공학석사 학위논문

Multi-Lane Detection in Highway and Urban Driving Environment

고속도로 및 시내 주행 환경에서의 다중차선인식

2013 년 8 월

서울대학교 대학원

전기·컴퓨터공학부

허 준 화

Multi-Lane Detection in Highway and Urban Driving Environment

고속도로 및 시내 주행 환경에서의 다중차선인식

지도교수 서 승 우

이 논문을 공학석사 학위논문으로 제출함
2013 년 6 월

서울대학교 대학원
전기·정보공학부
허 준 화

허준화의 공학석사 학위논문을 인준함
2013 년 6 월

위 원 장 최 진 영 (인)

부위원장 서 승 우 (인)

위 원 조 남 익 (인)

Abstract

Over the past few decades, the need has arisen for multi-lane detection algorithms for use in vehicle safety-related applications. In this thesis dissertation, we propose new multi-lane detection algorithms that work well in highway and urban driving situations. These algorithms detect four lane marks, including driving lane marks and adjacent lane marks. First, in the case of highway driving situations the algorithm first detects the two driving lane marks stably, and the algorithm generates ROIs for detecting multiple lanes based on the fact that all lanes are parallel with the same intervals. Additionally, the system recognizes types and colors of each lane. For the urban driving environments, on the other hands, the algorithm basically detects more than four lanes including driving lane marks and adjacent lane marks, and it also aims to detect non-parallel lanes such as are found at intersections, in splitting lanes, and in merging lanes. To detect multi-lane marks successfully in the absence of parallelism, we adopt Conditional Random Fields (CRFs), which are strong models for solving multiple association tasks. We show that CRFs are very effective tools for multi-lane detection because they find an optimal association of multiple lane marks in complex and challenging urban road situations. Through simulations, and by using video sequences and Caltech Lane Datasets with runtime rates of 30 fps, we verify that our algorithm successfully detects non-parallel lanes as well as parallel lanes appearing in highways and urban streets.

Key words: Intelligent Vehicle, Computer Vision, Lane Detection

Student number: 2011-23389

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

For the past few decades, a huge number of researches for intelligent vehicles and autonomous driving vehicles have been progressed. For the safe driving for intelligent vehicles or autonomous driving vehicles, various sensors are adopted in the vehicles, such as a GPS for localization, a front camera for recognizing the front view and a radar for scanning the front area. All sensors are primarily necessary for the safe driving, but the most important one among them can be the front camera because it roles as human eyes and accounts for recognizing almost all details of objects in front of driving vehicles.

With the front camera, various kinds of vision systems for intelligent vehicles have been developed, such as lane detection, vehicle detection, traffic sign recognition, pedestrian detection, road detection, etc. Among them, the basic vision system for autonomous driving is said to be a lane detection which makes a vehicle keep safely driving along the lane. In the dissertation, we basically handle the lane detection algorithms, especially detecting multiple lanes in highway and urban driving environments.

The ultimate goal of the lane detection is detecting and recognizing all lanes on

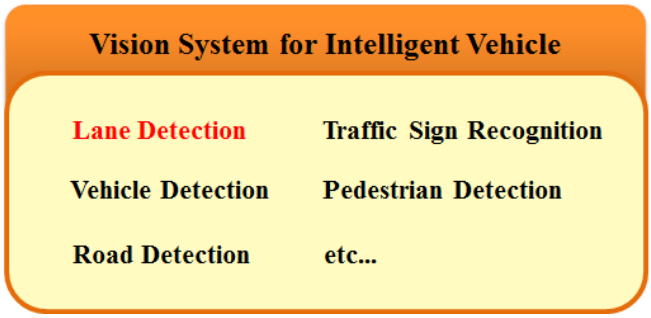


Figure 1.1. Vision systme for intelligent vehicles and autonomous vehicles.

roads in various situations. Detection and Recognition mean that the system can provide not only the location and curvature information of all lanes but also the type and color of each lane. Also, wherever vehicle drives, the system can robustly operate in various weather conditions (sunny, cloudy, rainy, snowy, etc.) and various driving environments (highway, urban, tunnel, etc.).

If the lane detection is robust enough to operate in various situations, the system can be adapted in various vehicle safety-related applications, not only LDWS (Lane Departure Warning System), LKAS (Lane Keeping Assistance System), but also LCW (Lane Changing Warning), DNPW (Do-Not-Pass Warning), and vehicle localization based on the lane detection as well.

The dissertation is aiming to detect all visible lanes on roads, especially focused on the two driving lanes which a car is currently driving and the two adjacent lanes which are next to the driving lanes. This is so called multi-lane detection. By successfully detecting multi-lanes on roads, it can provide multi-lane information for various safety-related applications mentioned above and for the safe drivings of autonomous vehicles as well.

Chapter 2

Related Work

A huge number of lane detection algorithms have been proposed for various applications in Advanced Driver Assistance Systems (ADAS) for intelligent vehicles, such as Lane Departure Warning Systems (LDWS), Lane Keeping Assistance Systems (LKAS), ego-vehicle localization systems, and other autonomous driving systems. As shown in Table 2.1, lane detection algorithms have been developed mainly in two ways based on their detection scope and driving situations. Detection scope can vary slightly, with algorithms detecting either driving lanes or multiple lanes. In single lane detection, detection targets are restricted to only the two driving lane marks surrounding a vehicle [5][7][9][12]. This reflects a narrow concept of lane detection and is only applicable to LDWS or LKAS. In multi-lane detection, the algorithms detect not only the driving lane where the vehicle is driving, but also the lanes adjacent to the driving lane [6][8][10][11][13]. In contrast with single lane detection algorithms, which have limited use, multi-lane detection algorithms are expected to be used in many vehicle safety-related applications.

In terms of driving situations, algorithms were initially designed for highway-driving , but were gradually developed for more complex and challenging environments, such as urban settings. In highway situations, the algorithms can be simplified due to the assumption of parallelism, which implies that lane marks are parallel and

Figure 2.1. The cases that appear often in urban environments but cannot be solved by parallelism.

have the same lane width [7][10][11][12][13]. In urban situations, on the other hand, the algorithms become more sophisticated because various factors arising from urban streets must be considered, such as frequent splitting and merging of lane marks, high curvatures, crossroad signs, and curbs [5][8][9].

In conventional approaches to multi-lane detection, most algorithms are based on several assumptions. In [11][12] the authors model the relationship between camera and road geometry with constant parameters to detect lanes efficiently. The approaches are effective in detecting multiple lanes that are located in the hypothesized areas, but the systems are not perfectly stable, as the rectified plane can be harmed easily by the vibration, tilting, and lateral movement of driving vehicles.

In [8][10][11][13] the authors assume that all lanes are parallel with each other on the road plane, making the adjacent lanes easily detectible and also eliminating false positives. In highways this assumption is valid in most cases of multiple lanes. However, in urban situations where non-parallel lanes usually appear (e.g. intersection, splitting, and merging), lane detection algorithms with parallelism assumptions can cause misdetection of the non-parallel lanes.

Table 2.1. Technology migration in the development of lane detection algorithms.

