

# Robust Accurate Lane Detection and Tracking for Automated Rubber-Tired Gantries in a Container Terminal

Yunjian Feng and Jun Li, *Senior Member, IEEE*

**Abstract**—Lane detection and tracking technique is the autonomous driving basis for Rubber-Tired Gantries (RTGs), vital to the automation and intelligence of man-driven container terminals. However, the existing lane detection methods developed for common road scenarios cannot meet the high-precision and robust all-weather requirements of RTG autonomous driving. To solve this problem, we propose an Adaptive Edge-based Lane Detection and Tracking (AELD) method considering RTG lanes' characteristics in this paper. First, the candidate edges of lane lines are detected and paired based on the enhanced gradient features. Next, inverse perspective mapping (IPM) is employed to search the right edges, followed by an adaptive sliding-window method. Ultimately, we develop an adaptive Kalman filter to track lane lines robustly, detecting confidence weighting by relaxing the constraint of lane line width. The proposed method is tested in a real container yard, the centerline's average position error is 2.051 pixels, and the detection success rate is close to 100%.

**Index Terms**—Container terminal, Lane detection, Lane tracking, Kalman filter, Rubber-Tired Gantry.

## I. INTRODUCTION

**R**UBBER-tired gantries (RTG) are core handling equipment in manual container terminals [1]. Driving RTG manually is hard and experience-dependent work for humans, and achieving RTG's autonomous driving is one of the highest priorities for the automation and intelligence upgrading of manual container terminals. As the basis for autonomous driving of RTG in container terminals, robust lane line detection and tracking technology with high precision suitable for all-weather container yard scenarios are urgently desired. An RTG in a real container yard scene is illustrated in Fig. 1. It is mounted with four surveillance cameras on its uprights 3~5 meters above the ground. Each camera's optical axis is tilted 45°. Fig. 1(b) illustrates the image of lanes in a container yard captured by one of the installed cameras. The lane lines themselves and the spacing between them have fixed widths. Each side wheel of RTG has to move between the two yellow lane lines or along the assumed centerline of the lane. The lanes in container yards are far narrower than the regular lanes for cars, and a slight deviation may cause RTG to veer from the lane and result in a collision with containers.



Fig. 1. An RTG in a real container yard scene, including (a) an RTG mounted with cameras and (b) lane lines in an actual container yard.

Therefore, higher-precision lane detection and tracking are required for RTG autonomous driving. Specifically, lane detection's position and azimuth errors must be less than 1 cm and 0.5°, respectively. In the actual scenes, many interferences significantly challenge lane detection and tracking reliability. For example, lane lines may be subject to smudges, occlusions, and breakages due to oil pollution, surface water, and wear. Changes in weather and light conditions are also non-negligible factors. Moreover, shadows, cracks, and other marks on the ground make the detection task more difficult. As a result, existing lane detection methods for autonomous vehicles do not apply well to RTGs for container yards' characteristics and high requirements.

In the recent two decades, lane detection algorithms have been extensively studied, and the existing methods can be divided into image-processing and deep-learning-based ones. Among them, the methods based on deep learning rely on big datasets with ground truth. Unfortunately, it is too expensive to make a lane dataset containing numerous samples under different weather and lighting conditions with accurate annotations. Meanwhile, the available open-source lane datasets are almost related to ordinary roads for cars and do not simply apply to lane detection and tracking for RTG autonomous driving.

In comparison, image-processing-based methods do not require too many image samples to train models and allow

\* This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFF0500904, and Shenzhen Fundamental Research Program under Grant JCYJ20190813152401690, and Qingdao New Qianwan Container Terminal (QQCTN). (Corresponding author: Jun Li.)

Y. Feng and J. Li are with the Ministry of Education Key Laboratory of Measurement and Control of CSE, Southeast University, Nanjing 210096, China (e-mail: fengyunjian@seu.edu.cn; j.li@seu.edu.cn).

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

explicitly introducing more constraints into the detection and tracking. But the existing methods designed for ordinary roads cannot meet the precision and robustness requirements of RTG autonomous driving in the container yard. This paper proposes a low-cost, high-precision, and robust vision-based method for lane detection and tracking in a container terminal. It has vast application potential for RTG's assistance driving, remote control, and autonomous driving.

The main contributions of this paper include:

1) Proposing an Adaptive Edge-based Lane Detection (AELD) algorithm. Adaptive histogram equalization [2] is adopted to enhance the gradient change at the edge of lane lines, significantly improving the edge detection effect under complex lighting conditions. In addition, an adaptive sliding window method considering structural constraints is proposed to implement the problematic detection of defiled and interrupted lane lines. As a result, all-weather robust lane detection can be attained.

