{3.5}$$

In the fine-tuning step, the global feature factor effects were weaker than those of the local feature factor for the optimization because the position of the horizon line was roughly determined in the coarse-optimization. Therefore, we assumed only the local feature factor in the fine-tuning step into the fitness function as follows,

$$F^f(Y, \theta) = L(Y, \theta). \tag{3.6}$$

3.3.3 Global Feature

In the pre-processing stage, an input image is filtered by the Gaussian filter and downsampled. The downsampled image was used for global feature estimation and coarse stage optimization. Global feature estimation covers all pixels of image, but it is computationally expensive. Thus, we create a probability map of the horizon, which indicates the probability of the horizon at each row of an image. The map was used as a global feature factor in the fitness function. A textural feature and a color feature were used to extract the region that contains the horizon [34, 33]. Although both features are significant in extraction process for input images without blur, the textural feature is not applicable, where the color feature performs better. We used a color feature to determine the probability of existence of the horizon, similar to [33]. The creation steps of the probability map of the horizon are shown in Fig. 3.4. First, the image were divided into nine regions ($I = 9$). The height of these regions h was a fifth of the image height H, and 50 percent of the regions overlapped with neighboring regions as shown in Fig. 3.4(b). The region with larger change in color distribution compared with neighboring regions has a higher probability of

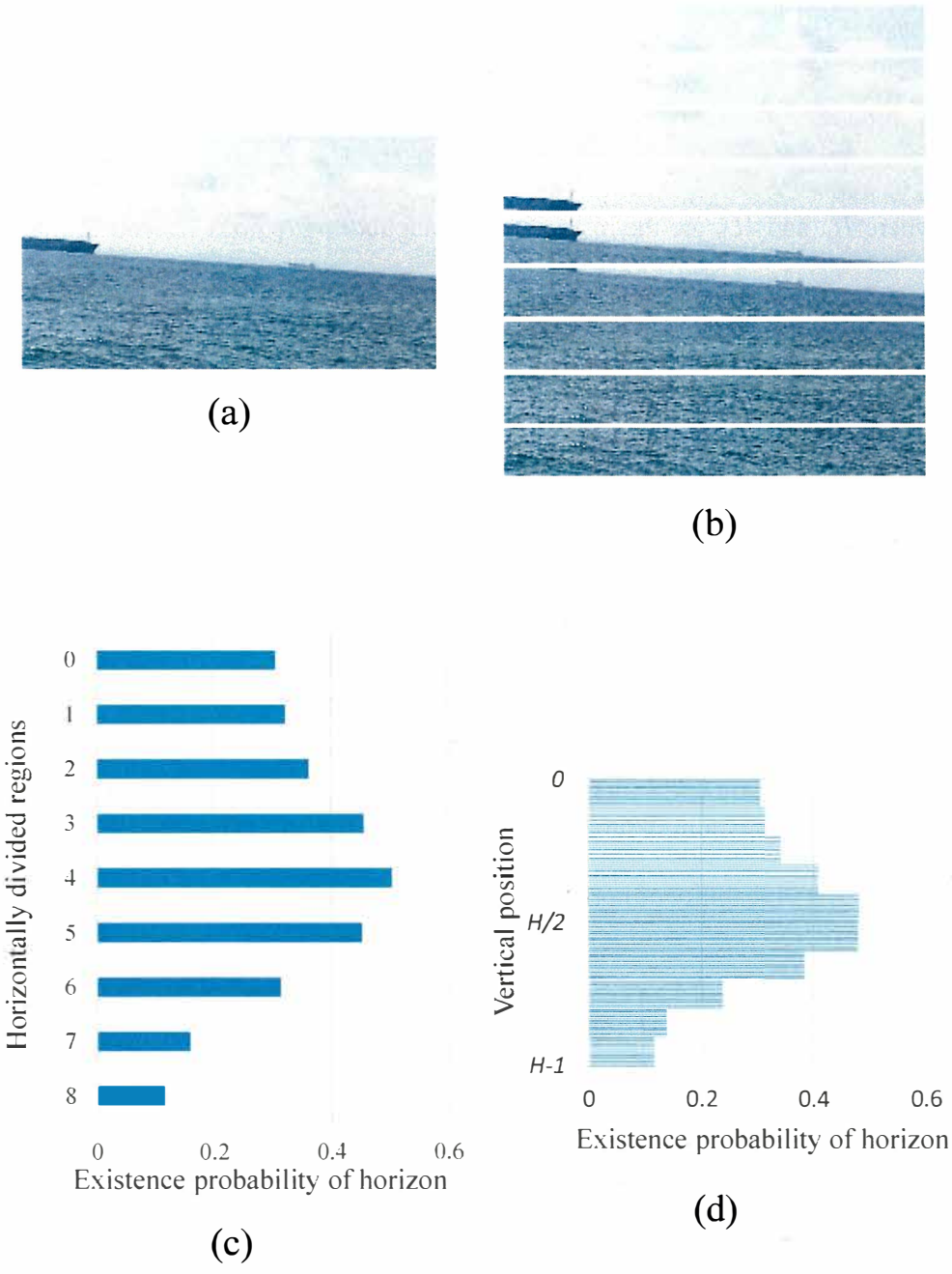


Figure 3.4: Creation steps of the probability map of horizon. (a) Downsampled image, (b) divided regions, (c) existence probability of horizon for each region, (d) probability map of horizon.

containing the horizon. A color histogram was calculated for each region to evaluate the color distribution due to processing speed considerations, and $N = 64$ bins were used for each color. To compare the histograms of two regions, the Hellinger

distance was calculated as follows,

$$D(R_n, R_m) = \sqrt{1 - \frac{\sum_{j=0}^{N-1} \sqrt{H_n(j) \times H_m(j)}}{\sqrt{\sum_{j=0}^{N-1} H_n(j) \times \sum_{j=0}^{N-1} H_m(j)}}}, \tag{3.7}$$

where H_n and H_m are the histograms of neighboring regions R_n and R_m, respectively. $H(j)$ indicates the j-th bin of the histogram. Then, as shown in Fig. 3.4(c), the existence probability of horizon was calculated for each region as follows,

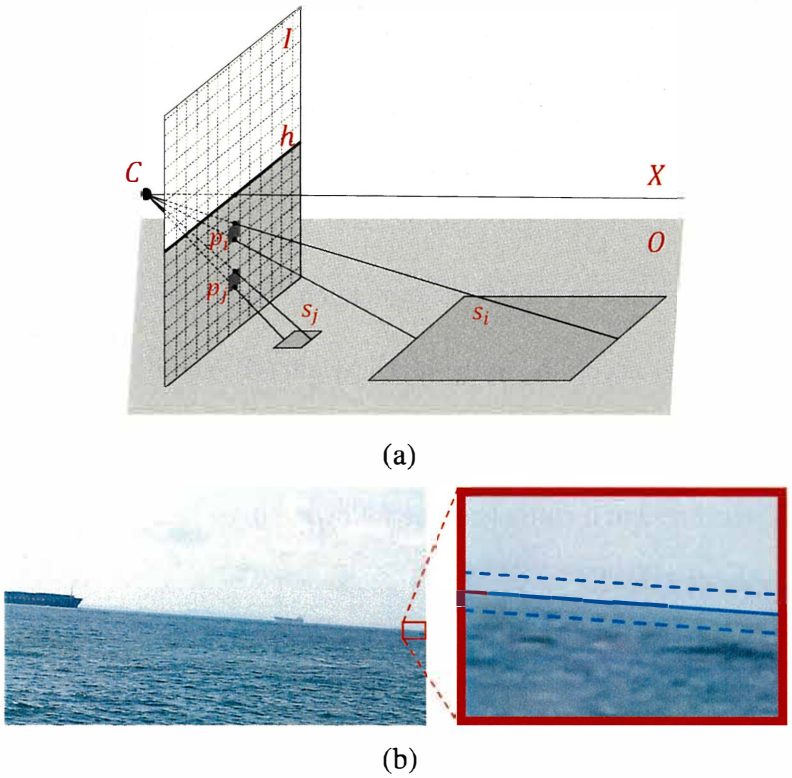
$$P(i) = \begin{cases} D(R_i, R_{i+1}) & \text{if } i = 0 \\ D(R_i, R_{i-1}) & \text{if } i = I - 1 \\ \frac{D(R_i, R_{i-1}) + D(R_i, R_{i+1})}{2} & \text{otherwise} \end{cases}, \tag{3.8}$$

where I is the number of the divided regions and nine in this research. Subsequently, a probability map of horizon was created for each row of the image as shown in Fig. 3.4(d). The probability of horizon at Y-th row of the image is calculated as follows,

$$G(Y) = \begin{cases} P(0) & \text{if } Y < h/2 \\ P(I-1) & \text{if } Y > (H - h/2) \\ \frac{P(\lfloor \frac{Y}{(h/2)} \rfloor - 1) + P(\lfloor \frac{Y}{(h/2)} \rfloor)}{2} & \text{otherwise} \end{cases}. \tag{3.9}$$

3.3.4 Local Features

For the fast evaluation of a candidate line, we defined the local features of the horizon line using a vanishing line characteristic. As shown in Fig. 3.5(a), we propose three essential characteristics. First, the horizon line is a straight line. Second, an



(a)

(b)

Figure 3.5: (a) Description of the horizon line in image plane, where bold line h indicates horizon line, the line X is line which intersects with h on the image plane I passes through the camera center C and is parallel to the sea surface plane O, the p_j and p_i pixels of image, the s_j and s_i are regions of sea surface those are projected into p_j and p_i respectively. (b) Local features of the horizon line in the sample frame and can be estimated in a narrow range close to the horizon.

appearance of the above horizon line is different from that of the side below the horizon line. Third, regions close to the horizon line tend to be texture-less. The reason is that the wide area of the sea surface is projected to a few pixels near the horizontal. Thus, the regions, which are close to the horizon line, have texture-less. As shown in Fig. 3.5(b), these characteristics can be estimated in the narrow regions close to the horizon line as local features. Reducing the area of the region, which evaluates the candidate, can reduce the processing time of the optimization process. The same local feature estimation is used in both the coarse and fine-step optimizations. In addition, the local features are estimated on a gray-scale image

using the following equation:

$$L(Y,\theta) = \frac{1}{N_c \times W_{\max}} \times \sum_{j=1}^{Z} \sum_{i=0}^{K} C(Y,\theta,j,x_i), \tag{3.10}$$

where Z is a parameter to control the evaluation range of the local feature, K is the number of samples according to the image width (W) and it is $K = W/d$. d is the sampling step and x_i follows $x_i = i \times d$. C is a function to estimate the local feature of the horizon line and it consists of the following three functions:

$$\begin{aligned} C(Y,\theta,j,x_i) = AB(Y,\theta,j,x_i) + A(Y,\theta,j,x_i) \\ + B(Y,\theta,j,x_i), \end{aligned} \tag{3.11}$$

where AB is a function that estimates the difference in appearance above and below the horizon area. A and B are functions that estimate the texturelessness of the above and below-side regions of the horizon line, respectively. To estimate the local features at given Y, θ, j, and x_i, four pixels ($S1, S2, M1$, and $M2$) were assumed, as shown in Fig. 3.6. The $S1$ and $M1$ points are symmetric with respect to the candidate line and symmetric to point $O1$. The $S2$ and $M2$ points are symmetric with respect to the candidate line and symmetric to point $O2$. $O1$ and $O2$ are points on the candidate line at a given x_i and $x_i - d/2$, respectively. When the four pixels are in the image plane, the function C is calculated using Eq. (3.11) otherwise C is 0. N_c is the total number of combinations of Y, θ, j, and x_i, when the four pixels are in the image plane. A W_{max} is the total value of the weights of the features used

in functions AB, A, and B. Function AB, A, and B are as follows:

$$AB(Y,\theta,j,x_i) = \begin{cases} w_1 & \text{if } |I_{S1} - I_{M1}| > T \\ w_2 & \text{otherwise} \end{cases}, \tag{3.12}$$

$$A(Y,\theta,j,x_i) = \begin{cases} w_3 & \text{if } |I_{S1} - I_{S2}| < T \\ 0 & \text{otherwise} \end{cases}, \tag{3.13}$$

$$B(Y,\theta,j,x_i) = \begin{cases} w_4 & \text{if } |I_{M1} - I_{M2}| < T \\ 0 & \text{otherwise} \end{cases}. \tag{3.14}$$

Here, AB counts symmetric points with respect to the candidate line with a different color. Functions A and B count the points that have similar neighboring points along with the candidate line. I_{S1}, I_{S2}, I_{M1}, and I_{M2} are the pixel values at $S1$, $S2$, $M1$ and $M2$, respectively. As shown in Fig. 3.6, the distance from the points $S1$, $M1$, $S2$, and $M2$ into the candidate line is j. A threshold value of T is used to evaluate whether the points are similar or different. w_1, w_2, w_3, and w_4 are the weights of the features.

3.4 Experimental Results

3.4.1 Dataset and Evaluation Criteria

We verified the performance of the proposed method using the Singapore Maritime Dataset (SMD) and Buoy-Dataset (BD) because which are publicly available and

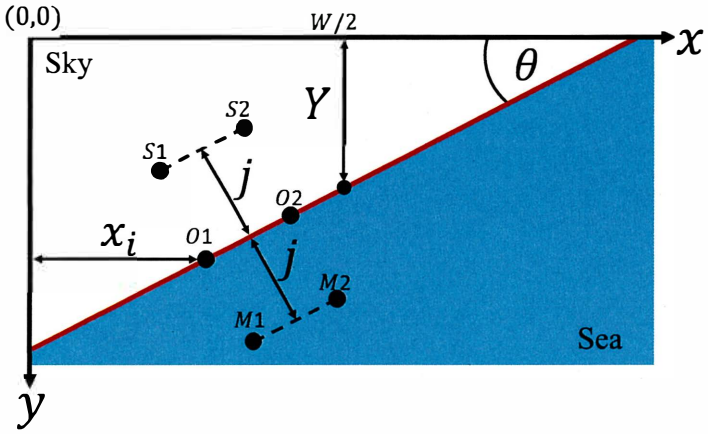


Figure 3.6: Local feature estimation. Four pixels are used for local feature estimation at given Y, θ, j, and x_i. These are $S1$, $S2$, $M1$, and $M2$. The red line describes a candidate line given by an orientation angle θ and a vertical position Y.

mostly used in recent researches[32, 33, 34]. The SMD consists of onboard and onshore videos. The onboard videos were captured by a camera mounted on a moving board, and the onshore videos are captured by a static camera installed onshore. The videos contain complex maritime scenes that have strong noise caused by wakes and waves, and color changes in the sea. The resolution of the SMD videos was 1920×1080 pixels. The BD consists of videos captured by a camera mounted on a floating buoy with resolution of 800×600 pixels. A challenge for the onboard videos of SMD and BD is the large variation in the orientation and position of the horizon line between adjacent frames. The details of SMD and BD are given in Table 4.3. The ground truth of the horizon line is given by the vertical position Y and the orientation angle θ.