Targeting / Situation	Single Lane Detection	Multi-Lane Detection
Highways	[7][12]	[6][10][11][13], Chapter 3
Urban environments	[5][9]	[6][8], Chapter 4

2.1 Contributions of the Dissertation

In this dissertation we investigate methods of multi-lane detection that can handle urban situations as well as highway situations. Algorithms are specialized for each driving environment.

In chapther 3, the dissertation introduces a new multi-lane detection algorithm for highway driving environments. As almost all of lanes in highways are parallel to each other with keeping the same intervals, the algorithm uses the property to detect multiple lanes in highway driving environments. Even if a car blocks an adjacent lane and the lane is partially visible, the algorithm still can successfully detect the occluded lane by using parallelism. To make it possible, we generate ROIs for adjacent lanes based on the driving lane detection results. And, we densely detect each lane in each ROI.

Further, the algorithm aims to recognize the properties of the lanes, such as type and colors. To recognize the properties, we first extract the road color information and use HSV channels to specify the color differences. The algorithm are tested on several video clips with a resolution of 1024 x 768 color images which is about 27 km long.

In chapter 4, we propose a new algorithm to detect multiple lane marks on roads and to cover the various types of lanes appearing in urban streets, e.g. intersection, splitting, and merging lanes. Our approach does not require any assumptions of geometric modeling or parallelism in order to prevent the algorithm from failing due to

changing parameters, or causing misdetection of non-parallel lane cases. Instead, we utilize one of the graphical models known as Conditional Random Fields (CRFs) to robustly detect multiple lanes in various urban road situations.

Conditional Random Fields (CRFs) are widely applied in object recognition and image segmentation problems [14][15]. CRFs consist of a set of nodes and edges. Each node has a label on it, indicating the state of the node. Edges are two connected nodes that depend on each other. Unlike simple association algorithms, which determine the state of an object based solely on its observation, CRFs determine the state of an object by considering not only its observations but also the states and observations of other connected objects. Therefore, CRFs are adaptable to changing environments and are reliable as considering multiple factors. This ability of CRFs makes them suitable for use in multi-lane detection problems, where CRFs can find the best association case out of all possible cases. Because of the effectiveness of CRFs in associating multiple targets, our algorithm can overcome the shortcomings of conventional lane detection algorithms.

Chapter 3

Mutli-Lane Detection in Highway Driving Environment

3.1 Algorithm Overview

When multiple number of parallel lanes are clearly laid on road, human can recognize each of them without any difficulties. Further, even if some objects partially occlude a lane and only a small portion of the occluded lane is visible, human can still clearly recognize a lane. The reasoning of the recognition is that when there are at least two visible lanes, the other lanes which are parallel to the visible lanes can be recognized using the fact that the all lanes are parallel to each other.

With the reasoning idea, the dissertation proposes algorithm of detecting multiple lanes in highways driving environment. At first, the algorithm detects the two driving lane marks. Then, with the calibration results and parallel assumption, the ROI (Regions of Interest) for driving lanes and adjacent lanes are generated. Finally, the algorithm proceeds to detect each lane in each ROI. Especially, the algorithm scrutinizes the existence and the property of the lane in each ROI, recognizing type and color of them. Because we use ROI to detect multiple lanes, false positive rates can be dramatically reduced and true positive rates can be increased even if the some objects such as cars and trucks occlude the lanes. The details of the algorithm are

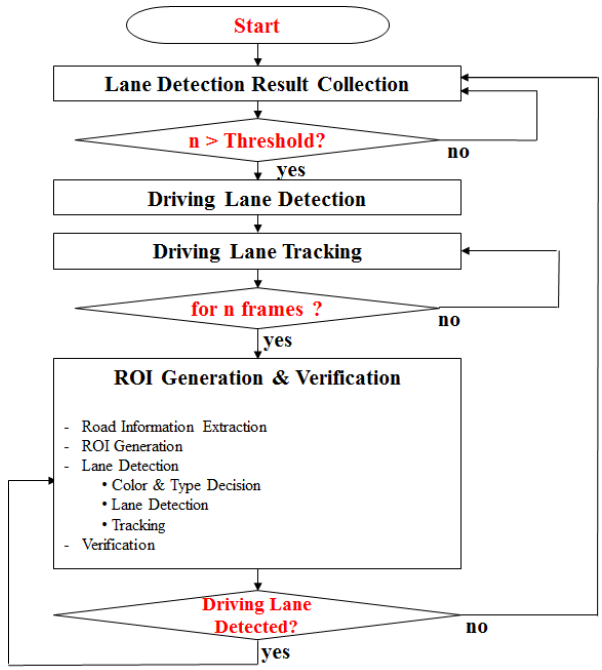


Figure 3.1. The detail flowchart of the algorithm for detecting multi-lanes in highway driving environment.

presented in Fig. 3.1 and will be discussed in details in this chapter.

Of course, the algorithm uses the parallel assumption, so it cannot detect splitting, merging and crossing lanes which sometimes happen in highway driving environment. However, these lanes are rarely exist in highways and the majority of lanes are parallel lanes, so the algorithm do not consider the situations. The recognition of the non-parallel lane situation will be covered in next chapter.

3.2 Driving Lane Detection

To properly generate the ROI for detecting multiple lanes including two driving lanes and two adjacent lanes, at least two lanes are detected accurately as being references. In the algorithm, we detect two driving lane to generate ROIs of each lane. To detect the two driving lanes accurately, we use several steps to verify that the detected

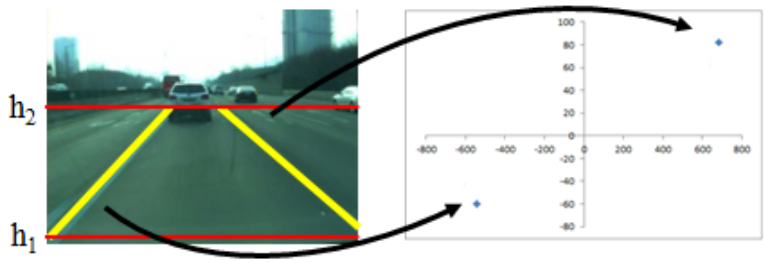


Figure 3.2. Accumulation of detected lanes into a feature space.

lanes are accurate driving lane.

3.2.1 Lane Detection

To detect the driving lane properly, we first accumulate the lane detection results for about twenty frames. In each frame, we detect two major lanes using a simple algorithm which will be introduced in detail in the next chapter. When the lanes are detected, a lane is transferred into a feature space which is defined in the dissertation. Let x_1 and x_2 is points of intersection with h_1 and h_2 which is horizontal lines in image plane. Then a lane can be transferred into a point (x_1, x_2) in a coordinate which we define. Fig. 3.2 show how to accumulate the detection results into the feature space.

After the number of lanes exceed a threshold, here we set threshold as 40, we find the two major groups in the feature space. The left image in Fig. 3.3 shows the accumlation result of lanes. To find the two major groups, we conduct sequential RANSAC (RANdom SAmple Concensus). RANSAC is well known algorithm to cluster or find inliers, while discarding outliers. The sequential RANSAC is extended version of RANSAC, and it conducts RANSAC several times to find several groups. The specific parameters such as a threshold for the number of inliers, should be adjusted for the sequential RANSAC.