2) Designing an adaptive Kalman filter based on detection confidence weighting. The inter-frame continuity of image sequences and the width constraint of lane lines are fully exploited to adjust the Kalman gain coefficient dynamically. This method dramatically improves the success rate of lane detection and the location precision of centerlines and achieves stable lane tracking.

3) Conducting extensive experiments on several datasets and tests in a practical container yard. The results show that our method is suitable for complex lighting and weather conditions, e.g., daytime, nighttime, sunny and rainy days, and other interference factors like shadows, smudges, and occlusions. Overall, the proposed method significantly outperforms the existing methods regarding lane detection success rate and accuracy. It can meet the high precision and all-weather requirements of RTG autonomous driving in container terminals.

The rest of this paper is organized as follows. First, section II reviews the related work on lane detection based on image processing and deep learning. Next, we propose the AELD method in Section III and conduct experiments and tests in Section IV. Finally, Section V concludes the paper.

## II. RELATED WORK

There are few studies on lane detection and tracking for RTG autonomous driving in container yards. The most related work is the lane detection of ordinary roads, which generally refers to public roads, such as urban roads, highways, etc. Two categories of the work are overviewed, including image-processing and deep-learning-based methods.

### A. Image-Processing-Based Methods

The existing image-processing-based lane detection methods rely on handcrafted features, e.g., color [3]-[5], and edge [6]. Generally, color features can be enhanced by converting into color space of HLS, LAB, and YUV for illumination-robust lane detection. Yoo *et al.* [6] designed a gradient-enhancing method to convert a color image into a grayscale one to obtain

the maximum lane gradient. Gaikwad *et al.* [7] used a piecewise linear stretching function (PLSF) to improve the contrast level of the region of interest (ROI). As for edge features, edge extractors of Canny [8] and Sobel [9] are extensively applied in the edge detection of lane lines [3], [5], [6]. In [4], a new edge detection scheme, EDlines, was developed to segment line edges from the grayscale image. Given the color and edge features, inverse perspective mapping (IPM) can generate an image's bird's eye view for facilitating lane detection [10], [11]. Moreover, Hough Transform (HT) [12], Random Sample Consensus (RANSAC) [13], and sliding window [3], [5] are adopted as post-processing to fit lane lines [14]-[17]. Considering the characteristics of lane lines, extra constraint information, e.g., vanishing point [18], is introduced to improve the accuracy of lane detection. For example, Luo *et al.* [16] took five complementary constraints into a Hough transform-based unified framework to remove most fake lane candidates. On the other hand, the Kalman filter [4], [19] and particle filter [20], [21] are commonly utilized to track lane lines based on previous predictions and current measurements. Recently, some novel lane detection strategies were presented by researchers, e.g., template matching [14], dynamic programming [16], and graph-embedding-based methods [22]. More comprehensive literature reviews on traditional image-processing-based lane detection methods can be found in [23] and [24].

The image-processing-based lane detection has been extensively and intensely studied, and its effect has been sufficiently verified in road scenarios. However, the existing approaches do not aim at lane detection of RTG in a container yard. Thus, they cannot achieve detection robustness and accuracy or even fail to detect RTG lanes. Our prior work [5] has attempted to solve the lane detection problem of RTG. Still, it cannot be suitable for complex container yard scenes because the color and edge features adopted are not robust enough. Given the advantages of image-processing approaches, we propose a new robust method for challenging RTG's lane detection and tracking by exploiting the lanes' structural constraints.

### B. Deep-Learning-Based Methods

Early deep-learning-based methods detect lane lines through segmentation. Neven *et al.* [25] designed a branched, multi-task architecture called LaneNet to achieve pixel-wise embeddings and binary lane segmentation. Then, every lane instance is distinguished by clustering with the embedding features. Ko *et al.* [26] proposed a lane-detection network called a Point Instance Network (PINet) to estimate key points and segment instances to alleviate the redundancy of pixel-wise prediction. Remarkably, PINet needs only a few key points to locate lane lines. In another network, Vanishing Point Guided Network (VPGNet) [27], a vanishing point is utilized as a global geometric context to guide lane detection under adverse weather conditions.