In previous studies [32, 33, 34, 43, 38], the performance is commonly evaluated by mean absolute error (MAE) and percentile error at 25th, 50th, and 95th. The 95th percentile error indicates the detection result on complex scenes and it is used to imply that the how detection method is robust and consistent over datasets with

Table 3.1: Details of datasets.[32]

Dataset	Buoy	Singapore maritime	
		Onboard	Onshore
No. of videos	10	11	28
No. of frames	996	2772	12604
Min(Y-mean(Y))	-281.68	-436.30	-13.54
Max(Y-mean(Y))	307.82	467.86	9.95
Standard deviation of Y	107.98	145.10	1.52
Min(θ-mean(θ))	-15.72	-26.34	-9.99
Max(θ-mean(θ))	20.72	12.99	0.51
Standard deviation of θ	4.40	1.11	0.04

great diversity [32, 33]. To compare our results, our study used the same percentile errors. Moreover, MAE was used for analyzing parameter analysis.

3.4.2 Parameter Setting

The proposed method needs to adjust the values of the parameters. Because they affect the performance, it is necessary to investigate the optimal values of parameters. However, the number of parameters is very large. Hence, we focused on population size, generation size, and threshold of the local features (T) because these have a significant influence on performance. The experimental results are described in Sect. 3.4.3. The other parameters were empirically fixed. The details are described as follows. The population and generation sizes of the GA are 20 and 20 in both coarse and fine optimization stages. The crossover and mutation probabilities are 0.6 and 0.07, respectively, which affect the convergence speed and diversity of the population. The orientation adjustment parameter ρ and height adjustment parameter λ are decoded from the chromosome. In the coarse-optimization stage, the adjustment ranges of each parameters are described as follows,

- Orientation adjustment ρ: $[-\pi/4, \pi/4]$,
- Height adjustment λ: $[-3H/5, 3H/5]$.

To detect the horizon line that is particularly out of an image, we set the range of height adjustment by a value that larger than the height of an image. In coarse optimization, the parameters were roughly optimized. Hence, in the fine-step optimization, the orientation and height adjustment ranges exhibited a reduction of six times and two times, respectively, with the ranges of coarse optimization. The chromosome length has 16 bits because each parameter is represented by eight bits in both optimization stages. We downsampled the input image for global feature estimation and coarse optimization. The scale to downsample was $S = 1/4$ for the BD and $S = 1/8$ for the SMD. For quick local features estimation, the range parameter to evaluate the local features was $Z = 6$ and the sampling step was $d = 4$. In addition, the threshold value was $T = 20$, and the weights of the local feature were $w_1 = 3$, $w_2 = -2$, $w_3 = 1$, and $w_4 = 2$.

3.4.3 Effectiveness of EVP and Parameter Analysis

As described in Sect. 3.3.2, in our coarse-to-fine evolutionary method, the EVP is used for the optimization of HL parameters. In this section, to confirm the effectiveness of EVP in this study, its performance was compared with the simple GA (see Sect. 3.3.2) and exhaustive search (ES) on the SMD. The same fitness function was used for the EVP and the simple GA, also the same function was used as an evaluation function in the ES. In the experiment of this paper, the same size of candidates was used in both coarse and fine optimization stages. In the EVP and simple

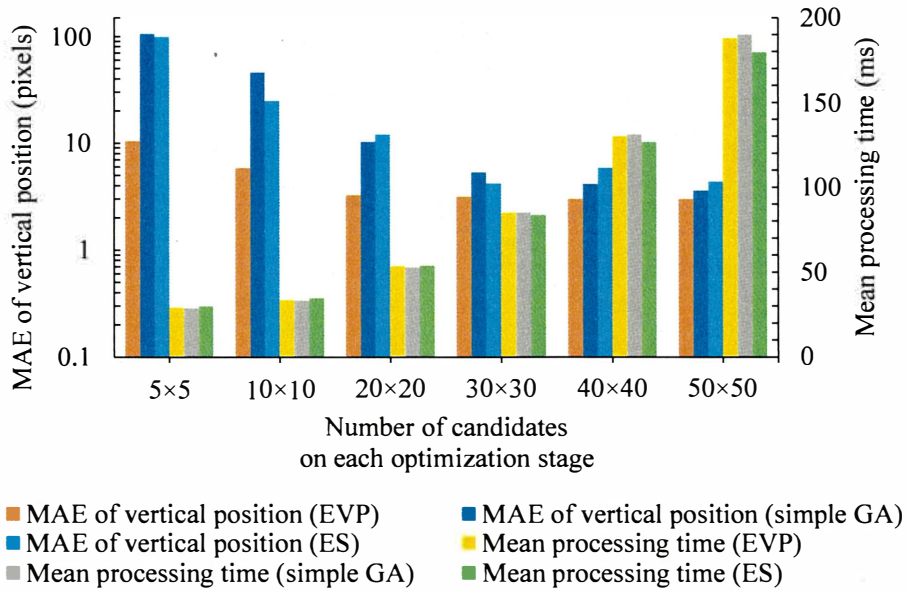


Figure 3.7: Accuracy and processing times (EVP, simple GA, and ES), which were performed by fixing the threshold value of the local features ($T = 20$) and changing the number of candidates on SMD.

GA, the population and generation sizes were the same, 5, 10, 20, 30, 40, and 50. In the ES, we uniformly sampled the search space of two parameters (Y and θ) for HL and the sampling resolutions were similarly 5, 10, 20, 30, 40, and 50.

Figure 3.7 shows the MAE of vertical position and average processing time per frame on various numbers of candidates. The MAE of EVP was smaller than the simple GA and ES in all the candidate sizes. Moreover, the processing times of all methods were almost the same. Therefore, the EVP is effective for HL detection on the video. Although the number of candidates on each stage increased from 20×20, the accuracy of the EVP increased a little. Considering fast detection of HL, 20 and 20 for the population and generation sizes were optimal combination for the proposed method. Hence, these values were used for comparison experiments with related work.

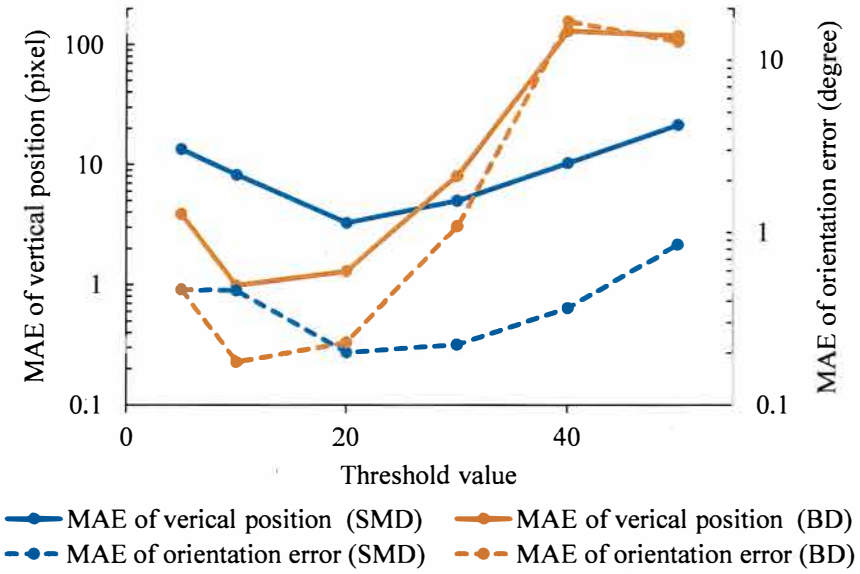


Figure 3.8: Accuracy of the proposed method, which was performed on two datasets, SMD and BD by fixing the generation and population sizes (set as 20) and changing the threshold value of the local features.

To find the optimal threshold value of the local features (T), various values were set to the proposed method to compare. The results are shown in Fig. 3.8. The lowest MAEs on BD and SMD were $T = 10$ and $T = 20$, respectively. However, while comparing our method with related works, we used the same threshold value $T = 20$ on all datasets like Ganbold et al.[51].

3.4.4 Comparison Results and Consideration

First, we compared the variation of proposed methods to clarify the effectiveness of the improvements. A variation one (V1) is a method that is not used in the global feature (GF) estimation and coarse-to-fine approach (CF). A variation two (V2) is a method that is not used in the global feature estimation [51]. Also, the method introduced in section 3.3 (CFEM) used all improvements. To compare the detection

result of V1 with other variations, we set the parameter values on the V1 method as follows. The height range of the position parameter is the same as the input image height. The orientation adjustment range is $[-\pi/4, \pi/4]$, which is also the conceivable range. The chromosome length per parameter is 10 bits. The population and generation sizes are 80 and 80. Except the above parameters are the same as parameter values in section 3.4.2. To compare the detection result of V2, we used the parameter values, which is introduced in [51].

Second, the performance of proposed method was compared with the state-of-the-art HL detection methods, which were compared in [33]. For fair comparison, the same datasets and same evaluation criteria are used in this study. Jeong et al.[33] compared their proposed method on the SMD and BD with state-of-art methods including MusCoWERT[32], MSCM-Life[49], the method of Fefilatyev et al. (FGSL)[54], the method of Lipschutz et al. (LHSL)[48] and two methods on [44] those are the Hough method (Hough) and the intensity variation analysis based method (IntV).

Hough [44] method applied the median filter to smooth the input image. Then, it extracted the edge image using the Canny edge detector. Then, the Hough transform was used to find the optimal parameter of horizon line from the edge map.

IntV [44] method first performed the median filtering. Then, it searched the points having the maximum edge magnitude in each column of the smoothed image. Then, the least-square method was applied to maximal edges to find the optimal line.

FGSL [54] method selected N candidate horizon lines by performing the Hough transform on the edge image. Gaussian distribution was then used to model the color information of two regions separated by the candidate line. The similarity

between the color distribution of the sea and the sky regions was measured using the Bhattacharyya distance. The candidate horizon line having the lowest similarity was determined to be the final solution. The number of candidate lines greatly affected the processing speed of the system. Thus, 10 candidate lines were selected for the computational efficiency.

LHSL [48] method also selected N candidate horizon lines, like the FGSL method. However, the different color model was used to model the color distribution of the regions. The optimal line was then selected by calculating the distance between the two histograms. In the experiment, the number of candidate lines equal to the FGSL method was selected.

MSCM-LiFe [49] 10 vertical median filters having different sizes were used to generate the multi-scale image. For each scale image, Canny edge detection and the Hough transform were used to find the 10 candidate horizon lines. For each scale, the mean multi-scale images were generated by accumulating the scale image. Then, the method found the points having the maximum intensity variation in each column of the mean multi-scale images. The optimal horizon line was then selected by

In MuSCoWERT [32], the multi-scale image was generated by applying median filters of various sizes. An edge detection was performed to generate the edge map for each scale image. Then, lengths were calculated for all edges to consider their weight. After that, the Radon transform was applied to each weighted edge map to select N candidate horizon parameters. The final horizon line was then determined by voting on the candidate horizon parameters of each scale.

Jeong et al. [33] vertically divided an image into several regions and extracted

the sky-sea region using the difference between the color distributions of the consecutive regions. The difference between the two regions was calculated by the Bhattacharya distance, and a region with the largest distance was selected as the sky-sea region. Then, multi-scale edge detection was applied to the sky-sea region and merged into one edge map. Finally, the Hough and least-squares methods were sequentially used to find the horizon.

The statistics of the errors in parameters Y and θ on SMD are listed in Tables 3.2 and 3.3. The statistics of the errors in parameters Y and θ on BD are listed in Table 3.4. The experimental results show that the proposed method performs better on the SMD and BD datasets. In particular, the median positional error and median orientation error of the proposed method were relatively smaller than those of all the compared methods in all datasets. In addition, the proposed method can detect the horizon line when the input image is blurred, as shown in the bottom image of Fig. 3.13. In the 95th percentile, the orientation angle error of [33] was smaller than that of the proposed method on the SMD, but the position error of the proposed method was smaller than that of the other methods.

Comparison of proposed method variations demonstrates that the method with coarse-to-fine has high accuracy and high processing speed even in high-resolution images. In low-resolution images, the effectiveness of coarse-to-fine was low compared with the result on high-resolution images. The global feature estimation in the fitness function has increased the accuracy compared with methods without it. As shown in Fig. 3.9, the utilization of global features in optimization criteria had the effect of reducing false positives caused by changes in sea color. Although the vertical position error decreased, orientation angle error increased when the horizon

(a) (b)

Figure 3.9: Comparison of the detection results. (a) Result of the V2 [51]. (b) Result of proposed method.

line was mostly occluded by objects, as shown in the bottom row images in Fig. 3.9.

Average processing times per image by our and comparative methods are provided in Table A.3. The results of the Jeong ROI[33], MSCM-Life[49], FGSL[54], LHSL [48], Hough[44], IntV[44] were taken from [33]. They were implemented using Phyton and executed on an Intel E5-1680 CPU. The MusCoWERT was taken from [32] and it was implemented using MATLAB 2015b, and the result was obtained on an Intel i7-3770 CPU. The proposed method and method of the V2 [51] were implemented using C/C++ and were executed on an Intel i7-3770 CPU. The MusCoWERT, MSCM-Life, FGSL, and LHSL were required several seconds per image. The methods of IntV and Hough were relatively fast, but they were low detection accuracy. The methods of Jeong ROI[33], the V2 [51], and the proposed method processed the image within one-tenth seconds and they were reliable detection accuracy.

The experimental results demonstrate that the proposed method can detect the

Table 3.2: Comparision of horizon detection on onboard videos from the SMD

Vertical position error (pixels)			
	25th percentile	50th percentile	95th percentile
Jeong ROI[33]	**0.51**	1.23	3.99
MusCoWERT[32]	0.54	1.49	8.17
MSCM-Life[49]	1.16	2.84	505.78
LHSL[48]	13.78	25.65	507.92
FGSL[33]	5.28	10.85	581.44
IntV[44]	13.36	24.89	498.17
Hough[44]	2.27	221.67	520.34
Ours (V1)	213.11	397.49	660.17
Ours (V2) [51]	0.58	1.26	4.74
Ours (CFEM)	**0.51**	**1.09**	**3.83**
Orientation angle error (degrees)			
	25th percentile	50th percentile	95th percentile
Jeong ROI[33]	**0.05**	0.12	**0.39**
MusCoWERT[32]	0.06	0.25	0.88
MSCM-Life[49]	0.17	0.38	5.50
LHSL[48]	0.88	1.37	6.52
FGSL[33]	0.67	1.00	3.88
IntV[44]	0.87	1.35	6.12
Hough[44]	0.25	1.00	4.57
Ours (V1)	0.50	2.11	9.48
Ours (V2) [51]	0.06	0.14	0.55
Ours (CFEM)	**0.05**	**0.11**	0.44

HL at high speed with high accuracy, is effective for HL detection on high resolution video data. Example frames of the horizon line detection results for SMD and BD are shown in Fig. 3.10, 3.12, 3.11 which indicates that the proposed method accurately detected the horizon line in various maritime scenes. The detection results for complex scenarios are shown in Fig. 3.13. The proposed method failed for data in which the horizon line was mostly occluded by objects.