By conducting the algorithm, we can get the two major groups, and each repre-

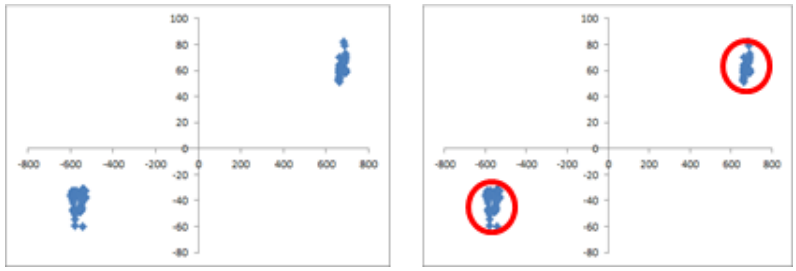


Figure 3.3. Finding two major groups using sequential RANSAC.

sentative value of a group denote the result of detected driving lane. The right image in Fig. 3.3 show the clustering results. No outliers exist in the sample image, but the grouping step can also be robustly done even if outliers exist. The found groups can be transferred into image space and be the representative detection results of two lanes for few frames. Here, we assume that the two lanes are driving lanes. Of course the assumption can have a lot of arguments, but when testing the algorithm, the two driving lanes are detected with high possibilities. This is because we collect only two major detected lanes in a frame and get the result from their accumulation.

3.2.2 Lane Verification

After finding two lanes which are driving lanes with high possibilities, we have to verify whether they are truly driving lanes or accidentally detected false positives. To verify, we use tracking and confirm that they appear in a continuous frames or not. If they are tracked for 20 frames, we confirm that they are driving lanes not the false positive one.

To track the lane, we use particle filter [17] which is well-known tracking algorithm in computer vision field. We can define particle as a set of coefficient of fitted polynomials or a set of lane features which are extracted. In both way, lane can be tracked well, but the way of using polynomials have some defects such as vibrations of tracking results. Therefore, we use a set of lane features as a particle.

3.3 Multiple Lane Detection via ROI Generation

With two given driving lanes, here, we generate ROIs for detecting other lanes, both driving lanes and adjacent lanes in later frames. Also, with the ROIs, we recognize type and color of each lane. This section introduce how to detect and recognize multiple lanes with ROIs.

3.3.1 Road Color Extraction

To successfully recognize color of each lane, we first extract the road color information. We define an area between two driving lanes as a road and extract the color information of it. In Fig. 3.3, the red-dotted area between two driving lanes are the road area, and we sample the color information in the area. By converting the RGB color image to HSV space, we obtain the value of hue and saturation.

3.3.2 ROI Generation

To generate the ROI, we first need the calibration information, which can be obtained using a checker board. And, using the parallel assumption, we can produce the bird eye's view of the image using two parallel driving lanes. By selecting two

Figure 3.4. Extracting color information of roads.

points having the same height in each lane, bird eye's view can be constructed by the total four points. Then, the location of adjacent lanes are estimated in bird eye's view and can be also obtained in image plane by transferring their location. The estimated location of adjacent lanes are not really accurate, but they are acceptable as only thing needed is generating ROI which is sufficient for detecting lanes. Around the estimated location of lanes, we set an area of 4 times of estimated lanes width as ROI. And, in vertical direction we set every 40 cm as a scan line, using calibration information. Fig. 3.5 shows the result of ROI generation. For adjacent lanes, the ROI is slightly biased to the horizontal direction, but the ROIs are sufficiently covering the lane and can possibly detect the lane.

3.3.3 Lane Detection in ROI

Before detecting lanes in each ROI, the type and color of lane should be decided so that the different algorithm for detecting each property can be conducted. At first, for one ROI, we count the number of points which are corresponding to road color and the number of points which are corresponding to color lane (e.g. orange lanes or blue lanes). To be a blue lane, the saturation value of the pixel should be higher

Figure 3.5. ROI generation.

than 20, the threshold that we set here in the algorithm. To be a orange lane, the hue value of the pixel should be around 40, which is a general property of orange color. After collecting the numbers of each color lane, we calculate the ratio of orange lane segment to road color segment. If the ratio exceeds a threshold which is about twenty percent, it can be decided that orange lane exist in the ROI. The same procedure can be done for a blue lane segment.

After deciding the existence of color lane, detecting lanes in given ROI is relatively simple than detecting lane without ROIs. As in Fig. 3.5, we use Ridge property to detect lane feature. We use two filters to find left and right edges in Fig. 4.1 and extract the lane marking feature if the a distance between left and right edges are within the estimated lane width. The more details are explained in the next chapter. To detect a orange lane, we filter not the gray image but the hue domain image which the orange color in the hue space image is more visible than in the gray image. To detect a blue lane, we filter the saturation domain image, where the saturation value of blue lane is higher than that of road image.

The lane feature extraction result are shown in Fig. 3.6. Red color denotes the detection result of orange lane, green color denotes the white lane, and the blue stripe

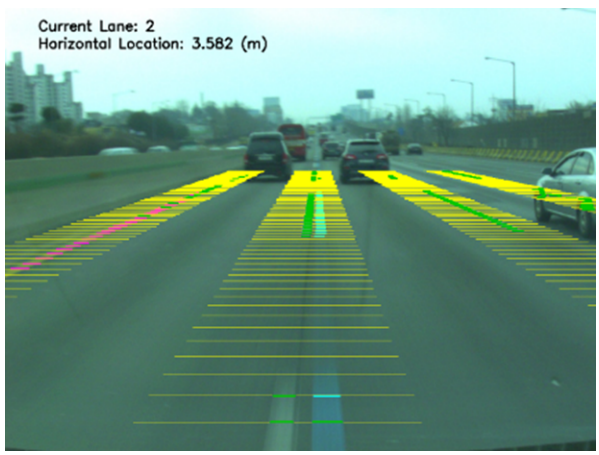


Figure 3.6. Lane Feature Extraction.

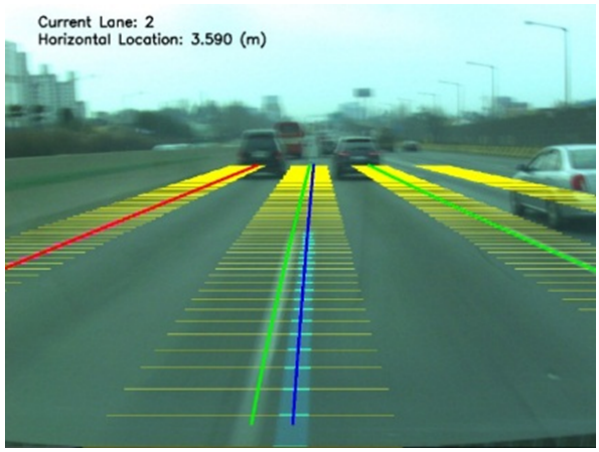


Figure 3.7. Lane Fitting Results.

corresponds to the blue lane.