Moreover, spatial information plays a vital role in lane detection and segmentation. For example, a spatial convolutional neural network (SCNN) was adopted in [28] and generalized traditional layer-by-layer convolutions to slice-by-

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

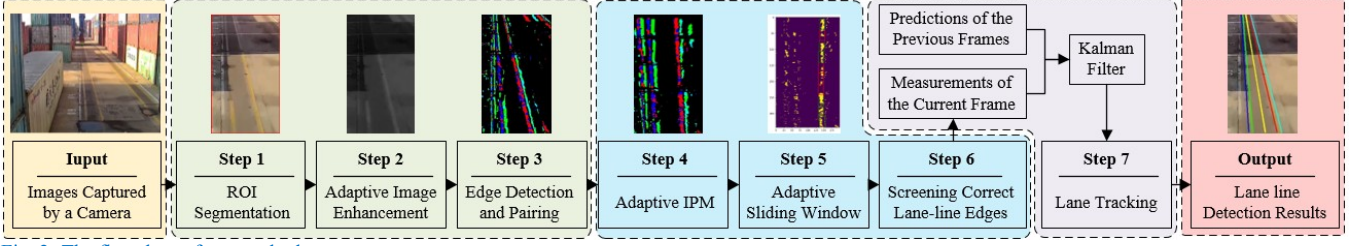


Fig. 2. The flowchart of our method.

slice convolutions, enabling messages to pass between pixels across rows and columns in a layer. Along a similar route, REcurrent Feature-Shift Aggregator (RESA) [29] gathers information in vertical and horizontal directions and passes them with different strides to aggregate global features. This method can achieve excellent results on two popular lane detection benchmarks, CULane [28] and Tusimple [30]. Unlike image segmentation approaches, Qin *et al.* [31] presented a novel lane-detection paradigm that treats the process of lane detection as a row-based selecting problem. As a result, it can significantly reduce computational costs and improve the running speed. A review [32] summarizes and analyzes deep-learning-based methods comprehensively.

Deep-learning-based lane detection algorithms have recently achieved impressive results on many public benchmarks and attracted considerable attention. However, most treat lane detection as a pixel-wise image segmentation problem. As a result, incorrect segmentation at the minutiae may decrease the prediction precision of the lane centerline position. Low precision and expensive preparation of big datasets have limited their application to RTG's lane detection and tracking.

### III. METHODS

This section proposes the AELD algorithm. We treat a lane line as a set of two edges rather than a whole. As aforementioned, each lane consists of two parallel lane lines. Thus, each lane has four edges, i.e., a left outer (LO), a left inner (LI), a right inner (RI), and a right outer (RO), or can be divided into two inners and two outers. It is worth using the structural constraints among them to improve the reliability of lane detection and the calculation precision of centerline position. The proposed method consists of edge detection and pairing, inverse perspective mapping (IPM), lane detection, and lane tracking. The procedure of our method is illustrated in Fig. 2. First, we utilize image enhancement and gradient feature extraction to implement edge detection and pairing. Then, the parallel relationship of lane-line edges is

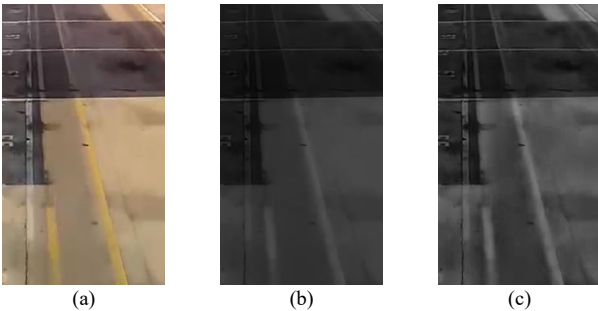


Fig. 3. Results of image preprocessing, including (a) ROI, (b) grayscale, and (c) enhancement.

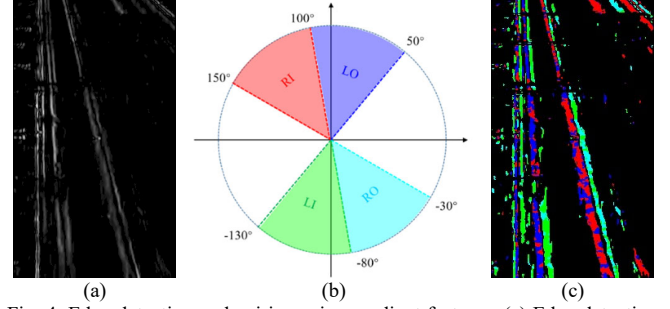


Fig. 4. Edge detection and pairing using gradient features. (a) Edge detection based on gradient magnitude, (b) a gradient direction division, and (c) edge pairing by gradient direction.

recovered by an adaptive vanishing-point-guided IPM (AIPM). Subsequently, an adaptive sliding window method is employed to detect lane lines. Finally, lane tracking is performed using an adaptive Kalman filter according to the inter-frame continuity of the image sequence.

#### A. Edge Detection and Pairing

1) *Region of Interest (ROI) Segmentation*: We use ROI segmentation to extract the area where the lane locates from an original image to reduce detection range and improve detection efficiency. Due to the fixed spacing between two lane lines and the specified cameras' field of view, the ROI does not change as RTG moves. Therefore, we take a fixed rectangle area from an original image as the ROI, as shown in Fig. 3(a).