Table 3.3: Comparision of horizon detection on onshore videos from the SMD

Vertical position error (pixels)			
	25th percentile	50th percentile	95th percentile
Jeong ROI[33]	0.99	2.09	12.87
MusCoWERT[32]	1.14	2.63	11.41
MSCM-Life[49]	1.63	3.88	81.59
LHSL[48]	14.96	27.92	109.00
FGSL[33]	5.88	11.53	64.70
IntV[44]	2.08	5.82	39.89
Hough[44]	3.12	165.02	460.24
Ours (V1)	1.43	4.32	573.72
Ours (V2) [51]	0.83	**1.94**	32.72
Ours (CFEM)	**0.77**	1.95	**11.00**
Orientation angle error (degrees)			
	25th percentile	50th percentile	95th percentile
Jeong ROI[33]	0.04	0.10	0.67
MusCoWERT[32]	0.14	0.21	1.07
MSCM-Life[49]	0.11	0.18	1.14
LHSL[48]	0.75	1.03	3.86
FGSL[33]	0.75	1.00	2.87
IntV[44]	0.14	0.52	5.37
Hough[44]	0.14	0.36	3.80
Ours (V1)	0.12	0.27	4.88
Ours (V2) [51]	**0.03**	0.09	**0.44**
Ours (CFEM)	**0.03**	**0.07**	0.74

3.5 Conclusion

In this study, we proposed a novel fast horizon line detection method that optimizes the horizon parameters by using a GA. We also adopted a coarse-to-fine approach to meet real-time processing requirements. In addition, we introduced a fast estimation of global and local feature estimations for quick optimization. Previous methods extracted the candidates of horizon using the edge information and use consecutive filtering to find the final solution. A limitation of these methods is that

Table 3.4: Comparision of horizon detection on videos from the Buoy dataset

Vertical position error (pixels)			
	25th percentile	50th percentile	95th percentile
Jeong ROI[33]	0.53	1.07	2.98
MSCM-Life[49]	1.54	2.97	11.56
LHSL[48]	0.66	1.50	3.76
FGSL[33]	0.60	1.35	3.84
IntV[44]	0.84	1.91	55.06
Hough[44]	0.77	1.76	4.46
Ours (V1)	0.46	1.01	2.93
Ours (V2) [51]	0.44	0.94	2.74
Ours (CFEM)	**0.37**	**0.80**	**2.29**
Orientation angle error (degrees)			
	25th percentile	50th percentile	95th percentile
Jeong ROI[33]	0.07	0.15	0.45
MSCM-Life[49]	0.33	0.57	11.56
LHSL[48]	0.17	0.33	0.67
FGSL[33]	0.18	0.36	0.79
IntV[44]	0.14	0.32	13.24
Hough[44]	0.18	0.37	0.89
Ours (V1)	0.09	0.19	0.54
Ours (V2) [51]	0.08	0.17	0.59
Ours (CFEM)	**0.06**	**0.14**	**0.44**

Table 3.5: Average processing time per frame in seconds

	Onboard	Onshore	Buoy
Jeong ROI[33]	0.07	0.07	0.02
MusCoWERT[32]	9.2	9.5	5.8
MSCM-Life[49]	6.73	6.83	2.26
LHSL[48]	13.75	13.76	2.30
FGSL[33]	36.58	36.63	8.61
IntV[44]	0.30	0.30	0.01
Hough[44]	0.11	0.10	0.01
Ours (V1)	0.11	0.29	0.29
Ours (V2) [51]	0.04	0.04	0.02
Ours (CFEM)	0.05	0.05	0.02

if the candidates cannot be extracted from edge information in the previous stage, they are not considered in the next stage, even though these candidates are survival candidates in the filtering of the next stages. Unlike these methods, our method is a heuristic optimization-based method and local and global features are concurrently utilized to evaluate each candidate. The proposed method does not extract edge information from multi-scale images, and even for blurred input images, it can detect the horizon line.

The proposed method was tested on the SMD and BD, which are publicly available datasets that contain complex maritime scenes. In addition, we compared the performance of the proposed method with that state-of-the-art methods, which used the same datasets. The experimental results indicated that the proposed method could detect the horizon line more accurately than the compared methods. In particular, the median positional error and median orientation error of the proposed method were relatively smaller than those of all the compared methods in all datasets. The processing speed of our method was approximately $20 fps$ for high-resolution images. However, the proposed method failed in scenarios in which the horizon line was mostly occluded by objects as shown at the top of Fig. 3.13. The mostly occluded case is out of range of the proposed method. In future work, we plan to explore other optimization criteria and features for robust detection of HL in complex scenarios such as the horizon is mostly occluded and the coastal line is visible.

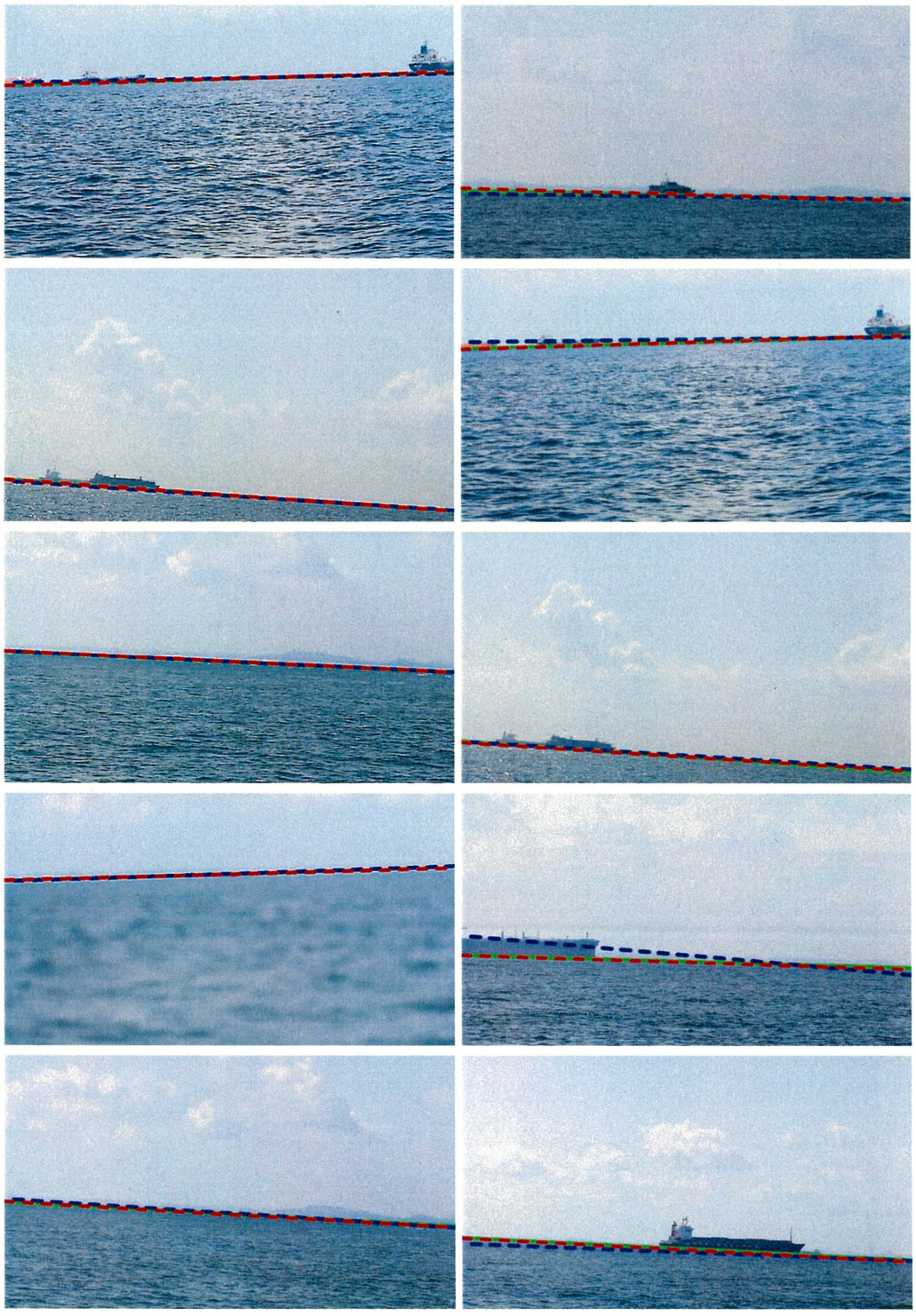


Figure 3.10: Sample frames of detection results from the onboard dataset of SMD. The red and blue dashed lines indicate the detection result of the horizon line and the coarse-step optimization result, respectively, and the green line indicates the ground truth.

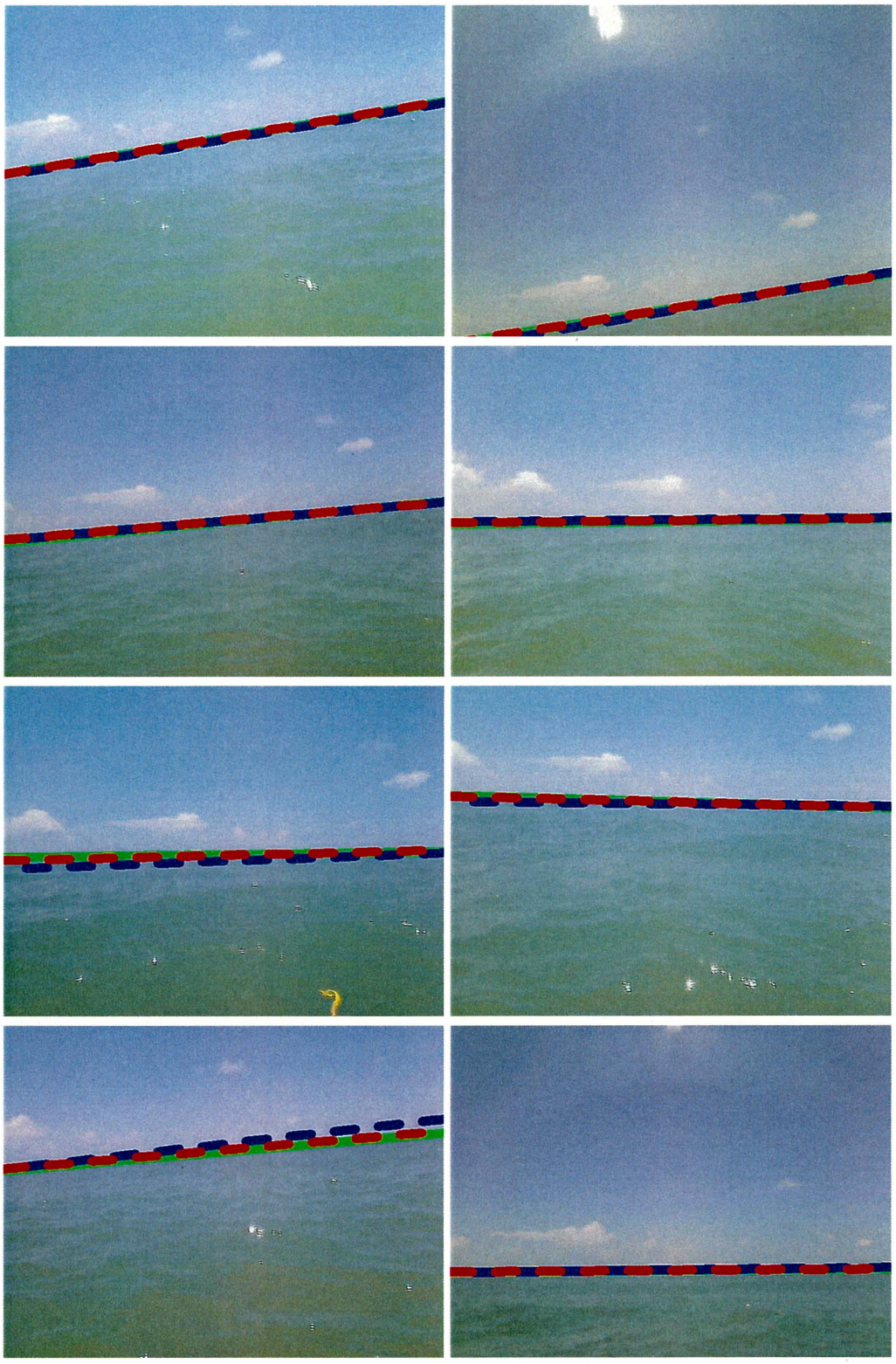


Figure 3.11: Sample frames of detection results from Bouy dataset. The red and blue dashed lines indicate the detection result of the horizon line and the coarse-step optimization result, respectively, and the green line indicates the ground truth.

Figure 3.12: Sample frames of detection results the onshore dataset of SMD. (c) Detection results on BD. The red and blue dashed lines indicate the detection result of the horizon line and the coarse-step optimization result, respectively, and the green line indicates the ground truth.

Figure 3.13: Detection results on complex scenarios, with occlusion of the horizon and blurred images. The red and blue dashed lines indicate the detection result of the horizon line and the coarse-step optimization result, respectively, and the green line indicates the ground truth.

Chapter 4

Possibility of practical application

In this chapter, some practical applications of the proposed method are introduced. In previous studies, deterministic methods such as Hough and Radon transform-based methods are used to detect line detection. In chapter 3, we introduced the evolutionary method which metaheuristic method to detect the horizon line. As a result, the effectiveness of the evolutionary method is confirmed. Therefore, we tested the proposed method for two applications of borderline (single line) detection. First, we applied for detection of the horizon line in road scenarios images. Like the proposed method in chapter 3, GA is utilised for optimization of line parameters, and local features estimation is used for the objective function. The local features are derived from vanishing line characteristics. The section 4.1 introduces the methods in detail. Second, we proposed the evolutionary method for finding the boundary line for mower. The boundary line is a single line in the image plane which separates the image into the cut grass and the uncut grass regions. In the section 4.2, the method is introduced in detail. As a result of tests, the evolutionary method is useful in borderline line detection applications.

4.1 Horizon line detection in road images using genetic algorithm

In this section, I explain a fast horizon detection in road images using genetic algorithm. In mobile robots and autonomous vehicles, the information of depth and camera's orientation are necessary for the applications of positioning and navigation. We propose a horizon line (HL) detection method on the road image captured by a single fish-eye camera. The fish-eye camera can capture a broader scope of environmental information than camera with narrow angle view. It is important to detect the objects by one camera for advanced driver assistance systems. The HL can provide the important information about depth and camera's orientation for the single view estimation. The detection of HL is assumed as an optimization problem. The optimization of the line parameters is performed by the genetic algorithm (GA). We implement the real-time detection method of HL using GA. There has fisheye dataset for autonomous driving [55] but there has not ground truth of horizon line of the videos. Therefore, we created by fish-eye camera with the various environment, and we tested the proposed method on this dataset. Experimental results demonstrate the efficiency of the proposed method.