With the extracted lane features, we fit the lane model on the features. Here, we use RANSAC (RANdom SAmple Consensus) to fit the third order polynomials while discarding outliers. Fig. 3.7 shows the fitting result on each ROI. The lane in fourth right ROI is not detected because of the insufficient number of lane feature.

After detecting lanes in each ROI, we again conduct the particle filters [17] to track and display the lanes safely and stably. For tracking, we use the detected lane features to track the lanes.

3.4 Experiment

To validate the algorithm, we tested on the normal PC, Intel(R) Core i5- 2400 CPU @ 3.10GHz and 4.00GB RAM. We took the several video clips in highways in the normal daily weather. The length of the video clip is about 15 minutes with 27 km distances. The video is color image with a resolution of 1024 x 768. About 26,000 frames are tested, with targeting 67,790 of driving lanes and 26,632 of adjacent lanes. To get the statistics, we only consider a lane which visible length is more than 20 meter and the lateral offset error is less then 20 cm as a target lane. Therefore, the target

Table 3.1. Experiment results of the lane detection in highway driving environment .

Measure	# of targets	# of detected	TPR	FNR
Driving Lanes	67790	62607	0.9235	0.0765
Adjacent Lanes	26632	24529	0.9210	0.0790
Overall	94422	87136	0.9228	0.0771

number of adjacent lanes are relatively lower than the target number of driving lanes. This is because of the occlusion from cars.

Table 3.1 shows the experiment results of the algorithm. For driving lanes and adjacent lanes, both detection rates are above 92 percent. Generally, if the lanes are detected, they are continuously detected by the particle filter algorithm. However, by the occlusions form cars or some disturbances, if the algorithm loose the driving lanes, system fails to detect all lanes because the ROI for detection is generated by the driving lanes. Therefore, it takes to about 30 or 40 frames to regenerate the system, and then the detection rates goes to down. If the duration of regenerating the algorithm is reduced, the overall detection rates will be improved.

Fig. 3.8 shows the detection results of multiple lanes in highway driving environment. The road is 4 lanes, and wherever the car is driving, the result shows that all target lanes are successfully detected. Also, when the cars is driving in the far-left or the far-right lane, it is very importance not to detect non-existence lane even though the ROI is generated. Sometimes but rarely, the algorithm detect the non-existence lane as a false positive, and the amount of them is not very significant. Therefore we do not include the statistics of them.

3.5 Summary

In this chapter, we introduce the algorithm for detecting multiple lanes in highway driving environment. By detecting two driving lanes and detecting other lanes with

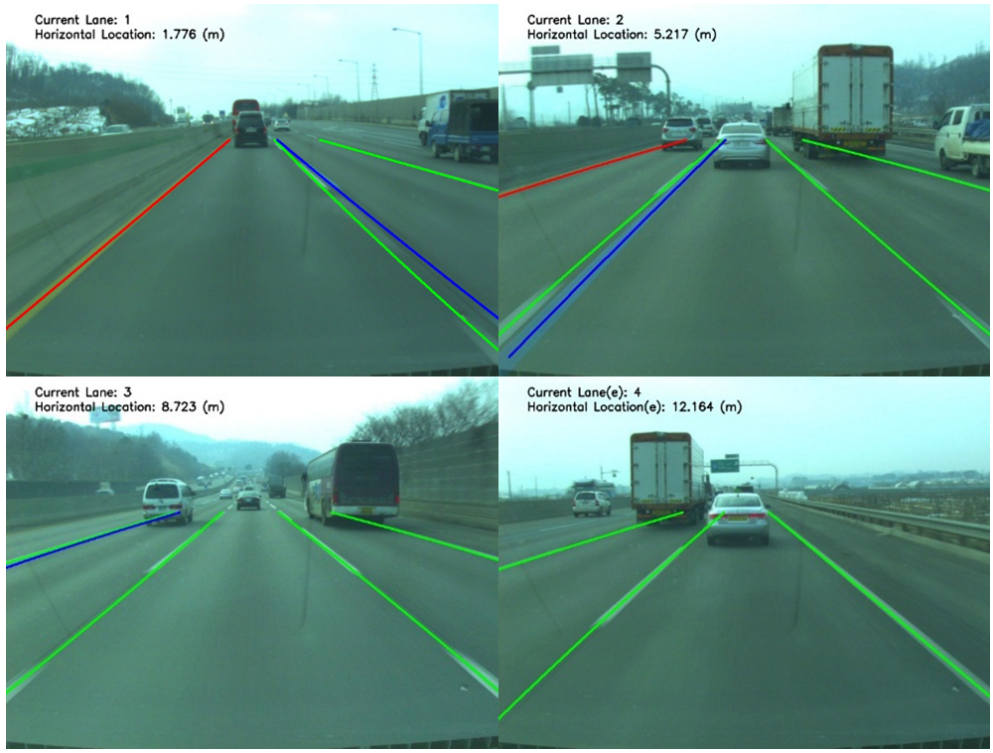


Figure 3.8. Detection results of the lane detection in highway driving environment.

generated ROI, the algorithm can safely and stably detect the multiple parallel lanes in highways. Also, the algorithm are aiming to recognize all of them, differencing colors and types. The overall detection rates is 0.923 and can be more improved if some defects are revised such as the delay of regenerating the system and the parameter optimization.

Chapter 4

Mutli-Lane Detection in Urban Driving Environment

4.1 Algorithm Overview

In urban driving situations, it is dangerous to use the parallelism assumption for detecting lanes because non-parallel lanes such as splitting, merging and crossing often appear on roads. In this chapter, we introduce a new lane detection algorithm which targets detecting these kinds of lanes as well.

To detect the non-parallel lanes, we mimic the reasoning of detecting complicated lanes, which we regard a white marking that spreads to a one direction as a lane. First, we extract lane marking features using edge filters. Then, we approach to detect lanes as a segmentation problem. The goal become to properly cluster the features in right groups. As a traditional segmentation problem, we generate a supermarking, which is a set of lane features, likely a superpixel. Then, using CRFs (Conditional Random Fields), we associate the supermarkings to be a lane. After associating the supermarking, we need to verify whether the clustered group is a real lane or not. With a few validation process, the final results of detecting lane can be obtained.

4.2 Lane Feature Extraction

Lane feature extraction mainly consists of two steps: lane feature detection and supermarking generation. First, we extract lane features by finding strong gradient values between the sets of neighboring pixels. Then we generate the *supermarkings*, which is a term we introduce in this dissertation. A supermarking is a cluster of neighboring lane features that are detected at the lane-feature-detection step.

4.2.1 Lane Feature Detection

As in a number of studies [1][2], we detect lane features by using the ridge property [7]. Similar to [2] and as shown in Fig. 4.1, we use two filters f_l, f_r to detect left and right edges by scanning line-by-line in an input image $I(x, y)$. The filter width w_f is predetermined by considering the perspective effect on the image coordinates.