2) *Adaptive Image Enhancement*: We gray the masked ROI segmentation in a colorful image, as shown in Fig. 3(b). Affected by complex environmental factors, the contrast between lane lines and the ground is so low that it may incur detection failure. To increase the contrast, we highlight the gradient features of lane-line edges through image enhancement. Histogram equalization (HE) [33] is a common and effective image enhancement method, but vanilla HE cannot solve the considerable differences between brightness and darkness regions. To resolve it, we chose the Contrast Limited Adaptive Histogram Equalization (CLAHE) [2]. The whole image is divided into several blocks, and each is handled by vanilla HE, followed by a bilinear interpolation fusion. Fig. 3(c) displays the enhanced image by CLAHE.

3) *Gradient-based edge detection and pairing*: After image enhancement, edge detection and pairing are performed using gradient features. First, the gradients of an image in horizontal and vertical directions,  $G_x(x, y)$  and  $G_y(x, y)$ , are extracted by a 5\*5 Sobel operator [9]. Then, the gradient magnitude feature  $GM(x, y)$  and the gradient direction feature  $GD(x, y)$  are calculated through Eqs. (1) and (2), respectively.

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

$$GM(x, y) = |G_x(x, y)|, \quad (1)$$

$$GD(x, y) = \frac{180}{\pi} \tan^{-1} \left( \frac{G_x(x, y)}{G_y(x, y)} \right). \quad (2)$$

The gradient of the region where edges locate is greater than that of the flat background. Since lane lines are approximately perpendicular to the image's X-axis, there exists a distinct change in gradients of lane-line edges in the X-axis direction against the background. As a result, the candidate edges can be detected by setting an appropriate threshold of GM, as shown in Fig. 4(a). In [4], four edges of a lane line can be paired by their gradient direction features. As illustrated in Fig. 4(b), the possible gradient direction range of lane lines can be divided into four intervals,  $[-130^\circ, -80^\circ]$ ,  $[-80^\circ, -30^\circ]$ ,  $[50^\circ, 100^\circ]$ , and  $[100^\circ, 150^\circ]$ , corresponding to the four lane-line edges. The edges can be assigned to the pairs to which their gradient direction feature belongs.

Fig. 4(c) illustrates the results of edge detection and pairing, where different colors indicate different pairs of edges. Although most edges can be detected and paired correctly, some lines and cracks parallel to lane lines may be detected by mistake, and thus parts of lane-line edges are wrongly paired. However, they can be well addressed in the following lane detection process.

### B. IPM and Lane Detection

1) *Adaptive Vanishing-point-guided IPM*: Due to the perspective transformation, two parallel lane lines intersect at a vanishing point in an image. Inverse perspective mapping is widely applied to restore the parallel relationship between two lane lines to facilitate lane detection. Authors [34], [35] conduct IPM by the camera imaging principle. These methods rely on the exact camera's intrinsic and extrinsic parameters. Therefore, their application is limited in practice. A simple IPM method is achieved by calculating the perspective transformation matrix between the original and transformed images. Given four pairs of no-collinear points, we can calculate the perspective transformation matrix by,

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = A_{3 \times 3} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad (3)$$

where A is a perspective transformation matrix, and  $[u, v, 1]$  and  $[x, y, 1]$  are, in turn, the homogeneous coordinates of an original and transformed image point.

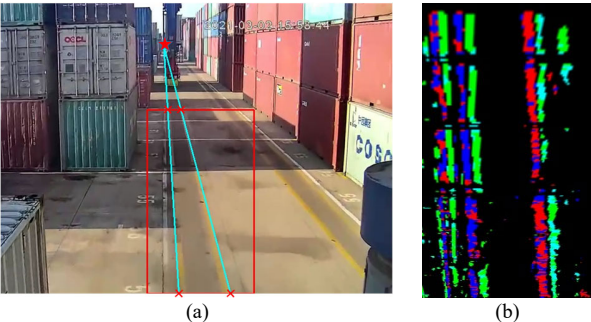


Fig. 5. (a) Illustration of AIPM and (b) transformed candidate edges by AIPM.

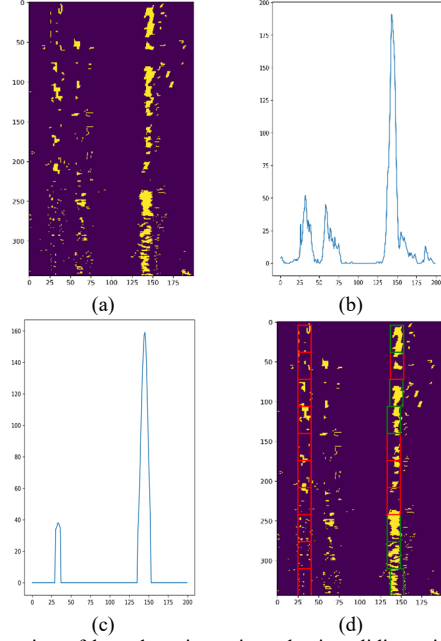


Fig. 6. Illustration of lane detection using adaptive sliding windows. (a) A transformed RO map, (b) a statistic histogram, (c) a filtered histogram, and (d) sliding windows.