4.1.1 Introduction

Autonomous vehicles (AVs) and advanced driver assistance systems (ADASs) have become the active research area aimed at improving traffic safety. In recent years, several techniques estimating depth and camera's orientation have been developed using the single image and stream observed with a monocular camera. In the last

two decades, the estimation techniques of relative camera pose and 3D structure from monocular images, have been widely developed.

The horizon line (HL) is defined by the Vanishing Line (VL) of ground/road plane. According to the suggestion of Gibson[56] and Hartley et al.[57], the vanishing line on the image plane can provide important information about depth and camera's orientation for single view estimation. a Okada et al.[58] and Bertozzi et al.[59] used the static HL (HL is fixed) for filtering to decrease the number of false positives. Unfortunately, each movement of the vehicle causes the variation of camera orientation and estimation of depth. Therefore, horizon line detection is needed. Common detection approaches of the HL and VL use the voting procedure on the extracted edge line segments to find points with many intersects[60, 61]. Fefilatyev et al.[54] present a method for detecting the VL by looking for similar pairs from scale-invariant feature transform (SIFT) keypoints. But this method focused on the only stochastically-textured plane such as ocean surface. Ahmad et al.[62] proposed the HL detection method, which is Local Features based method with machine learning. First, Maximally Stable Extremal Edges (MSEE) are extracted. Then SIFT is computed on MSEE. Finally, SVM classifier is used to classify MSSE pixels are into horizon or non-horizon. Limitation of this work is training data (which is needed for the new scene) and the horizon is expressed by boundary points (too flexible).

In this study, we propose the appearance-assumption-based method on the image to detect the HL. It is necessary to search the HL within the wide range since the camera mounted on a vehicle has oscillation. The exhaustive search is expensive in processing time. In order to solve this problem efficiently, this study exploit GA as

the optimization technique for the HL detection. Our proposed method has the following advantages. First, it can detect the HL on the image taken by an uncalibrated fish-eye camera. Because an uncalibrated image has high radial distortion effect. To minimize the radial distortion effect error, the HL is encoded into two linked lines in this research. Second, the detection algorithm with GA searches for the HL on input image. Thus, the preprocessing stages such as edge detection and clustering are not required. Third, our method can detect the HL from a single image. Moreover, our method is able to run in the real-time. For the reason that the proposed method uses the inheritance of genetic information from the detection result of the previous video frame. The proposed method is detailed described in the following sections. Experimental results are shown in Section 4.3.

4.1.2 Methodology

This section explains the HL detection method based on the appearance-assumption by using GA [14]. The main idea of this research is the appearance-assumptions of the HL is initially described. Then, the defined description is formulated into logical rules. This logical rule becomes the fitness function in GA. To become efficiency and robustness, we used the GA to find the best-fitted HL. After that, the most fitted HL for the formulated rule is directly searched from the input image. The HL has described not the single line, it is described by two linked lines in this research. The details of the methodology are explained by following paragraphs:

- Horizon Line in Road Images,
- HL Parameters and Structure Chromosome,

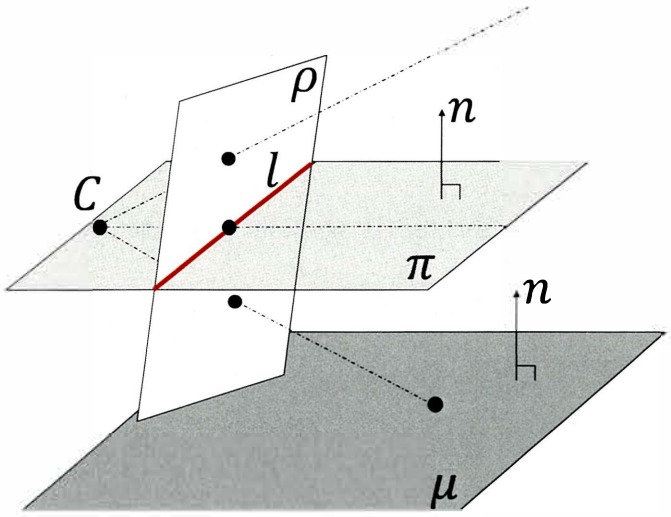
- Fitness Function.

Horizon Line in Road Images

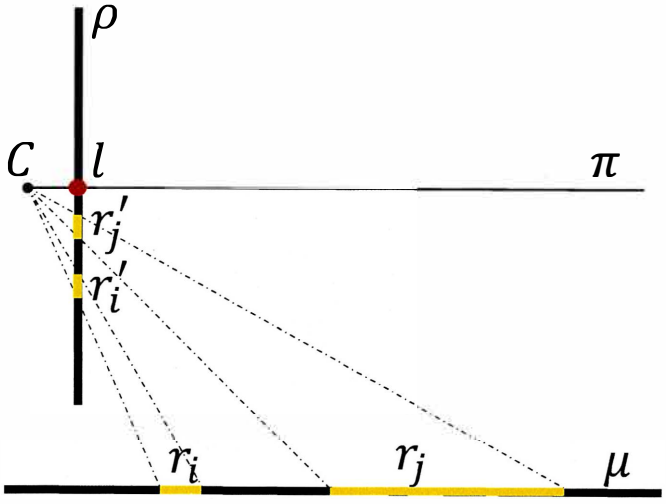
In this work, we considered that the HL also represents the vanishing line of the road surface on an image plane. In fact, the HL on the image plane is the vanishing line where the road surface vanishes and disappears on the image plane. If we assume that the road surface is relatively flat, all parallel planes to the sea surface plane intersect in a common line on the image plane. This common line considers the HL of the road surface plane [57]. The HL is geometrically illustrated in Fig. 4.1. From this understanding, we propose three essential characteristics. First, an appearance of above the HL is different from the appearance of below the HL. Second, regions which close to HL tend to be uniform textures. The reason is that the wide area of the sea surface is projected to a few pixels near the HL in the image illustrated in Fig. 4.1(b). Thus the regions which close to HL has texture-less. Third, the appearance of above the HL, for instance; obstacle and object, have anisotropic /uncommon/ characteristics.

HL Parameters and Structure Chromosome

As shown in Fig. 4.2, power lines are projected on image by curve due to distortion of the fish-eye camera. To minimize the distortion effect of the fish-eye camera, the HL is demonstrated by linked two lines at three points. The HL encoded into 4 parameters as a chromosome. These are the orientation of line φ, height adjustment of center point α, height adjustment of left point β, and height adjustment of right point δ. The GAs work with a strings of the parameter set, not the parameter



(a)



(b)

Figure 4.1: The formulation of a horizon line. The horizon line l is obtained at the intersection of the image plane ρ and the plane π which passes through the camera center C. The plane π is parallel to the ground surface plane μ. (b) The r_j, r_i are surface regions and the r_j', r_i' are projected regions of the $r_j > r_i$.

themselves. These chromosomes represent the line on the image. The length of chromosome affects the processing speed. Therefore, we minimized the length of the chromosome according to the range of parameters. Table 4.1 shows the range of each parameter. Eq. 4.1 describe the location of the center point. Eq. 4.2 and Eq. 4.3 formulate the location of the left and right points. In eqs.4.1-4.3, W and H

Table 4.1: List of each candidate line parameter's range

Parameter	Range	Length of gene	Step size
φ	$[-0.1\pi, 0.3\pi]$	5 *bit*	$0.2\pi/32$
α	$[-0.25H, 0.25H]$	8 *bit*	$0.5H/256$
β	$[-0.02H, 0.02H]$	4 *bit*	$0.04H/16$
δ	$[-0.02H, 0.02H]$	4 *bit*	$0.04H/16$

donate the width and height of the input image.

$$P_{center}: \quad (x_1, y_1) = (\frac{W}{2}, \frac{H}{2} + \alpha) \tag{4.1}$$

$$P_{left}: \quad (x_0, y_0) = (0, \frac{H}{2} + \alpha - \frac{W}{2} \times (\varphi) + \beta) \tag{4.2}$$

$$P_{right}: \quad (x_2, y_2) = (0, \frac{H}{2} + \alpha + \frac{W}{2} \times (\varphi) + \delta) \tag{4.3}$$

Fitness Function

The GA works on the maximization of fitness. The fitness value of a candidate F consists of three subfunctions in Eq. 4.4. These functions are inspired by the appearance-assumption of HL stated in section 6.1. Equations 4.5-4.7 formulate three sub-functions. Function TB calculates the number of the symmetric points if the pixel value difference of the two symmetric points with respect to the HL is greater than the threshold value TH. The TH is a threshold value to judge the similar or different. The TH is empirically fixed to 10. The function T calculates a point on the top side of HL when the pixel value difference between that point and neighbor point along with y-axis is greater than the TH. The function B calculates

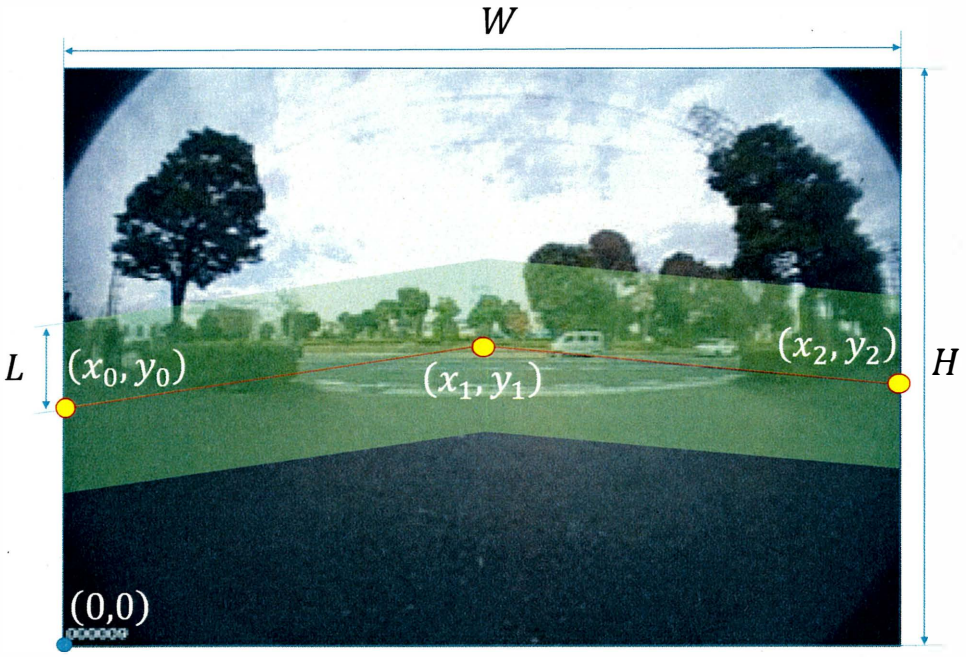


Figure 4.2: The description of the candidate line by red. (x_0, y_0),(x_1, y_1),(x_2, y_2) are coordinates of respectively the left, center and right. The (0,0) is the origin point and L is a parameter to control the evaluation range of the candidate. W and H denote the width and height of the input image.

a point on the bottom side of HL when the pixel value difference between that point and neighbor point along with y-axis is lower than the TH.

$$F = \frac{1}{3LW} \times \sum_{j=0}^{L} \sum_{x=0}^{W-1} (TB(j,x) + T(j,x) + B(j,x)), \tag{4.4}$$

$$TB(j,x) = \begin{cases} 1 & |P(x, Y(x)+j) - P(x, Y(x)-j)| > TH \\ 0 & otherwise \end{cases} \tag{4.5}$$

$$T(j,x) = \begin{cases} 1 & |P(x, Y(x)-j) - P(x+1, Y(x)-j)| > TH \\ 0 & otherwise \end{cases} \tag{4.6}$$

$$B(j,x) = \begin{cases} 1 & |P(x,Y(x)+j) - P(x+1,Y(x)+j)| < TH \\ 0 & otherwise \end{cases} \tag{4.7}$$

where L is a parameter to control the evaluation range when L is equal to 20. $Y(x)$ is given by Eq. 4.8.

$$Y(x) = \begin{cases} y_0 + \frac{y_1 - y_0}{x_1 - x_0} \times x & x < x_1 \\ y_1 + \frac{y_2 - y_1}{x_2 - x_1} \times (x - \frac{W}{2}) & otherwise \end{cases} \tag{4.8}$$

where (x_0, y_0),(x_1, y_1),(x_2, y_2) are coordinates of respectively the left, center and right points of the candidate. These coordinates are decoded from the chromosome of candidate.

4.1.3 Experimental Result

This section mentions the experimental results of the proposed method. The proposed method is tested on real traffic 10 sequences involving the moving objects of pedestrian, cars, and bicycles in different environments. The sequences 1-9 are captured when the car stops at the crossroad. The sequence 10 is captured when the car moves into the crossroad. The dataset is captured by the fish-eye camera mounted above 0.7m high from the ground panel in the car. The ground truths of HL and moving object are labeled manually per frame. Each sequence consists of 150 frames in 320 $\times$ 240 pixel resolution. The frame rate is 30fps. All the results are tested on the PC with CPU of Intel Core (TM) i7-3770 (3.40 GHz) and RAM of 8.0 GB. The details of the evaluation criteria and experimental results are explained

by following paragraphs.

Evaluation Criteria

To evaluate the detection of HL, we calculated absolute distance error and absolute orientation error. The absolute distance error is evaluated the following Eq. 4.9.

$$\tau_d = \sum_{x=0}^{W-1} |Y_x^{GT} - Y_x^{HL}| \tag{4.9}$$

where Y_x^{GT} is y coordinate of the point, it lies on the ground truth line at x coordinate. Y_x^{HL} is y coordinate of the point, it lies on the detected HL at x coordinate. The absolute orientation error is evaluated the following Eq. 4.10.

$$\tau_o = \frac{|\varphi^{GT} - \varphi^{HL}|}{2} \tag{4.10}$$

where φ^{GT} is the angle of ground truth line and φ^{HL} is the angle of detected HL, defined by Eq. 4.11.

$$\varphi^{HL} = \frac{tan^{-1}(\frac{y_1-y_0}{x_1-x_0}) + tan^{-1}(\frac{y_2-y_1}{x_2-x_1})}{2} \tag{4.11}$$

Where (x_0, y_0), (x_1, y_1), (x_2, y_2) are coordinates of respectively the left, center and right points of detected the HL.