We obtain the filtered images $I_l(x, y), I_r(x, y)$, and the images are thresholded by a function $T(\cdot)$ that gives variable thresholds according to the input intensity. As the input intensity gets lower, the function gives a lower threshold value. This function is used to make sure that the lane features can be robustly extracted regardless of

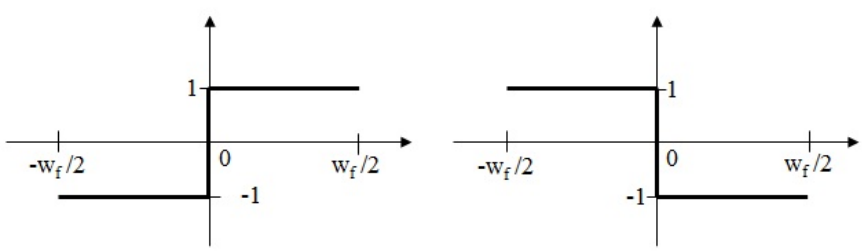


Figure 4.1. Filters that detecting left edges and right edges in the input images respectively.

any change in lighting situations.

$$I'_l(x,y) = \begin{cases} I_l(x,y) & \text{, if } I_l(x,y) > T(\frac{\sum_{i=x-w/2}^{x} I(i,y)}{w/2}) \\ 0 & \text{, otherwise} \end{cases}$$
$$I'_r(x,y) = \begin{cases} I_r(x,y) & \text{, if } I_r(x,y) > T(\frac{\sum_{i=x}^{x+w/2} I(i,y)}{w/2}) \\ 0 & \text{, otherwise} \end{cases} \quad (4.1)$$

With the two thresholded images, a lane feature $\mathbf{x_i} = (x_i, y_i)$, which is a vector denoting the location of the feature, can be extracted by finding a local maximum pair from each image. To be a lane feature, the interval of the two local maxima should be laid between the minimum estimated-lane-feature-width w_m and the maximum estimated-lane-feature-width w_M. Just as the filter width w_f, w_m and w_M are also preset by considering the perspective effect.

Algorithm 1 (Generating Supermarkings)

Input: The lane feature $\mathbf{x_i}$, N_x

$N_s \leftarrow 0$
For: i $\leftarrow$ 1 to N_x
  For: j $\leftarrow$ 1 to N_s
    If $dist(\mathbf{x_i}, \mathbf{x_{s_j}}, \theta_{s_j}) < \phi$
      $s_j \leftarrow \mathbf{x_i}$ and Update $\mathbf{x_{s_j}}, \theta_{s_j}$
    end **If**
  end **For**
  If $\mathbf{x_i}$ is not included in any s_j
    $s_{N_s+1} \leftarrow \mathbf{x_i}$
    $N_s \leftarrow N_s + 1$
  end **If**
end **For**

Output: The supermarking s_j, N_s

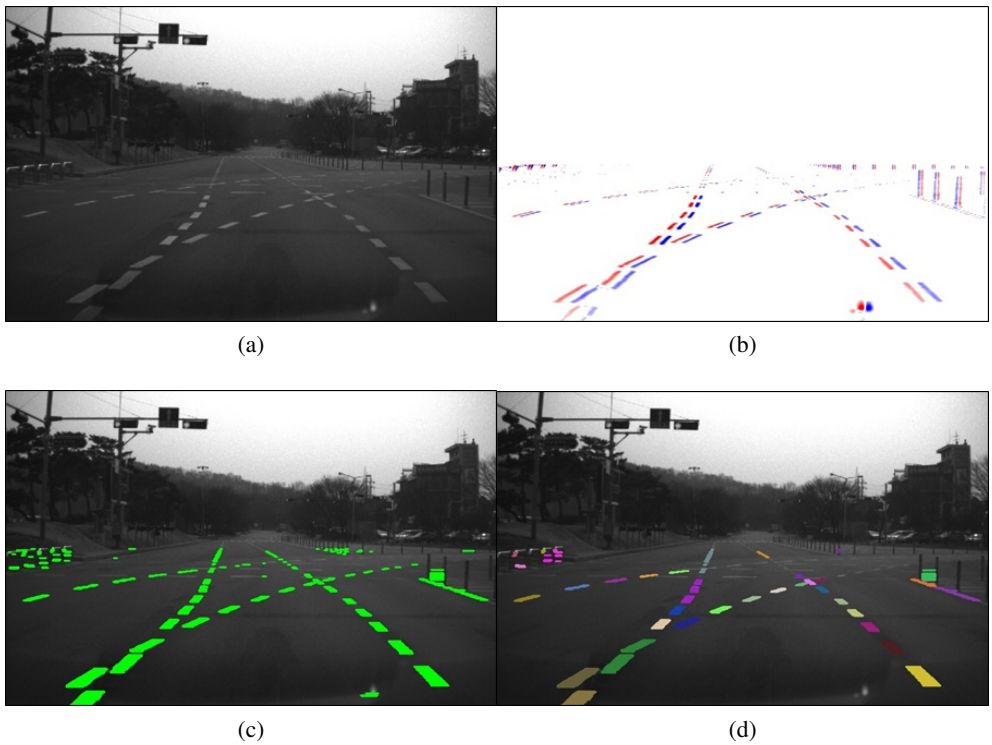
4.2.2 Supermarking Generation

The concept of the *superpixel* is commonly applied in image segmentation problems [16]. By forming a set of adjacent pixels having similar attributes into a superpixel, segmentation problems can be solved more effectively. The superpixel effectively reduces the complexity of a problem by requiring the processing of only a few superpixels instead of hundreds of thousands of pixels.

Similarly, lane features that are assumed to belong to the same lane marking can be clustered into a small group, thereby forming a *supermarking*. To be a supermarking, lane features should satisfy the following conditions: the features should be adjoined to each other and the features should have the same direction. Algorithm 1 presents the method for generating the supermarking sj which is a set of lane features.

In Algorithm 1, $\mathbf{x_{s_j}}$ and θ_{s_j} denote the location and direction angle of the supermarking s_j in the image coordinates. The values are updated when a new lane feature is included in the supermarking. The function $dist(\mathbf{x_i}, \mathbf{x_{s_j}}, \theta_{s_j})$ calculates the geometric distance between an input feature $\mathbf{x_i}$ and a supermarking s_j. When the distance is lower than a threshold ϕ, which is found empirically, the lane feature $\mathbf{x_i}$ gets included in the supermarking s_j. If the input feature is not included in any supermarkings, the feature creates a new supermarking. Although the process is conducted sequentially and not globally, the supermarkings can still be generated successfully as the input lane features are sorted in ascending order in the image coordinates.

An example of generated supermarkings is shown in Fig. 4.2d. After all supermarkings are generated, as in Fig. 4.3, each supermarking is described by a set of coefficients, $\mathbf{h_i}, \mathbf{c_i}, \mathbf{t_i}, \theta_{h_i}, \theta_{c_i}, \theta_{t_i}$, denoting the head position, the center position, the tail position, the head direction angle, the center direction angle and the tail direction angle of the supermarking s_i, respectively. The calculation is done in an image plane



(a) (b)

(c) (d)

Figure 4.2. The lane marking extraction and the supermarking generation result. (a) The input image, (b) The thresholded filtered image, (c) The extracted lane markings, (d) and the generated supermarkings (right).

without calibration, but the algorithm can be more generalizable and reliable if done in world coordinates.