We can manually select the original and transformed image points according to the position of lane lines in an image. However, the positions of lane lines change with the RTG's deflection and movement. Therefore, the fixed IPM does not work and may affect the precision of the subsequent detection. We present an adaptive vanishing-point-guided IPM (AIPM) to settle this problem.

As shown in Fig. 5(a), marked by a red five-pointed star, the vanishing point coordinates can be given by the least squares method according to the linear equations of two lane-line edges in the previous frame. Then, the vanishing point and two lane-line edges' lower endpoints are connected by two lines, i.e., the cyan lines, as illustrated in Fig. 5(a). The intersections of the two lines, given as the red crosses in Fig. 5(a), and the boundary of ROI, marked by the red rectangle in Fig. 5(a), are new corresponding points used to update the perspective transformation matrix of the current frame. Since the positions of lane lines between adjacent frames do not change abruptly, the dynamic updating perspective transformation matrix can adapt to the subtle changes in successive images. Fig. 5(b) shows the transformed result of candidate edges by AIPM.

2) *Lane Detection Based on Adaptive Sliding Window*: Note that the candidate edges transformed by IPM are perpendicular to the X-axis of the corresponding image. Here, we adopt the adaptive sliding window method to detect lane lines. Taking RO as an example, the transformed RO map is projected to the X-axis of the image to obtain a statistical histogram, as shown in Fig. 6(a) and (b), respectively. Median filtering is performed on the histogram to eliminate noise by one-dimensional convolution. Moreover, only the abscissas with statistical values that exceed 25% of the maximum are retained, as shown in Fig. 6(c). The abscissas of peak points in the filtered histogram represent the initial base point coordinates of RO.



> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

Next, we use a fixed-size window to search lane lines along the Y-axis direction starting from the initial base point in the transformed RO map, as shown in Fig. 6(d). Here, the window width  $w$  is 10 pixels, and the window height  $h$  is equal to the step size  $s$ , where,

$$h = s = \frac{H}{N}. \quad (4)$$

In Eq. (4),  $H$  represents the height of ROI, and  $N$  denotes the preset search times,  $N=10$ . Suppose that the number of edge pixels in the window exceeds the threshold  $T_p$ , the window is valid, as shown by the green boxes in Fig. 6(d), and thereby the edge pixels within it are valid. The base point abscissas of the following search window can be updated by the average abscissas of all valid edge pixels inside the current window. Otherwise, if the number of the current window's edge pixels is lower than the threshold  $T_p$ , the window, i.e., the red boxes in Fig. 6(d), continues to scan along the Y-axis direction with the unchanged base point abscissas until the preset  $N$  times of search is performed. For each lane-line edge, if the number of corresponding valid windows is greater than threshold  $T_w$ , the lane-line edge is considered valid.

Threshold  $T_p$  adapts to the segmentation results  $P_x$  of the lane-line edge and search times  $N$ ,

$$T_p = \frac{P_x}{N}, \quad (5)$$

where,

$$P_x = \sum_{i=x-w/2}^{x+w/2} \mathcal{H}_i. \quad (6)$$

In Eq. (6),  $\mathcal{H}_i$  represents the value corresponding to the  $i$ th column in the statistical histogram  $\mathcal{H}$ , and  $x$  represents the abscissa of an initial base point. The line equation of a lane-line edge can be obtained by linear fitting for the coordinates of the valid edge pixels. Since only valid edge pixels within the valid window are utilized in the fitting process, the invalid pixels do not affect the fitting result, thereby improving the accuracy of lane detection. Even if parts of a lane-line edge are missing due to smudges, occlusions, and breakages, the edge can still be successfully detected by setting an appropriate  $T_w$ .

3) *Screening Correct Lane-line Edges*: In the edge detection process, many edges that do not belong to any lane line are mistakenly detected. Although most of the false edges can be filtered out by the adaptive sliding window method, the rest result in multiple possible candidates for a lane-line edge. We propose an effective approach to screen the remaining false edges thanks to the RTG lanes' structural constraints. Considering the fixed spacing between two lane-line edges of the same pair, inner or outer, one can calculate the distance between the initial base points of every two candidate edges in the same pair. Then, the absolute difference between the resultant distance and the actual spacing of two lane-line edges can be obtained. If the edges whose differences are greater than the lane-line width, they have to be discarded. Finally, the candidate edges with a minimal difference are the final results of lane-line edge detection.