Experimental Result of Horizon Line Detection

In order to prove the performance of the proposed HL detection method, while rotating the input frame, we evaluated the position and orientation of the HL. The

Table 4.2: Performance result of horizontal line estimation

Number of sequence	Average absolute error of distance (pixels)	Average absolute error of orientation (degrees)
Sequence 1	4.26	1.75
Sequence 2	3.86	1.85
Sequence 3	2.65	1.35
Sequence 4	3.81	0.86
Sequence 5	3.47	2.79
Sequence 6	5.14	2.98
Sequence 7	4.35	1.37
Sequence 8	2.94	0.70
Sequence 9	2.13	1.57
Sequence 10	5.49	1.61

angle of the ground truth line can assume as equal to the rotation angle of the input frame because the orientations of ground truth lines equal to 0 without any rotation. The rotation range is fixed to be $[-10°, 10°]$ and rotation rate is $1°$ every two frames. The estimation result of HL orientation is shown in Fig. 4.3. The estimated orientation was similar to the rotation angle of frame. Table4.2 shows the average absolute error of distance and average absolute error of orientation on 10 sequences. The average processing time of HL detection was 32.52ms for one frame under the following condition. The probabilities of crossover and mutation were 0.70 and 0.01. The number of individuals and generation iterations was 30 and 50.

4.1.4 Conclusion

In conclusion, the absolute error of position and absolute error of orientation are low and the difference between the absolute errors are stable. Our proposed method can be used to detect the horizon line in road images. In the future, we would like to improve the proposed method and compare with the state-of-art methods.

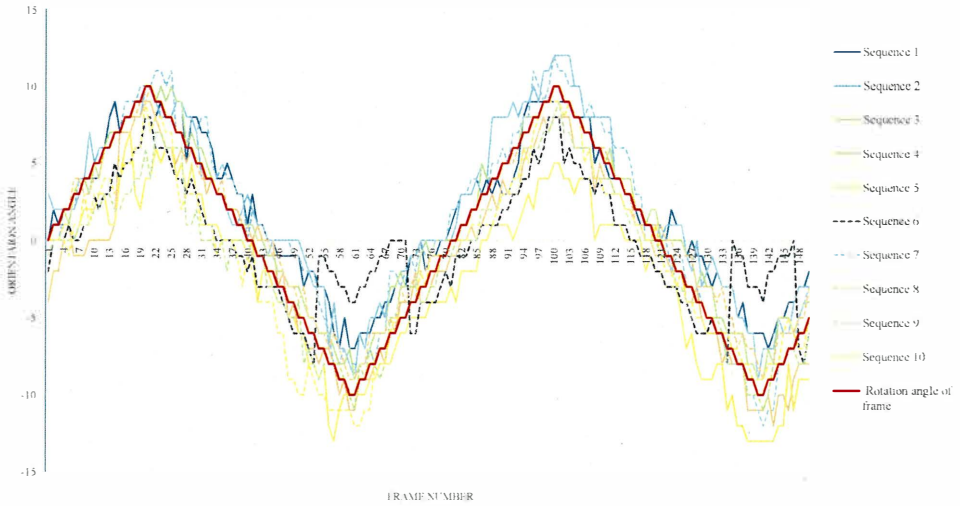


Figure 4.3: Results of HL orientation estimation on the 10 sequences with rotation. Red line represents rotation angles of dataset's frame.

4.2 Local texture based borderline detection of mowing using genetic algorithm

In this section, I explain a local texture based borderline detection of mowing using genetic algorithm. With the development of automatic robots technology, it is using for more and more fields, mowing is still done manually. One of the important reasons of this is that it is difficult to automatically plan the mowing path. Therefore, we proposed a method for finding the boundary line between the cut grass and the uncut grass based on GLCM (greylevel co-occurrence matrix) for this problem. The features we extracted were energy, contrast, correlation and entropy by each GLCM, then using the average of each feature to segment the image of the grass. We used these features to find a boundary line that maximizes the difference in features on both sides. Finally, GA (genetic algorithm) is used to solve the maximization

problem. The experiment shows that it is feasible to distinguish between the cut grass and the uncut grass using GA and GLCM features.

4.2.1 Introduction

Computer vision technique is more and more widespread along with the development of these techniques, and these are applied for many actual products. However, the mowing is not automated completely yet. Using traditional pure manual mowing takes not only much time, but also the cutting effect is not good. Nowadays, a widely used mowing machine cannot efficiently cut grassland automatically. If we use these mowing machines to cut grass, it will take much time to repeat mechanical labour. Although it can improve the efficiency of mowing, it is dangerous and annoying for operators. Therefore, if we can develop an automatic mowing robot will save a lot of human resources and material resources [63].

To develop a robot that can automatically cut grass for various scenes, a key component is planning the path of the mowing robot. This is used to guide the mowing robot to automatically identify the path that should be followed so that it does not need to operate the mowing robot. There are many means for planning the path, the mainstreams include using GPS and image, but the accuracy of GPS is not satisfied with our requirements. In this study, we seek out the boundary line of the cut and uncut parts by using an image. However, results in the following factors that are very hard, the various scenes, the non-regular distribution of grass, different illumination and so on. In this study, we propose a method that is basing GLCM and GA to find out the boundary line.

A. Texture with GLCM The texture is the surface identity of the object. Usually,

the surface of various objects has different texture features. The texture feature is one of the important features that can reflect the surface of different objects in an image. Different objects can be distinguished by comparing the texture information of the surfaces of different objects. As shown in Fig.4.4, the texture of the cut and uncut grass in the image is different. In this study, we used gray level co-occurrence matrix (GLCM) to extract texture features from an image. GLCM can extract the features of texture from the image in different directions. The method records the pixel pair information in part of the image with different directions to construct GLCMs, and then calculate features by GLCM with a series of equations. By comparing the texture information in an image, we can find the boundary line between the cut grass and the uncut grass, thus guiding the mowing robot to work on the path that should be followed.

4.2.2 Related Work

Image segmentation is an important method in computer vision that divides an image into parts by features. The traditional method uses color information to segment images. However, when using color information, it is affected by disturbance factors such as weather and illumination, the result of the segmentation may be unstable. Some methods use texture features to reduce the effects of these disturbance factors on an image. In [64, 65] propose using a fixed set of Gabor filters to extract the texture features from an image. In order to make the experimental results more stable. In [66] analyses the texture information of the image using order statistics, and segment images based on these statistics. In [67] uses GLCM to analyze the texture of the B-mode images and extract features to judge lesions. In [68] proposed

Figure 4.4: These pictures show us the boundary line between the cut grass and the uncut grass. The left side of each picture shows we the texture of uncut grass, the right side of each picture shows we the texture of cut grass.

a method that uses GLCM to extract the texture features, then use SVM to classify the obtained feature quantities. However, this approach requires a significant learning data-set. In this study, the GLCM is used to extract the features, and then our problem becomes an optimization problem, we use the GA to solve this optimization problem. The GA simulates the process of the evolution of biological proposed in [69, 1]. It is a search heuristic algorithm used to solve the optimization problem. In [70] using GA to do the segmentation of the computed tomography images to find the prostate. In [71] using a GA identifying a region in HSI for outdoor field weed sensing. In [72] using GA to search objects of interest in a complex scene. In [73] proposed a closed-loop image segmentation system which incorporates a GA to process images with variable environmental and can adapt to the changes in real-world such as time of day, clouds, etc. In [74] proposed an improved Sobel algorithm that uses a GA to automatically set the optimal threshold.

4.2.3 Methodology

The details of the methodology are explained by following paragraphs:

- Problem Description,

- Construction of GLCM,
- Construction of Chromosome of GA.

Problem Description

This study utilizes the method of GLCM and GA to detect the boundary line. The GLCM is used to extract the features in the image's texture. Also, GA is used to find the optimal solution of cut grass and the uncut grass boundary line of the image.

Input is given a gray scale image, noted as I. In this work, we extract GLCM features from I to construct the GLCM features vector which size is $n \times m$, the row is n, the column is m, and the dimensional of the vector is noted as dim, with $dim = 6$. Which include 4 dimensional of GLCM features and 2 dimensional of local information. We want to divide the I into two parts that are cut and uncut grass, noted as I_1 and I_2. The size of I_1 is $n_1 \times m_1$, size of I_2 is $n_2 \times m_2$. $I_1 \cup I_2$ = $\oslash$.$\mathrm{n}_1, n_2, m_1, m_2 \in N^+$. $P_{i,j}$ is an element which include the features in row i column j of the normalized symmetrical GLCM. And we calculate each part weight center vector by formula 1, noted as $G1$ and $G2$, respectively.

$$G_t[k] = \sum_{i,j \in I} P_{i,j}[k], k \in dim, P_{i,j} \in I_t, t \in 1, 2. \tag{4.12}$$

In this work, we define the function D for boundary line to measure the difference between I_1 and I_2.

$$D(I_1, I_2) = \sum_{i=0}^{n} \sum_{j=0}^{m} \sqrt{(G - P_{i,j})^2}, G = \begin{cases} G_2, & P_{i,j} \in I_1 \\ G_1, & otherwise \end{cases}. \tag{4.13}$$

With the D is defined, our purpose become find a boundary line b_m that has the maximal D.

$$b_m = argmax(D(I_1, I_2)) \tag{4.14}$$

Using this formula we can calculate the fitness value for each boundary line.

Construction of GLCM

To construct GLCM, pre-process the image and homogenize the processed image is necessary. At first, the gray level of the input image I need to be reduced, which operator use to reduce the amount of calculation and reduce the noise interference. The gray level is reduced to 8 as same as [75], the processed image noted as I_r . Then, we can use the Ir to construct the GLCM. And also, the size of the constructed GLCM is 8×8 as same as [76]. In the step we construct GLCM with four directions: $0°, 45°, 90°, 135°$. As shown in Fig.4.5. We use a 32×32 sliding window in the image. Then, count the pixel pairs in different directions in this sliding window, this operator show in Fig.4.6. It show us the construct of GLCM, the left side of figure is the pixels in sliding windows, the right side of figure is the GLCM. The Fig.4.6 is a example of $0°$ direction. The pixel pair (1,1) in left is only once, so the coordinate (1,1) of GLCM in right is 1. The pixel pair (5,7) in left is twice, so the coordinate (5,7) of GLCM in right is 2. Next, for each direction construct the GLCM according to the statistical values, which obtained by counting the pixel pairs that the pixels which are neighbouring in different directions on the sliding window.

In this study, we used the four features: energy, entropy, correlation and contrast.

The following function are equations of each feature.

$$Energy = \sum_{i,j=0}^{N-1} (P_{i,j})^2. \tag{4.15}$$

$$Entropy = \sum_{i,j=0}^{N-1} -ln(P_{i,j})P_{i,j}. \tag{4.16}$$

$$Contrast = \sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2. \tag{4.17}$$

$$Correlation = \sum_{i,j=0}^{N-1} P_{i,j}\frac{(i-\mu)(j-\mu)}{\sigma^2}. \tag{4.18}$$

The N is the number of gray levels. The μ is calculated by the following function.

$$\mu = \sum_{i,j=0}^{N-1} iP_{i,j}. \tag{4.19}$$

The σ^2 is calculated by the following function.

$$\mu = \sum_{i,j=0}^{N-1} P_{i,j}(i-\mu)^2. \tag{4.20}$$

Then, we calculate the average features in four directions for every features, the average features and the geometric information x, y is used to construct the features vector of GLCM G.

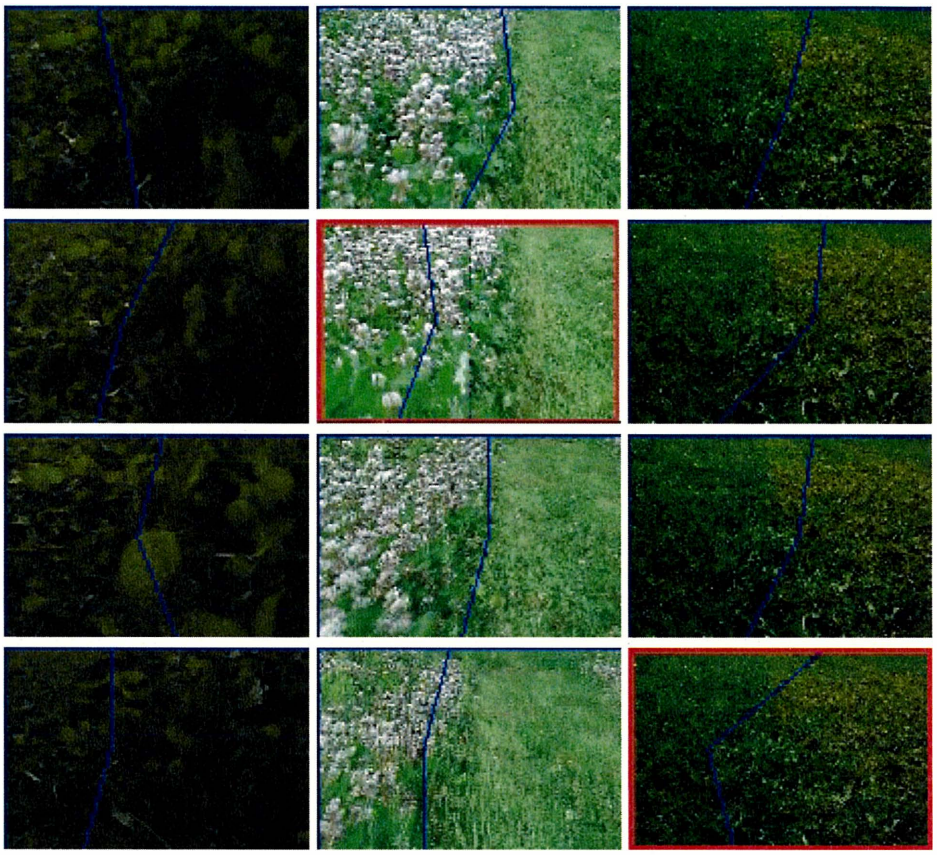


Figure 4.7: Examples of results. The left column is the images of data1, the middle column is the images of data 2, the right column is the images of data 3.

Table 4.3: Comparison between GA and K-means with overlap rate.

Method	Data	Overlap rate			
		min	max	avg	avg
GA	data 1	56.96%	99.31%	87.53%	87.61
	data 2	63.03%	99.16%	85.46%	
	data 3	74.17%	97.96%	89.01%	
K-means	data 1	21.28%	97.59%	74.21%	65.17
	data 2	31.62%	97.91%	76.53%	
	data 3	34.28%	77.20%	47.38%	

find the optimal solution of the weeding boundary line with the maximal D. The experimental results show that the method is effective for searching the weeding boundary line in different environments. In the future, we will use this method for more various environment.

or PI_2 and CI_2 at the same time, then consider this pixel is *correct pixel*, else the pixel is not *correct pixel*. The all pixels in this image is noted as all pixel. Overlap rate is:

$$overlap\,rate = \frac{correct\,pixel}{all\,pixel}. \tag{4.21}$$

Result Evaluation

In this experiment, for the GA parameters 50 initial individuals, and performed 20 iterations to get the optimal solution. We use the same parameter to deal with the three data. The average overlap rate for comparing the processing results of 931 photos with the correct was 87.6%. We used K-means as comparing method to segment the same images, the average overlap rate is 65.1%. We can find that the overlap rate of our method is higher than use K-means. Especially, as shown in Table 4.3 the results of data 3 indicate that our method has a great improvement for the difficult case with short grass. The example results shown in Fig.4.7. From the Fig.4.7, we can find that the method of using GLCM and GA to seek for the boundary of the cut and uncut grass is feasible. The image with red border in Fig.4.7 is the inaccurate result, that due to the features difference between different position in uncut grass is more then the cut and uncut grass (e.g., target image include flower and grass two parts).