4.3 CRF Modeling for Supermarking Association

With the given supermarkings, the goal turns into appropriately associating the supermarkings belonging to the same lane mark. Here, we adopt the Conditional Random Field (CRF), which is an undirected graphical model widely applied in object recognition and image segmentation problems. We demonstrate how to properly associate the supermarkings by formulating a CRF graph, modeling unary and pairwise terms, and minimizing the energy of the graphical model. The result becomes a

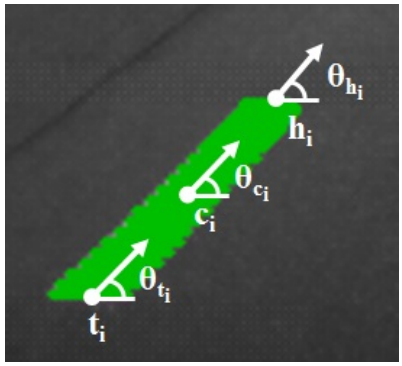


Figure 4.3. An example of a supermarking.

final set of supermarkings to which lane models are ready to be fitted.

4.3.1 Low Level Association

Similar to [3][4], we first conduct low-level association to group the supermarkings that are most obviously associated, thereby reducing the problem size. We compose the association probability matrix $\mathbf{M}$ with a size of n-by-n where n denotes the number of supermarkings. Each cell in $\mathbf{M}$ has the association probability of two supermarkings corresponding to the row and column index of the cell. For $i \neq j$, the probability is calculated by (4.2) and filtered by a threshold for choosing only solid associations. Here, we set the threshold as $0.975/z$ and σ as 0.25.

$$P(s_i, s_j) = \frac{1}{z} \exp(-\frac{|\theta_{c_i} - \theta_{c_i c_j}|^2 + |\theta_{c_j} - \theta_{c_i c_j}|^2}{\sigma^2}) \tag{4.2}$$

θ_{c_i} and θ_{c_j} are the center direction angle of each supermarking, and $\theta_{c_i c_j}$ is the direction angle of the vector $\mathbf{c_i} - \mathbf{c_j}$ which is the difference between the center position of supermarking i and j. In order to be associated at a low level, supermarkings s_i and s_j should satisfy the following conditions:

- The supermarkings should not overlap with each other in the y-axis in the image coordinates.

- s_i is only associable (only one non-zero value in a column corresponding to s_i in matrix $\mathbf{M}$) with s_j, and the association probability of s_i and s_j is the highest among the all associable cases of s_j with other supermarkings.

4.3.2 CRF Graph Formulation

Given a set of supermarkings, we construct a CRF graph $G = (V, E)$, where each vertex indicates a possible association of two supermarkings, and an edge connecting vertices denotes the existence of correlation among them. To determine the vertices, as shown in Fig. 4.4, we devise two measurement functions $dist_{geo}$ and $dist_{dir}$ that calculate the geometric distance and directionality of two supermarkings:

$$\begin{aligned} dist_{geo}(s_i, s_j) =& |(x_{t_j} - x_{h_i})sin\theta_{h_i} - (y_{t_j} - y_{h_i})cos\theta_{h_i}| \\ &+ |(x_{t_j} - x_{h_i})sin\theta_{t_j} - (y_{t_j} - y_{h_i})cos\theta_{t_j}| \\ dist_{dir}(s_i, s_j) =& |\theta_{t_j} - \theta_{h_i t_j}| + |\theta_{h_i} - \theta_{h_i t_j}| \end{aligned} \tag{4.3}$$

where h_i is the head of the supermarking s_i, t_j is the tail of the supermarking s_j, and $\theta_{h_i t_j}$ denotes the slope angle of a line made by h_i and t_j.

The two supermarkings are said to be associable and become a vertex $v_k = (s_i \rightarrow s_j)$ if both $dist_{geo}$ and $dist_{dir}$ are under thresholds that are roughly set to contain all positive samples from real data sets. Edges in the CRF graph can be made among all of the vertices that share at least one of the same supermarkings in the vertices through any hops in the graph, making a subset of vertices exactly as a clique.

Each vertex v_k in the V = $\{v_1, \cdots, v_K\}$ has a label l_k which can be 0 or 1. The label is 1 if the supermarkings are associated, or 0 if not. The label set L = $\{l_k, \cdots, l_K\}$ functions as states of supermarking associations during the energy minimization process, and the optimal set indicates the results of the association. In the meantime, the labels have constraints that 1) a head of a supermarking should be connected to

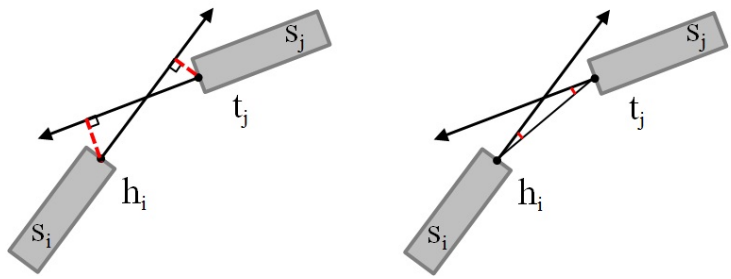


Figure 4.4. (a) The geometric distance measurement function $dist_{geo}$, (b) The directionality measurement function $dist_{dir}$.

at most one tail of another supermarkings and 2) a tail of a supermarking should be connected to at most one head from another supermarking. The constraints can be conceptually understandable, as lanes on roads should not be combined with each other while keeping their own directions. As in [4], the following equation shows the constraints:

$$\sum_{v_k \in H_i} l_k \leq 1, \text{where } H_i = \{v_i | (s_i \rightarrow s_j) \in V\} \; \forall s_j \in S$$
$$\sum_{v_k \in T_i} l_k \leq 1, \text{where } T_i = \{v_i | (s_j \rightarrow s_i) \in V\} \; \forall s_j \in S \tag{4.4}$$

H_i denotes a set of vertices to which the head of s_i is connected, and T_i is a set of vertices that are connected to the tail of s_i.

4.3.3 Modeling Unary and Pairwise Terms

In the CRF graph, the energy of the graph is depicted as the composition of unary terms and pairwise terms. A unary term of a vertex expresses its own probability, and a pairwise term implicates the probability of the relationship between vertices

connected by edges.

$$
\begin{aligned}
P(L|S) &= \frac{1}{z}\exp(-\Psi(L|S)) \\
\Psi(L|S) &= \sum_{l_k \in L} U(l_k|s_k) + \sum_{C \in cl(G)} \phi_C(l_C|s_C)
\end{aligned} \tag{4.5}
$$

In (4.5), $P(L|S)$ denotes the whole probability of a label set, and $\Phi(L|S)$ is the energy of the graph, consisting of the unary potentials $U(l|s)$ and clique potentials $\phi(l|s)$, which correspond to pairwise potentials in a general CRF model. C is a clique among the clique set $cl(G)$ in graph G.