### C. Lane Tracking

Lane detection with a single frame does not consider inter-frame continuity. Therefore, it may lead to poor detection stability for continuous image sequences. In addition, in some extreme cases, e.g., intense exposure and severe occlusion, lane detection may face failure, which is unacceptable for RTG autonomous driving. Therefore, we design an adaptive Kalman filter based on detection confidence weighting to track lane lines and introduce the lane-line width as a soft constraint into the filter. The lane detection confidence as a prior can help correct measurement noise variance and adaptively adjust Kalman gains.

1) *Kalman Filtering*: The Kalman filter assumes that the position of the tracking target varies linearly between two consecutive frames. It produces an estimated state  $\hat{x}_k$  as a weighted average of the predicted state  $x_k$  and the measurement  $z_k$ ,

$$\hat{x}_k = x_k + K_k (z_k - Mx_k), \quad (7)$$

$M$  is the measurement matrix, which maps the state space to the measurement space. In the standard Kalman filter, the  $k$ th frame's Kalman gains  $K_k$  are given by,

$$K_k = P_k M^T (M P_k M^T + R)^{-1}, \quad (8)$$

where  $P_k$  is the predicted estimate covariance of the  $k$ th frame.

$P_k$  is initialized as an identity matrix  $I$ , and it is calculated by,

$$P_k = S \hat{P}_{k-1} S^T + Q, \quad (9)$$

where  $S$  represents an state transition matrix and  $Q = s_Q \times I$  represent the process noise covariance, in which  $s_Q$  is a scaling coefficient. The posterior estimate covariance of the  $k$ -1th frame,  $\hat{P}_{k-1}$ , is updated by,

$$\hat{P}_k = (I - K_k M) P_k, \quad (10)$$

The measurement noise covariance  $R = s_R \times I$ , where  $s_R$  is a scaling coefficient.

2) *Soft Kalman Filter with Constraint of Lane-line Width*: We choose the endpoint of four lane-line edges as the tracking target whose coordinate varies linearly between two consecutive frames. In addition, considering the lane-line width is fixed, the distance between two edges' endpoints of a lane line should be a constant value. Thus, we add the distance as a new tracking target to limit its change range. This improvement can enhance the accuracy and stability of lane tracking. On the other hand, this constraint is soft, i.e., the distance estimate can vary gently instead of keeping unchanged. This flexibility avoids detection failure due to unsatisfied constraints as lane line status is poor.

The ordinates of lane-line edges' endpoints are invariable for an ROI. Hence, we only measure the abscissas of eight endpoints of four lane-line edges. We can obtain the measurement  $Z_k = [z_{LO}^l, z_{LO}^b, z_{LI}^l, z_{LI}^b, z_{RI}^l, z_{RI}^b, z_{RO}^l, z_{RO}^b]^T \in \mathbb{R}^{8 \times 1}$  from the results of lane detection, where the subscripts LO, LI,

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

RI, and RO, represent the edge to which the endpoint belongs, and the superscripts t and b represent the top and bottom endpoints of an edge. The predicted state vector  $X_k = [x_{LO}^t, \Delta x_{LO}^t, x_{LO}^b, \Delta x_{LO}^b, \dots, d_L^t, \Delta d_L^t, \dots]^T \in \mathbb{R}^{24 \times 1}$  is calculated according to the estimation of the previous frame. Taking two left lane-line edges LO and LI as an example,  $x_{LO}^t$  is the predicted abscissa of the top endpoint of LO, and  $d_L^t = x_{LI}^t - x_{LO}^t$  is the distance between two top endpoints of LO and LI, i.e., the upper width of the left lane line. The operator  $\Delta$  is used to calculate the variation between two consecutive frames. It means that the soft Kalman filter needs to predict not only the abscissas of endpoints but also the width of a lane line utilized as a soft constraint. The element  $M_{ij}$  (at row  $i$ , column  $j$ ) in the measurement matrix  $M \in \mathbb{R}^{8 \times 24}$  is defined as,

$$M_{ij} = \begin{cases} 1 & i=0,1,\dots,7; j=2i \\ 0 & \text{else} \end{cases}. \quad (11)$$

3) *Detection Confidence Weighting*: As given in Eq. (7), the estimation of the  $k$ th frame  $\hat{x}_k$  is a weighted average of the predicted state  $x_k$  and the measurement  $z_k$ . The weight  $K_k$ , i.e. the Kalman gain, is calculated by Eq. (8), where the measurement matrix  $M$  is constant. The predicted estimate covariance  $P_k$  updates to adjust the weight of the predicted state during the tracking. Generally, the noise of measure process and the measurement noise covariance  $R$  are invariable. But the performance of lane detection, as a measure, varies with the changing lane-line condition. For lane lines with vivid colors, sharp edges, and strong contrast, the performance of lane detection can be better and higher confidence should be given. On the contrary, the detection of contaminated, occluded, and separate lane lines is unreliable, so the lower detection confidence is exploited to down-weight the measurement. We correct  $R$  according to the detection confidence. Suppose that the detection confidence of a lane-line edge is higher, and the corresponding measurement noise covariance may decrease.