4.2.5 Conclusion

In this study, we present a method for finding weeding boundaries based on GLCM and GA. First, we extract the GLCM features in the image and then using these features to define a function D for measure boundary line. Finally, use GA to

6-bit binary gene to represent the position of the column for each point. That is to say, our chromosome code by 18-bit binary code. Each binary code of chromosome can map to a boundary line that total has 2^{18} possibilities.

4.2.4 Experimental result

The details of the experimental results are explained by following paragraphs. Those are experimental environment, evaluation criteria, and result evaluation.

Experimental Environment

In this experiment, we collect three videos in different environments and under high variability of lighting conditions. The video No.1 and No.3 are taken in an overcast day and grass distributed messy. However, grass of video No.3 is short compare then grass of video No.1, and it is more different to recognize the cut and uncut grass. The video No.2 is taken on a bright day and grass distributed evenly. Moreover, we extract 400, 200, 331 images from video No.1, No.2 and No.3 form data1 data2 and data3, respectively. Three data-set total of 931 grass images are used to test the performance of our method.

Evaluation Criteria

In this experiment, we used the overlap rate between the ground truth and our results to measure the accuracy. Note the part left then the boundary in ground truth as PI_1, note the part right then the boundary in ground truth as PI_2, note the part left then the boundary in our result as CI_1, note the part right then the boundary in our result as CI_2, note the pixel in this image as $q_{i,j}$. If the pixel is belong to PI_1 and CI_1

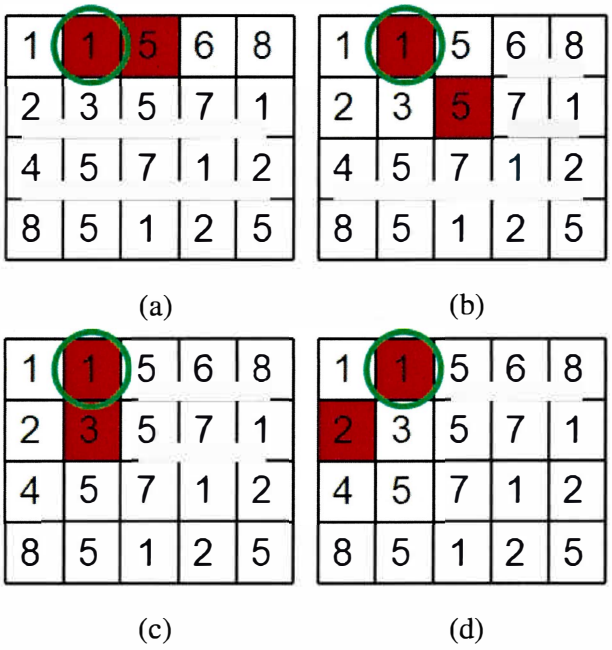


Figure 4.5: Construction of GLCMs in four directions. The four tables are the a part of I_r . The pixel marked by green circle in each table is being processed. The red painted girds of each table are pixel pair in each direction. (a), (b), (c) and (d) the direction are $0°, 45°, 90°, 135°$, respectively.

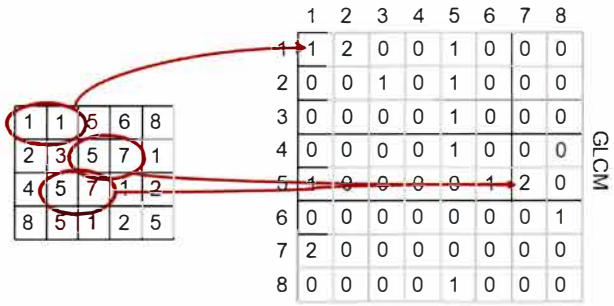


Figure 4.6: The operation of GLCM construction.

Construction of Chromosome of GA

In this study, we want to search the boundary line with the maximum value of function D. The boundary line of the grass image is always can divide image to the left and right two parts. We designed a boundary line control by three points which are in the image first, middle and bottom row to segment image. Then we use a

Chapter 5

Conclusion

In this thesis, we proposed an evolutionary approach for horizon detection and tracking in images. We formulate the horizon line detection as a global optimization problem, and a genetic algorithm is utilized for the optimization of the horizon line parameters. We also adopted a coarse-to-fine approach and fast local feature estimation to meet real-time processing requirements on high-resolution images. The local features are defined from vanishing line characteristics, and they can be fast estimated on narrow regions of an image.

As a result, the effectiveness of the method is confirmed in some researches, which are horizon detection in maritime images and horizon detection in road images. The proposed method can be used for other prominent line detection problems, and we tested the method on the real problem of borderline detection of mowing.

In the future, we would like to improve the proposed methods from line detection to curve detection problems and apply them to many practical systems.

Appendix A

Details of the SMD and Comparison Result

In this appendix section, we inform additional information abour datasets and the criteria used for an evaluation of the proposed method and comparisons. The proposed method applied to the Singapore Maritime Dataset (SMD) and Buoy-Dataset (BD). The SMD contains onboard videos taken by a camera mounted on moving board and onshore videos taken by a static camera installed onshore. Each frame has 1920×1080 pixels resolution. The SMD has challenges including a strong presence of wakes and waves, terrestrial features close to the horizon, and a difference in the color of the sea. The videos of BD are taken by a camera mounted on a floating buoy. Each frame has 800×600 pixels resolution. The onboard videos of SMD and BD have a large variation in the orientation and position of the horizon line between adjacent frames. All videos and ground truths of the horizon line are downloaded in [77]. The ground truth of the horizon line is represented by a vertical position Y relatively upper edge and an angle α between the horizon line and

Table A.1: Details of datasets.

Dataset	Buoy	Singapore maritime	
		Onboard	Onshore
No. of videos	10	10	35
No. of frames	996	2915	16819
Min(Y-mean(Y))	-261.4424	-364.9797	-197.7587
Max(Y-mean(Y))	339.1325	535.0281	395.6408
Standard deviation of Y	108.6655	162.4086	103.7937
Min(α-mean(α))	-15.9123	-3.8498	-1.6998
Max(α-mean(α))	20.5301	4.1307	3.9690
Standard deviation of α	4.4371	1.8976	1.1653

horizontal edge of an image.

Due to updating the SMD databases, different numbers of videos and frames are used in various research. In chapter 3, we organized the databases the same as [33, 32] in rorder to compare our result with their results. They used the sampled frames from videos of datasets. Specially, they not used the blurred frames in the videos. Liang et al. [34] have not cut the blurred frames from videos of the datasets. Therefore, we also tested the our method on dataset which same as [34]. The detailed procedure of organizing databases is available in [78]. We compared our results to the result of [34].

The details of the datasets are given in Table A.1.

In general, to evaluate the performance of the horizon line detection, the angle error α^{err} and vertical position error y^{err} are calculated in the estimated result and the ground truth at every frame shown in (A.1) and (A.2).

$$\alpha_i^{err} = \alpha_i^{gt} - \alpha_i^{est} \tag{A.1}$$

Table A.2: The number of frames and ground truths for onboard videos of SMD.

Name of test sequences	The number of ground truth/frames
MVI_0788_VIS_OB_HorizonGT	299
MVI_0789_VIS_OB_HorizonGT	279
MVI_0790_VIS_OB_HorizonGT	299
MVI_0792_VIS_OB_HorizonGT	299
MVI_0794_VIS_OB_HorizonGT	288
MVI_0795_VIS_OB_HorizonGT	255
MVI_0796_VIS_OB_HorizonGT	299
MVI_0797_VIS_OB_HorizonGT	299
MVI_0801_VIS_OB_HorizonGT	299
MVI_0804_VIS_OB_HorizonGT	299
The total number of frames	2915

$$y_i^{err} = y_i^{gt} - y_i^{est} \tag{A.2}$$

Average errors are calculated as:

$$\alpha_{ave}^{err} = \frac{1}{N}\sum_{i=0}^{N} |\alpha_i^{err}| \tag{A.3}$$

$$y_{ave}^{err} = \frac{1}{N}\sum_{i=0}^{N} |y_i^{err}| \tag{A.4}$$

where N is the total number of frames in the dataset.

In the [34], the original frame of compared methods as on H-PDF[48], H-DE[46], and H-CI[41] is down-sampled in order to estimate the performance under the approximate computational complexity. Scale factor for SMD is 0.05, 0.04 and 0.1 on H-PDF, H-DE and H-CI respectively. Scale factor for BD is 0.05, 0.05 and 0.1 on H-PDF, H-DE and H-CI respectively. Results of the proposed method are shown in Table A.4 to Table A.7. Average errors α_{ave}^{err} and y_{ave}^{err} are shown in Table A.4 and Table A.5 respectively. Standard deviations of vertical position error and

angle error are shown in Table A.6 and Table A.7.

As the experiment result, the proposed method with both global and local feature estimation has the highest performance in the onboard dataset of SMD and BD from compared methods. The average vertical position error and the average angle error of the proposed method are the lowest from all compared methods in the onboard dataset of SMD and BD. The proposed method with only local feature estimation also has a higher performance in the onboard dataset of SMD and BD than compared methods. In the onshore dataset of SMD, the average vertical position error of [34] is smaller than proposed method with only local feature estimation. The angle error of the proposed methods are smaller than all compared methods in the onshore dataset of SMD and BD. The experimental result demonstrates that the proposed method is more robust and can accurately detect the horizon line.

Table A.3: The number of frames and ground truths for onshore videos of SMD.

Name of test sequences	The number of ground truth
MVI_1469_VIS_HorizonGT	600
MVI_1470_VIS_HorizonGT	266
MVI_1471_VIS_HorizonGT	299
MVI_1474_VIS_HorizonGT	445
MVI_1478_VIS_HorizonGT	477
MVI_1479_VIS_HorizonGT	206
MVI_1481_VIS_HorizonGT	409
MVI_1482_VIS_HorizonGT	454
MVI_1483_VIS_HorizonGT	299
MVI_1484_VIS_HorizonGT	687
MVI_1578_VIS_HorizonGT	505
MVI_1582_VIS_HorizonGT	540
MVI_1583_VIS_HorizonGT	251
MVI_1584_VIS_HorizonGT	539
MVI_1587_VIS_HorizonGT	600
MVI_1592_VIS_HorizonGT	491
MVI_1609_VIS_HorizonGT	505
MVI_1610_VIS_HorizonGT	543
MVI_1612_VIS_HorizonGT	261
MVI_1613_VIS_HorizonGT	626
MVI_1614_VIS_HorizonGT	582
MVI_1615_VIS_HorizonGT	566
MVI_1617_VIS_HorizonGT	600
MVI_1619_VIS_HorizonGT	473
MVI_1620_VIS_HorizonGT	502
MVI_1622_VIS_HorizonGT	309
MVI_1623_VIS_HorizonGT	522
MVI_1624_VIS_HorizonGT	494
MVI_1625_VIS_HorizonGT	995
MVI_1626_VIS_HorizonGT	556
MVI_1627_VIS_HorizonGT	600
MVI_1640_VIS_HorizonGT	310
MVI_1644_VIS_HorizonGT	252
MVI_1645_VIS_HorizonGT	535
MVI_1646_VIS_HorizonGT	520
The total number of frames	16819

Table A.4: Average vertical position error (in pixel).

Methods	Dataset		
	Buoy	Singapore maritime	
	Onboard	Onboard	Onshore
MSCM[49]	4.2395	218.2402	10.9713
H-CI[41]	139.2665	50.9551	87.7430
H-HC[44]	6.1450	132.7571	42.0757
H-PDF[48]	174.2168	40.3386	43.3907
H-DE[46]	30.1506	32.0536	40.2241
ESSR-HV[34]	3.0944	15.7981	7.0688
Ganbold [51]	1.2698	4.6314	9.0122
Proposed	0.9740	4.4877	6.3846

Table A.5: Average angle error (in degree).

Methods	Dataset		
	Buoy	Singapore maritime	
	Onboard	Onboard	Onshore
MSCM[49]	0.5627	1.5096	0.3247
H-CI[41]	3.8630	1.8975	1.8172
H-HC[44]	0.4079	0.8753	0.3651
H-PDF[48]	6.7451	2.4016	0.8765
H-DE[46]	3.4527	3.8837	3.1037
ESSR-HV[34]	0.3847	0.4112	0.3124
Ganbold [51]	0.2311	0.2330	0.1805
Proposed	0.1812	0.3386	0.2869

Table A.6: Standard deviation on vertical position error.

Methods	Dataset		
	Buoy	Singapore maritime	
	Onboard	Onboard	Onshore
MSCM[49]	15.0124	214.6823	35.5998
H-CI[41]	116.2509	85.3445	158.8357
H-HC[44]	33.2009	209.1154	109.3672
H-PDF[48]	221.9335	89.9382	126.1882
H-DE[46]	60.6384	59.8974	55.4034
ESSR-HV[34]	17.6911	50.0784	13.4168
Ganbold [51]	1.0033	39.2421	17.7337
Proposed	1.3738	12.1686	12.8971

Table A.7: Standard deviation on angle error.