We design the unary terms using the distance functions defined in (4.3), showing how likely the two supermarkings are to be associable. Here, we use a quadratic regression model to determine the unary potentials.

$$
\begin{aligned}
U(l_k|s_k) &= -ln(P(l_k|s_k)) \\
P(l_k|s_k) &= \frac{1}{1+\exp(-f_{dist}/\sigma)} \\
f_{dist} &= \begin{bmatrix} d_g \\ d_d \\ 1 \end{bmatrix}^T \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix} \begin{bmatrix} d_g \\ d_d \\ 1 \end{bmatrix} \\
d_g &= dist_{geo}(s_{k_1}, s_{k_2}) \\
d_d &= dist_{dir}(s_{k_1}, s_{k_2})
\end{aligned} \tag{4.6}
$$

The coefficients in the quadratic regression model are trained from a number of positive and negative association samples of supermarkings collected from sample video clips.

To model the clique potentials, we build a linear regression model to measure the probability of each clique. For each clique, a lane model in a second order polynomial is fitted on a group of supermarkings from vertices that are labeled 1 in the clique. We calculate the average algebraic error between the lane model and all of the

lane features in the supermarkings. Then the potential of a clique can be calculated by the following equation:

$$\begin{aligned} \phi_C(l_C|s_C) &= -ln(P(l_C|s_C)) \\ P(l_C|s_C) &= \frac{1}{1+\exp(-(\phi - f_{err})/\sigma)} \\ C \in cl(G), f_{err} &= \sum_{k=1}^{K(C)} |y_k - f_C(x_k)|^2/K(C) \end{aligned} \tag{4.7}$$

In (4.7), f_{err} indicates the average errors between the lane model f_C and the lane features $\mathbf{x_k}$. The ϕ and σ are the center and the degree of steepness of the regression curve, which are determined by a training. $K(C)$ is the number of lane features from 1-labeled supermarkings in a clique C.

4.3.4 Energy Minimization

With all defined unary and clique potentials, the energy of the graph can be obtained. Then the best label set L^* can be determined by minimizing the energy of the graph.

$$\begin{aligned} L^* &= \arg\min_L \Psi(L|S) \\ &= \arg\min_L \sum_{l_k \in L} U(l_k|s_k) + \sum_{C \in cl(G)} \phi_C(l_C|s_C) \\ &= \arg\max_L \sum_{l_k \in L} ln(P(l_k|s_k)) + \sum_{C \in cl(G)} ln(P(l_C|s_C)) \end{aligned} \tag{4.8}$$

The complexity of the problem is O(n^4), which is high due to the large search space. It does not critically affect the run time performance, however, because the computational cost is significantly reduced through the low level association with the supermarkings. On average, less than 10 possible association cases are found and

(a) (b)

Figure 4.5. The supermarking association result: (a) The low level association result, (b) The final association result after the energy minimization.

examined in one frame.

Through energy minimization, the final association result between supermarkings is produced. Out of all generated final supermarkings, only those that satisfy certain criteria of length, size and curvature become final candidates to be recognized as lanes. Then, the lane model is fitted on the final lane candidates and displayed.

4.4 Experiment

The algorithm was tested on the normal PC, Intel(R) Core i5- 2400 CPU @ 3.10GHz and 4.00GB RAM. We tested several video clips with a resolution of 752 x 480, taken in urban streets in the normal daily weather. A total of 2064 frames were tested, and the number of target lanes was 7609. For testing targets, we only considered a lane whose length was longer than about 10m among all detectable four lanes and non-parallel lanes. The numbers only showed the detection results without any tracking algorithms.

As various disturbances exist in urban streets, such as road marking, car, and other noises, we calculate statistics in three categories: the case in which disturbances exist, the case in which disturbances do not exist, and the combined cases. A

true positive rate (TPR) and a false positive rate (FPR) are calculated as TPR = (the number of detected lanes) / (the number of target lanes), FPR = (the number of false positives) / (the number of target lanes).

The algorithm runs at the rate of 30.2 frames per second. As shown in Table 4.1, our algorithm works nearly perfectly in clean situations, but the performance is relatively degraded when affected by disturbances from cars or road markings. Table 4.2 shows how the algorithm deals with non-parallel lane cases. A total of 471 non-parallel lanes are found and detected at the success rate of 0.915. It is shown that it is possible to detect non-parallel cases successfully with the proposed algorithm and much higher detection rates can be achieved when tracking algorithms are applied.

The overall detection results are demonstrated in Fig. 4.6. The first and second rows show that the target lanes are detected well in both situations. In the third row, the algorithm also successfully detects various non-parallel lanes. In the fourth row, however, false positives can be generated if there are heavy disturbances on the roads.

To check the validity of our algorithm, we compare our results with M.Aly's [8], which also aims to detect multi-lanes in urban environments. Given the Caltech Lane Datasets [18], we test both algorithms. In the datasets, we consider all visible lanes as target lanes, and only the detected lanes, which have a small displacement and the same direction with the ground truth, are counted as true positives. As the criteria differ, our statistics of M.Aly's are slightly different from the published results [8]. Because of a lack of non-parallel cases in the datasets, the results do not show the

Table 4.1. Multi-lane detection results in urban situations.

Measure	TPR			FPR	FP/frame
Type	Driving	Adjacent	Overall	Overall	Overall
Clean	0.995	0.964	0.980	0.008	0.030
Disturbance	0.908	0.865	0.887	0.126	0.437
Overall	0.966	0.930	0.949	0.048	0.177

Table 4.2. Comparison of parallel and non-parallel lane cases.

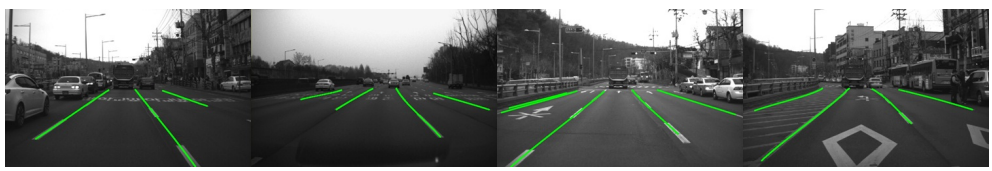
Measure	# of targets	# of detected	TPR	FNR
Parallel lanes	7138	6787	0.951	0.049
Non-parallel lanes	471	431	0.915	0.085
Overall	7609	7218	0.949	0.051

Table 4.3. Comparison with M.Aly [8].

Scene	M.Aly [8]			Our Method		
	TPR	FPR	FP/Frame	TPR	FPR	FP/Frame
1	0.813	0.099	0.384	0.892	0.125	0.488
2	0.839	0.224	0.672	0.865	0.209	0.628
3	0.934	0.148	0.542	0.850	0.111	0.408
4	0.890	0.102	0.418	0.898	0.063	0.259
Overall	0.871	0.148	0.529	0.874	0.131	0.469

strong points of our algorithm, but we still achieves a slightly better performance even without parallelism, as shown in Table 4.3.