The core matter is how to estimate the detection confidence accurately. As aforementioned, the adaptive sliding windows method is adopted in the process of lane detection. The number of valid windows reflects the detection reliability of the current lane-line edges, i.e., the more valid windows, the higher the detection confidence of the edge. Based on this prior, the confidence of lane detection can be given by,

$$c_L = e^{-\frac{n_L}{N} \ln \beta}, \quad (12)$$

where  $c_L$  and  $n_L$  represent the detection confidence and the number of valid windows for lane-line edges  $L$ , respectively.  $L \in \{LO, LI, RI, RO\}$ . The preset search number  $N=10$  and the corrected coefficient  $\beta=0.01$ . The two endpoints of a lane-line edges have the same  $c_L$ , and the corresponding confidence vector is  $C_k = [c_{LO}, c_{LO}, c_{LI}, c_{LI}, c_{RI}, c_{RI}, c_{RO}, c_{RO}]^T \in \mathbb{R}^{8 \times 1}$ . The

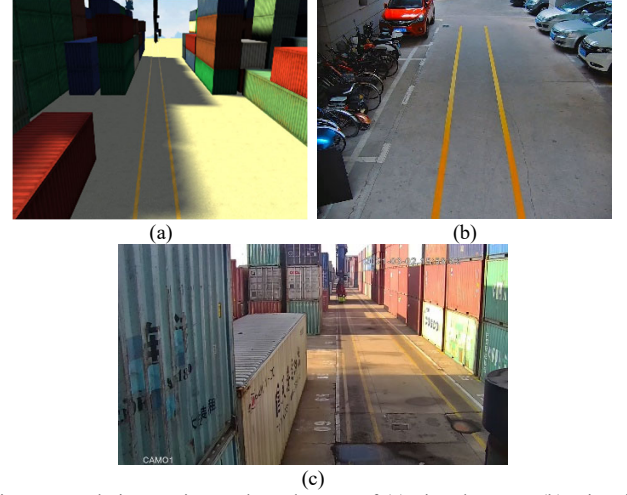


Fig. 7. Sample images in our three datasets of (a) virtual scenes, (b) mimetic scenes, and (c) real scenes.

corrected measurement noise covariance  $R'$  is given by,

$$R' = \text{diag}(C_k)R, \quad (13)$$

where  $\text{diag}(\cdot)$  is a diagonalization. By introducing  $R'$ , the adaptive Kalman gains can be denoted as,

$$K_k = P_k M^T (M P_k M^T + R')^{-1}. \quad (14)$$

## IV. EXPERIMENTS

### A. Dataset Preparation

There is no public dataset available above RTG lane detection. We made three specific datasets for verifying and testing our method, i.e., virtual scene, mimetic scene, and real scene datasets. Samples of three datasets are shown in Fig. 7.

1) *Virtual Scene Dataset*: We built a virtual environment for testing RTG lane detection using ThreeJS, a web 3D engine, including an RTG, containers, and RTG lanes, as shown in Fig. 7(a). To test our method under different lighting conditions, we simulate the light environments in the morning, noon, dusk, and night. This dataset contains 2,000 images of lanes with segmentation labels. We select for test 100 sample images with labels.

2) *Mimetic Scene Dataset*: We built a movable test platform to simulate an upright of an RTG to facilitate the experiments. The installation height of the camera is 3200mm, and the angle between the optical axis and the ground is  $45^\circ$ . We lay out a mimetic yellow lane outdoors, 10mm in width and 10m in length. The inner spacing and outer spacing of the lane are 100mm and 120mm, respectively. In the daytime and night, we use the movable platform to collect 1600 images of lane lines with different disturbances, as displayed in Fig. 7(b). 100 images are sampled and labeled manually as the test dataset.

3) *Real Scene Dataset*: To verify our method in the real scenes, we collected 800 images of lanes during RTG movement in a container yard using a post-mounted camera, as shown in Fig. 7(c). This real-scene dataset contains many images in which lane lines are suffered from shadow, water, and smudges. Finally, we select 100 labeled images as the test set.