Methods	Dataset		
	Buoy	Singapore maritime	
	Onboard	Onboard	Onshore
MSCM[49]	0.7830	2.2527	0.4380
H-CI[41]	5.1883	2.8911	2.9519
H-HC[44]	0.6015	1.5724	0.6409
H-PDF[48]	8.9383	2.8667	1.5719
H-DE[46]	4.2080	4.3988	4.0729
ESSR-HV[34]	0.6773	0.7229	0.4085
Ganbold [51]	0.1954	0.4368	0.2745
Proposed	0.2619	0.8369	0.5445

Appendix B

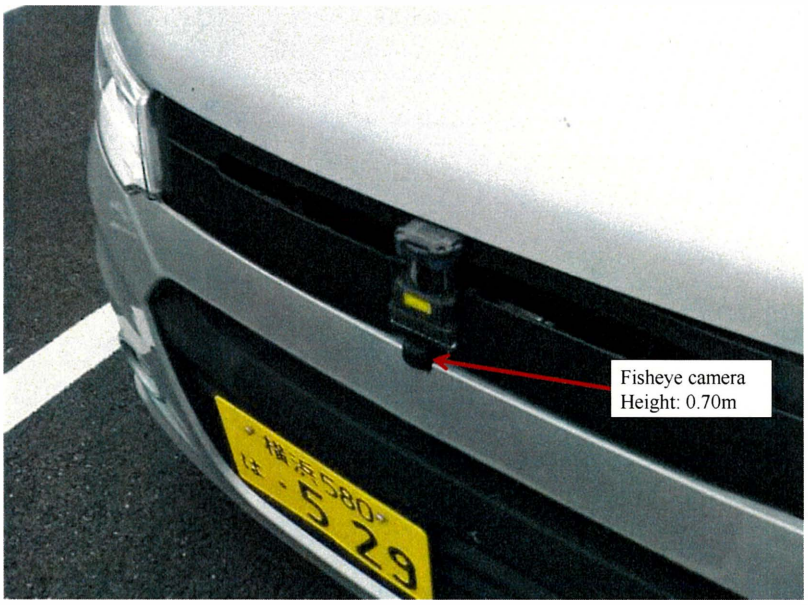
Dataset Captured by Fisheye Camera

Data was recorded using a fisheye camera mounted on the front-side bumper of the car The location of camera shown in Fig. 1. The camera is at a height of 0.7 m. The horizontal angle of view is 192 degrees. The vertical angle of view is 151 degrees. To estimate the position on the image plane where the moving object on the ground appears, image data was acquired by the following situation.

- Situation of car: stopped (0 km/h).
- Distance between car to measurement point: 0.2 m to 1.2 m (0.1 m increments), 1.4 m, 1.6 m, 1.8 m, 2.0 m, 3.0 m, 4.0 m, 5.0 m, 7.0 m, 10.0 m, 15.0 m, 20.0 m, 25.0 m
- Orientation between car direction and measurement point: 0, 15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 175, 180 (degrees)

Fig. B.2 shows the measured points on the input scene and the position of measured points. As shown this figure, the vanishing of the ground on the image plane starts from 25m. In other words, it is able to filter the near objects from 20m using horizon

line. The dataset is captured the real traffic 10 sequences involving the moving objects of pedestrian, cars, and bicycles in different environments. Sample frames are shown in Fig. B.3 and Fig. B.4. The sequences 1-9 are captured when the car stops at the crossroad. The sequence 10 is captured when the car moves into the crossroad. The ground truths of HL and moving object are labeled manually per frame. Each sequence consists of 150 frames in 320 $\times$ 240 pixel resolution. The frame rate is 30fps.



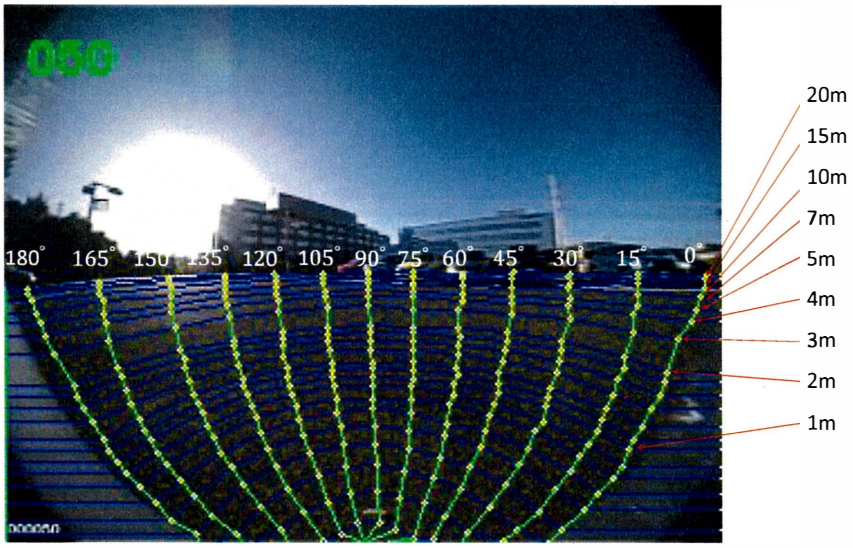
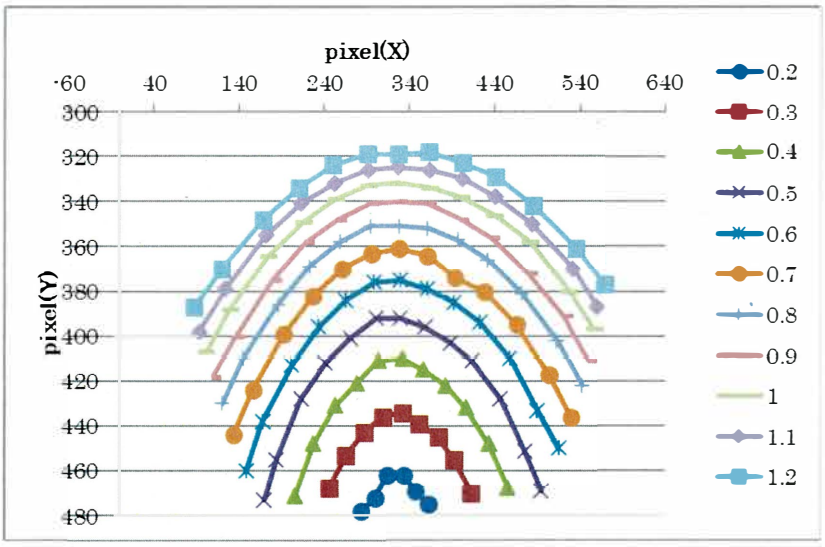
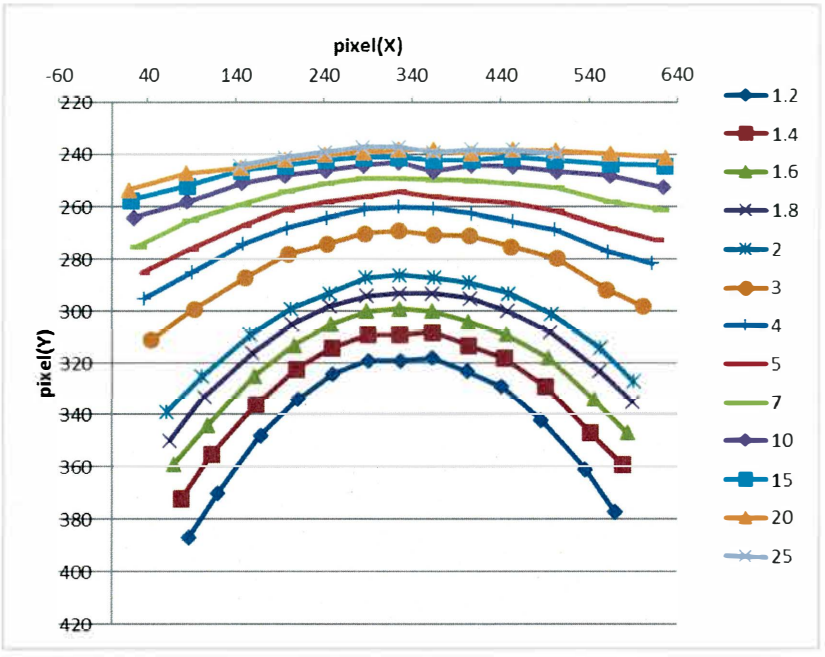


Figure B.1: The location of the fisheye camera is shown in the top side figure. The measured points on the input scene is shown in the bottom side figure.



(a)



(b)

Figure B.2: The location of measured points. (a): The location of measured points between from 0.2m to 1.2m. (b): The location of measured points between from 1.2m to 25m.

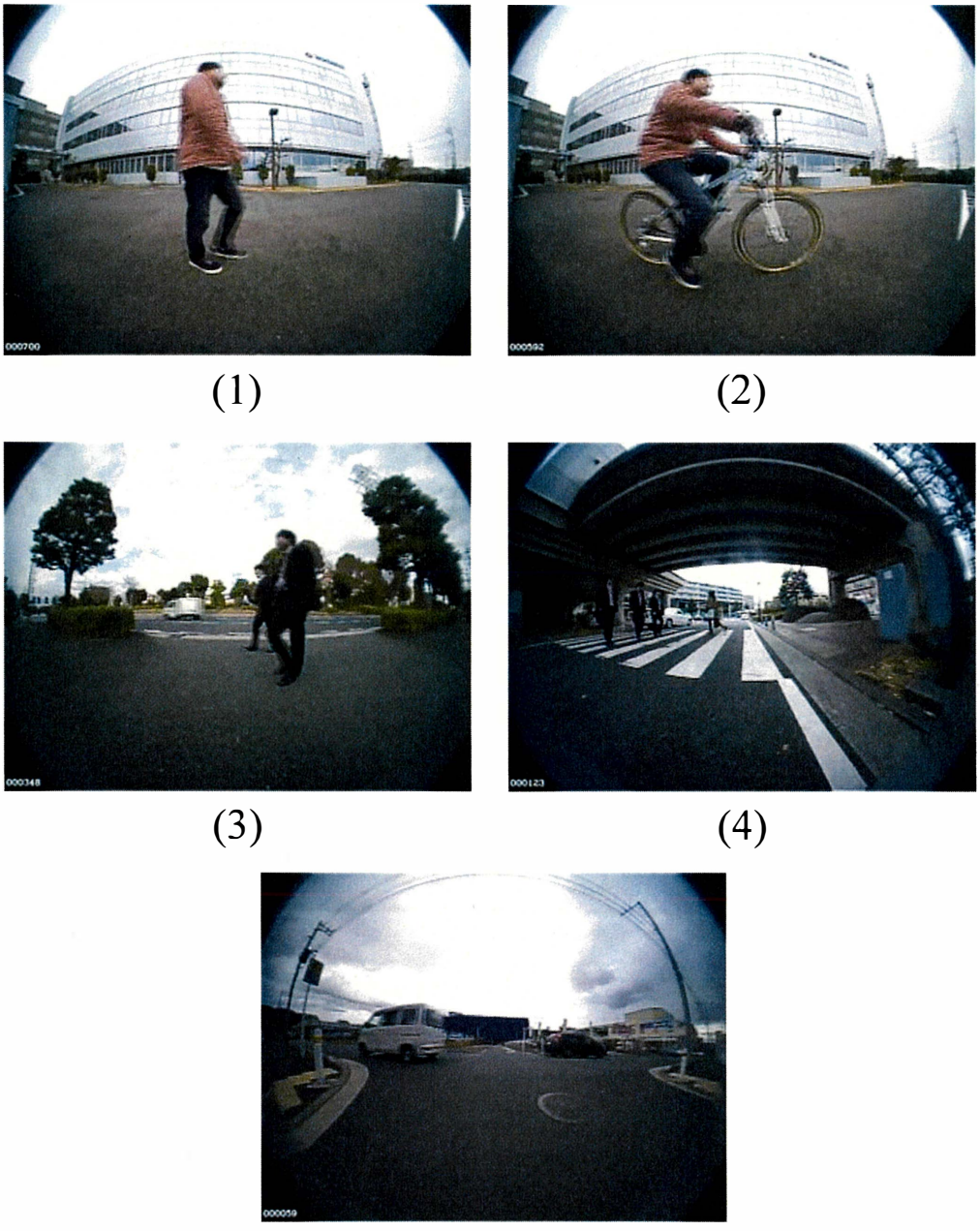


Figure B.3: Sample frames of sequence 1-5.

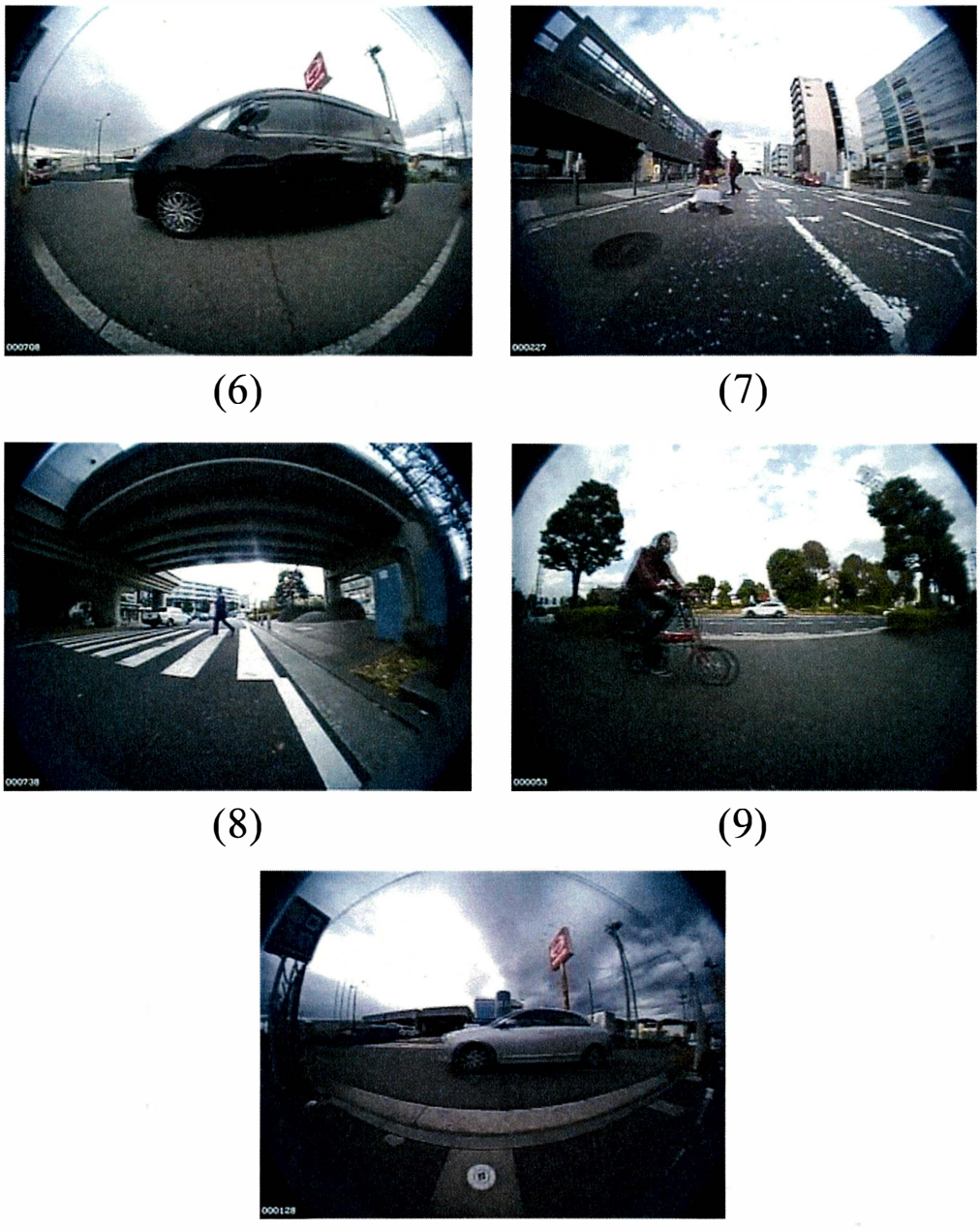


Figure B.4: Sample frames of sequence 6-10.