(a) The cases without disturbances.



(b) The cases with disturbances.

(c) The cases where non-parallel lanes exist.

(d) The cases of false positives.

Figure 4.6. The multi-lane detection results using the Conditional Random Fields (CRFs) in urban streets.

Chapter 5

Conclusion

This dissertation introduces a new approach to multi-lane detection both in highway and urban driving situations. For the highways, the algorithm basically use parallelism to ease detecting multiple parallel lanes. Further the algorithm recognize each properties of lanes so that it can be applied in various vehicle safety-related applications, not only LKAS (Lane Keeping Assistance System), LCW (Lane Changing Warning) but also vehicle localization by detecting central yellow lane.

However, there is an issue that the detection rates are degraded when a car blocks driving lanes and the whole system begins to restart. To minimize this situation, the algorithms needs to be impoved and can robustly cover the various risky situations for the system. Also, if the situation is not inevitable, the system should reduce the delay for being regenerated. If the problems are resolved, the system will be more robust to adopt in real-time applications.

For the urban driving situations, the algorithm can detect various kinds of lanes using CRFs without any assumptions of lanes or road models. Even without the parallelism assumption, the algorithm still successfully detects not only simple parallel cases but also non-parallel cases appearing in urban streets, such as splitting, merging and crossing lanes. The experiment results shows that the system operates well in clean situations, but the detection rates are degraded when disturbances exists on

roads. Also, the false positive rates are dramaticall increased. This is because we did not adapt a tracking algorithm such as Kalman Filter or Particle Filter. If the tracking algorithms are applied, then the experiment results are surely improved, showing good statistics.

Finally, the both algorithms are seperated for each driving environment. For the completion and the integration of the algortihms, a new algorithm which can operate well in both driving environment should be developed by integrating both algorithms suggested here. Then, regardless of the driving situations, the algorithm can successfully detect and recognize all non-parallel lanes as well as parallel lanes existing on roads. That algorithm, then, can be the ultimate lane detection algorithm that can be applied in various intelligent vehicle applications.

Bibliography

[1] E. Pollard, D. Gruyer, J.-P. Tarel, Ieng Sio-Song, and A. Cord, "Lane marking extraction with combination strategy and comparative evaluation on synthetic and camera images," *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on* , pp.1741-1746, 5-7 Oct. 2011

[2] T. Veit, J.-P. Tarel, P. Nicolle, and P. Charbonnier, "Evaluation of Road Marking Feature Extraction," *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pp.174-181, 12-15 Oct. 2008

[3] C-H Kuo, C. Huang, and R. Nevatia, "Multi-target tracking by on-line learned discriminative appearance models,"*Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp.685-692, 13-18 June 2010

[4] B. Yang and R. Nevatia, "An online learned CRF model for multi-target tracking," *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp.2034-2041, 16-21 June 2012

[5] S. Zhou, Y. Jiang, J. Xi, J. Gong, G. Xiong, and H. Chen, "A novel lane detection based on geometrical model and Gabor filter," *Intelligent Vehicles Symposium (IV), 2010 IEEE* , pp.59-64, 21-24 June 2010

[6] K. Zhaom, M. Meuter, C. Nunn, D. Muller, S. Muller-Schneiders, and J. Pauli, "A novel multi-lane detection and tracking system," *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp.1084-1089, 3-7 June 2012

[7] A. Lopez, C. Canero, J. Serrat, J. Saludes, F. Lumbreras, and T. Graf, "Detection of lane markings based on ridgeness and RANSAC,"*Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pp. 254- 259, 13-15 Sept. 2005

[8] M. Aly, "Real time detection of lane markers in urban streets," *Intelligent Vehicles Symposium, 2008 IEEE*, pp.7-12, 4-6 June 2008

[9] Z. Kim, "Robust Lane Detection and Tracking in Challenging Scenarios," *Intelligent Transportation Systems, IEEE Transactions on*, vol.9, no.1, pp.16-26, March 2008

[10] Y. Jiang, F. Gao, and G. Xu, "Computer vision-based multiple-lane detection on straight road and in a curve," *Image Analysis and Signal Processing (IASP), 2010 International Conference on*, pp.114-117, 9-11 April 2010

[11] M. Nieto, L. Salgado, F. Jaureguizar, and J. Arrospide, "Robust multiple lane road modeling based on perspective analysis,"*Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pp.2396-2399, 12-15 Oct. 2008

[12] C.-W. Lin, D.-C. Tseng, H.-Y. Wang, "A Robust Lane Detection and Verification Method for Intelligent Vehicles,"*Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on*, vol.1, pp.521-524, 21-22 Nov. 2009

[13] H. Deusch, J. Wiest, S. Reuter, M. Szczot, M. Konrad, K. Dietmayer, "A random finite set approach to multiple lane detection," *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on* , pp.270-275, 16-19 Sept. 2012

[14] H. M. Wallach, "Conditional random fields: An introduction," Technical Report MS-CIS-04-21, Department of Computer and Information Science, University of Pennsylvania, 2004.

[15] C. Sutton and A. McCallum, “An Introduction to Conditional Random Fields for Relational Learning,” *Introduction to Statistical Relational Learning*, MIT Press, 2007.

[16] X. Ren and J. Malik, “Learning a classification model for segmentation,” *Proc. 9th Int. Conf. Computer Vision*, volume 1, pages 10-17, 2003.

[17] Isard, Michael, and Andrew Blake. ”Condensationconditional density propagation for visual tracking.” International journal of computer vision 29.1 (1998): 5-28.

[18] http://vision.caltech.edu/malaa/research/lane-detection/

국문초록

지난 몇 십년간, 자동차 안전주행 관련 기술을 위한 다중차선 인식 알고리즘의 중요도가 부각되어왔다. 본 논문에서는 고속 도로 및 시내 주행환경에서의 다중차선 인식 알고리즘에 대해 각각 제안을 한다. 이들 알고리즘은 기본적으로 자차가 주행하고 있는 주행차선 및 측면의 주행차선을 인지하는 것을 목표로 한다. 고속도로 주행환경에서의 다중차선 인식 알고리즘의 경우, 주행차선을 안정적으로 인식한 후 차선의 평행조건을 활용하여 주변차선의 관심영역을 발생하고 인식한다. 이 때, 각 차선들의 종류 및 색상을 추가적으로 인지한다. 한편, 시내 주행환경에서의 다중차선인식 알고리즘도 주행 및 주변차선을 기본적으로 인지하는데, 분기, 병합, 교차로 등 차선이 평행하지 않은 경우도 인식을 목표로 한다. 이러한 어려운 상황에서도 차선을 성공적으로 인지하기 위해, 그래픽 모델 중의 하나인 CRF(Conditional Random Field)를 도입하여 복잡한 상황에서의 차선도 인지를 가능하게 하였다. 자체 촬영 영상 및 Caltech 영상을 활용하여 고속도로 및 시내 주행상황에서 각 알고리즘이 성공적으로 동작한다는 것을 검증하였다.

주요어 : 지능형자동차, 차량용비젼, 차선인식
학 번 : 2011-23389