Bibliography

[1] J. Holland, "Adaptation in natural and artificial systems, univ. of mich. press," *Ann Arbor*, 1975.

[2] P. Civicioglu and E. Besdok, "A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artificial intelligence review*, vol. 39, no. 4, pp. 315–346, 2013.

[3] M. Mitchell, "An introduction to genetic algorithms. cambridge," 1996.

[4] H. J. Ko and G. W. Evans, "A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3pls," *Computers & Operations Research*, vol. 34, no. 2, pp. 346–366, 2007.

[5] M. Gen, F. Altiparmak, and L. Lin, "A genetic algorithm for two-stage transportation problem using priority-based encoding," *OR spectrum*, vol. 28, no. 3, pp. 337–354, 2006.

[6] S. Nakaya, T. Koide, and S. Wakabayashi, "An adaptive genetic algorithm for vlsi floorplanning based on sequence-pair," in *2000 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 3. IEEE, 2000, pp. 65–68.

[7] C. L. Valenzuela and P. Y. Wang, "Vlsi placement and area optimization using a genetic algorithm to breed normalized postfix expressions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 390–401, 2002.

[8] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 394–407, 2019.

[9] L. Xie and A. Yuille, "Genetic cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1379–1388.

[10] K.-C. Wong, "Evolutionary algorithms: concepts, designs, and applications in bioinformatics," in *Nature-Inspired Computing: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2017, pp. 111–137.

[11] A. Ismail, A. Sheta, and M. Al-Weshah, "A mobile robot path planning using genetic algorithm in static environment," *Journal of Computer Science*, vol. 4, no. 4, pp. 341–344, 2008.

[12] A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1564–1572, 2012.

[13] T. Nakane, N. Bold, H. Sun, X. Lu, T. Akashi, and C. Zhang, "Application of evolutionary and swarm optimization in computer vision: a literature survey," *IPSJ Transactions on Computer Vision and Applications*, vol. 12, no. 1, pp. 1–34, 2020.

[14] U. Ganbold, T. Akashi, C. Zhang, H. Tomita, and H. Kubota, "Horizontal line detection using genetic algorithm," in *International Workshop on Advanced Image Technology (IWAIT) 2019*, vol. 11049. International Society for Optics and Photonics, 2019, p. 110490S.

[15] T. Akashi, Y. Wakasa, K. Tanaka, S. Karungaru, and M. Fukumi, "Using genetic algorithm for eye detection and tracking in video sequence," *Journal of Systemics, Cybernetics and Informatics*, vol. 5, no. 2, pp. 72–78, 2007.

[16] M. Awad, K. Chehdi, and A. Nasri, "Multicomponent image segmentation using a genetic algorithm and artificial neural network," *IEEE Geoscience and remote sensing letters*, vol. 4, no. 4, pp. 571–575, 2007.

[17] W.-B. Tao, J.-W. Tian, and J. Liu, "Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm," *Pattern Recognition Letters*, vol. 24, no. 16, pp. 3069–3078, 2003.

[18] L. Trujillo and G. Olague, "Automated design of image operators that detect interest points," *Evolutionary Computation*, vol. 16, no. 4, pp. 483–507, 2008.

[19] C. B. Perez and G. Olague, "Genetic programming as strategy for learning image descriptor operators," *Intelligent Data Analysis*, vol. 17, no. 4, pp. 561–583, 2013.

[20] J. Sato and T. Akashi, "Deterministic crowding introducing the distribution of population for template matching," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 13, no. 3, pp. 480–488, 2018.

[21] C. Zhang and T. Akashi, "Fast affine template matching over galois field." in *BMVC*, 2015, pp. 121–1.

[22] D. Sholomon, O. David, and N. Netanyahu, "A generalized genetic algorithm-based solver for very large jigsaw puzzles of complex types," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, 2014.

[23] J. Sato and T. Akashi, "High-speed multiview face localization and tracking with a minimum bounding box using genetic algorithm," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 12, no. 5, pp. 736–743, 2017.

[24] R. Senaratne, S. Halgamuge, and A. Hsu, "Face recognition by extending elastic bunch graph matching with particle swarm optimization." *Journal of Multimedia*, vol. 4, no. 4, 2009.

[25] C. Liu and H. Wechsler, "Evolutionary pursuit and its application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 570–582, 2000.

[26] S. E. Asch and H. A. Witkin, "Studies in space orientation: I. perception of the upright with displaced visual fields." *Journal of Experimental Psychology*, vol. 38, no. 3, p. 325, 1948.

[27] J. J. Gibson, "The perception of the visual world." 1950.

[28] C. Herdtweck and C. Wallraven, "Estimation of the horizon in photographed outdoor scenes by human and machine," *PloS one*, vol. 8, no. 12, p. e81462, 2013.

[29] R. Verbickas and A. Whitehead, "Sky and ground detection using convolutional neural networks," in *International Conference on Machine Vision and Machine Learning (MVML)*, 2014.

[30] S. Todorovic, M. C. Nechyba, and P. G. Ifju, "Sky/ground modeling for autonomous mav flight," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 1. IEEE, 2003, pp. 1422–1427.

[31] A. Carrio, H. Bavle, and P. Campoy, "Attitude estimation using horizon detection in thermal images," *International Journal of Micro Air Vehicles*, vol. 10, no. 4, pp. 352–361, 2018.

[32] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Muscowert: multi-scale consistence of weighted edge radon transform for horizon detection in maritime images," *JOSA A*, vol. 33, no. 12, pp. 2491–2500, 2016.

[33] C. Y. Jeong, H. S. Yang, and K. Moon, "Fast horizon detection in maritime images using region-of-interest," *International Journal of Distributed Sensor Networks*, vol. 14, no. 7, p. 1550147718790753, 2018.

[34] D. Liang and Y. Liang, "Horizon detection from electro-optical sensors under maritime environment," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 1, pp. 45–53, 2019.

[35] T. Akashi, U. Ganbold, and H. Tomita, "Moving object detection apparatus and moving object detection method," Jun. 1 2021, uS Patent 11,024,042.

[36] A. M. Neto, A. C. Victorino, I. Fantoni, and D. E. Zampieri, "Robust horizon finding algorithm for real-time autonomous navigation based on monocular vi-

sion," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2011, pp. 532–537.

[37] L. Wang, Q. Wu, J. Liu, S. Li, and R. R. Negenborn, "State-of-the-art research on motion control of maritime autonomous surface ships," *Journal of Marine Science and Engineering*, vol. 7, no. 12, p. 438, 2019.

[38] A. Stateczny, W. Kazimierski, P. Burdziakowski, W. Motyl, and M. Wisniewska, "Shore construction detection by automotive radar for the needs of autonomous surface vehicle navigation," *ISPRS International Journal of Geo-Information*, vol. 8, no. 2, p. 80, 2019.

[39] Y. Wang, R. Tan, G. Xing, J. Wang, X. Tan, X. Liu, and X. Chang, "Aquatic debris monitoring using smartphone-based robotic sensors," in *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IEEE, 2014, pp. 13–24.

[40] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 1993–2016, 2017.

[41] S. Fefilatyev, D. Goldgof, M. Shreve, and C. Lembke, "Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system," *Ocean Engineering*, vol. 54, pp. 1–12, 2012.

[42] H. Bouma, D.-J. J. de Lange, S. P. van den Broek, R. A. Kemp, and P. B. Schwering, "Automatic detection of small surface targets with electro-optical

sensors in a harbor environment," in *Electro-Optical Remote Sensing, Photonic Technologies, and Applications II*, vol. 7114. International Society for Optics and Photonics, 2008, p. 711402.

[43] Y. Sun and L. Fu, "Coarse-fine-stitched: A robust maritime horizon line detection method for unmanned surface vehicle applications," *Sensors*, vol. 18, no. 9, p. 2825, 2018.

[44] E. Gershikov, T. Libe, and S. Kosolapov, "Horizon line detection in marine images: which method to choose?" *International Journal on Advances in Intelligent Systems*, vol. 6, no. 1, 2013.

[45] G.-Q. Bao, S.-S. Xiong, and Z.-Y. Zhou, "Vision-based horizon extraction for micro air vehicle flight control," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 3, pp. 1067–1072, 2005.

[46] S. M. Ettinger, M. C. Nechyba, P. G. Ifju, and M. Waszak, "Towards flight autonomy: Vision-based horizon detection for micro air vehicles," in *Florida Conference on Recent Advances in Robotics*, vol. 2002, 2002.

[47] ——, "Vision-guided flight stability and control for micro air vehicles," *Advanced Robotics*, vol. 17, no. 7, pp. 617–640, 2003.

[48] I. Lipschutz, E. Gershikov, and B. Milgrom, "New methods for horizon line detection in infrared and visible sea images," *Int. J. Comput. Eng. Res*, vol. 3, no. 3, pp. 1197–1215, 2013.

[49] D. K. Prasad, D. Rajan, C. K. Prasath, L. Rachmawati, E. Rajabally, and C. Quek, "Mscm-life: multi-scale cross modal linear feature for horizon de-

tection in maritime images," in *2016 IEEE region 10 conference (TENCON)*. IEEE, 2016, pp. 1366–1370.

[50] C. Jeong, H. S. Yang, and K. Moon, "A novel approach for detecting the horizon using a convolutional neural network and multi-scale edge detection," *Multidimensional Systems and Signal Processing*, vol. 30, no. 3, pp. 1187–1204, 2019.

[51] U. Ganbold and T. Akashi, "The real-time reliable detection of the horizon line on high-resolution maritime images for unmanned surface-vehicle," in *2020 International Conference on Cyberworlds (CW)*. IEEE, 2020, pp. 204–210.

[52] M. Atiquzzaman, "Coarse-to-fine search technique to detect circles in images," *The International Journal of Advanced Manufacturing Technology*, vol. 15, no. 2, pp. 96–102, 1999.

[53] C. Spence, J. Pearson, and J. Bergen, "Coarse-to-fine image search using neural networks," *Advances in Neural Information Processing Systems*, vol. 7, pp. 981–988, 1994.

[54] S. Fefilatyev, M. Shreve, and D. Goldgof, "Detection of the vanishing line of the ocean surface from pairs of scale-invariant keypoints," in *International Conference on Image Analysis and Processing*. Springer, 2013, pp. 161–169.

[55] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricár, S. Milz, M. Simon, K. Amende *et al.*, "Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving," *arXiv preprint arXiv:1905.01489*, 2019.

[56] J. J. Gibson, *The ecological approach to visual perception: classic edition.* Psychology Press, 2014.

[57] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[58] R. Okada, Y. Taniguchi, K. Furukawa, and K. Onoguchi, "Obstacle detection using projective invariant and vanishing lines," in *null.* IEEE, 2003, p. 330.

[59] M. Bertozzi, A. Broggi, M. Carletti, A. Fascioli, T. Graf, P. Grisleri, and M. Meinecke, "Ir pedestrian detection for advanced driver assistance systems," in *Joint Pattern Recognition Symposium.* Springer, 2003, pp. 582–590.

[60] H. Kong, J.-Y. Audibert, and J. Ponce, "Vanishing point detection for road detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 96–103.

[61] F. Schaffalitzky and A. Zisserman, "Planar grouping for automatic detection of vanishing lines and points," *Image and Vision Computing*, vol. 18, no. 9, pp. 647–658, 2000.

[62] T. Ahmad, G. Bebis, E. E. Regentova, and A. Nefian, "A machine learning approach to horizon line detection using local features," in *International Symposium on Visual Computing.* Springer, 2013, pp. 181–193.

[63] R. W. Hicks II and E. L. Hall, "Survey of robot lawn mowers," in *Intelligent Robots and Computer Vision XIX: Algorithms, Techniques, and Active Vision*, vol. 4197. International Society for Optics and Photonics, 2000, pp. 262–269.

[64] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[65] T. P. Weldon, W. E. Higgins, and D. F. Dunn, "Efficient gabor filter design for texture segmentation," *Pattern recognition*, vol. 29, no. 12, pp. 2005–2015, 1996.

[66] A. Schepelmann, R. E. Hudson, F. L. Merat, and R. D. Quinn, "Visual segmentation of lawn grass for a mobile robotic lawnmower," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 734–739.

[67] X. Yang, S. Tridandapani, J. J. Beitler, D. S. Yu, E. J. Yoshida, W. J. Curran, and T. Liu, "Ultrasound glcm texture analysis of radiation-induced parotid-gland injury in head-and-neck cancer radiotherapy: An in vivo study of late toxicity," *Medical physics*, vol. 39, no. 9, pp. 5732–5739, 2012.

[68] G.-m. Xian, "An identification method of malignant and benign liver tumors from ultrasonography based on glcm texture features and fuzzy svm," *Expert Systems with Applications*, vol. 37, no. 10, pp. 6737–6741, 2010.

[69] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing*, vol. 4, no. 2, pp. 65–85, 1994.

[70] P. Ghosh and M. Mitchell, "Segmentation of medical images using a genetic algorithm," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 1171–1178.

[71] L. Tang, L. Tian, and B. L. Steward, "Color image segmentation with genetic algorithm for in-field weed sensing," *Transactions of the ASAE*, vol. 43, no. 4, p. 1019, 2000.

[72] D. L. Swets, B. Punch, and J. Weng, "Genetic algorithms for object recognition in a complex scene," in *Proceedings., International Conference on Image Processing*, vol. 2. IEEE, 1995, pp. 595–598.

[73] B. Bhanu, S. Lee, and J. Ming, "Adaptive image segmentation using a genetic algorithm," *IEEE Transactions on systems, man, and cybernetics*, vol. 25, no. 12, pp. 1543–1567, 1995.

[74] Z. Jin-Yu, C. Yan, and H. Xian-Xiang, "Edge detection of images based on improved sobel operator and genetic algorithms," in *2009 International Conference on Image Analysis and Signal Processing.* IEEE, 2009, pp. 31–35.

[75] P. Mohanaiah, P. Sathyanarayana, and L. GuruKumar, "Image texture feature extraction using glcm approach," *International journal of scientific and research publications*, vol. 3, no. 5, pp. 1–5, 2013.

[76] G. Beliakov, S. James, and L. Troiano, "Texture recognition by using glcm and various aggregation functions," in *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence).* IEEE, 2008, pp. 1472–1476.

[77] D. Prasad. Singapore-maritime-dataset. [Accessed: 2020-02-29]. [Online]. Available: https://sites.google.com/site/dilipprasad/home/singapore-maritime-dataset

[78] D. Liang. Illustration on usage of singapore maritime dataset. [Accessed: 2020-02-29]. [Online]. Available: https://github.com/LeongDong/Sea-sky-line-Detection