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

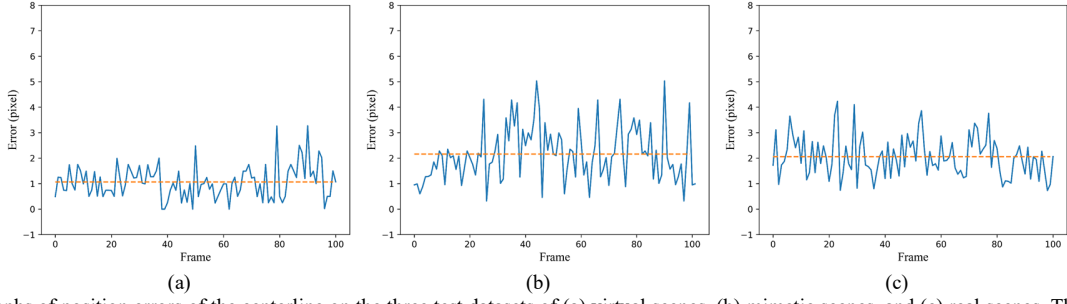


Fig. 8. Line graphs of position errors of the centerline on the three test datasets of (a) virtual scenes, (b) mimetic scenes, and (c) real scenes. The orange dotted lines represent the average value.

### B. Evaluation Metrics

The evaluation metrics, Accuracy and F1-score, are adopted for quantitatively evaluating the performance of lane line segmentation. They can be calculated by,

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}, \quad (15)$$

$$F1\text{-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \quad (16)$$

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}. \quad (17)$$

where  $TP$ ,  $FP$ , and  $FN$  represent the number of true positive, false positive, and false negative samples. Instead of outputting the segmentation maps of lane lines, we predict the positions of lane-line edges. For the convenience of comparing the methods based on segmentation, we extract the regions enclosed by lane-line edges and the border of ROI as the segmentation results. In the RTG autonomous driving scene, the main aim of lane detection is to calculate the position and azimuth errors between RTG and the lane centerline. Therefore, the estimated precision of the centerline position is an important evaluation metric. Since the lane lines in container yards are straight, we can measure the precision by the average position error as follows,

$$S_e = |x_t - x_t^*| + |x_b - x_b^*|, \quad (18)$$

$$AvgError = \frac{1}{M} \sum S_e, \quad (19)$$

where  $x_t$  and  $x_b$  represent the abscissa of the top and bottom endpoints of the predicted centerline.  $x_t^*$  and  $x_b^*$  represent the corresponding ground truth.  $S_e$  is the sum of distances between predicted results and ground truths. If  $S_e$  is greater than 10 pixels, the detection fails according to the real requirement.  $M$  represents the number of test images. The detection success rate (DSR) is the ratio of the successful detection number of images

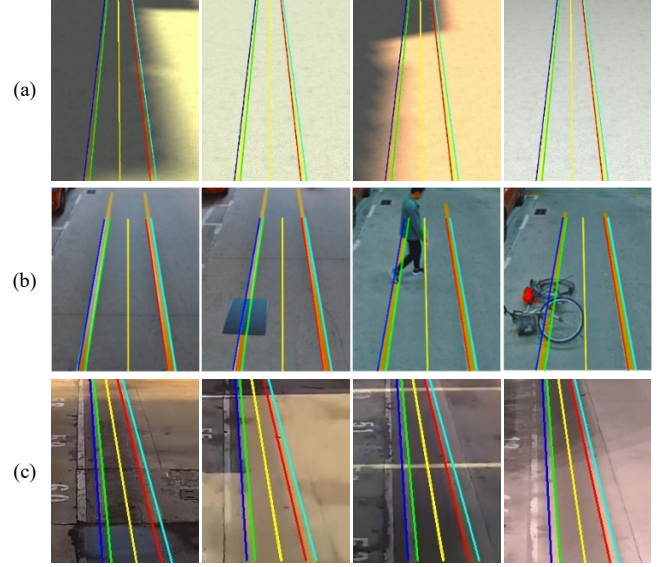


Fig. 9. Qualitative results on the three datasets of (a) virtual scenes, (b) mimetic scenes, and (c) real scenes.

to the total number of images in the test set.

### C. Experimental Results of Lane Detection and Tracking

The quantitative evaluation results on three datasets of our method are reported in TABLE I. It shows excellent performance in segmenting lane lines on all three datasets due to accurate edge detection. The F1 scores on the three datasets are 95.4, 95.0, and 89.6, respectively, indicating that our method balances precision and recall well. The estimated precision of the centerline position reaches the highest value on the virtual scene dataset, and the average position error is only 1.071 pixels, thanks to interference-free imaging. In other the two real datasets, the average position error of the centerline is approximately 2 pixels, implying that our method can support RTG autonomous driving by accurately predicting the centerline position and azimuth. With the help of the adaptive Kalman filter, stable and reliable lane tracking is achieved with detection success rates (DSR) up to 100% on the three datasets.

The centerline position errors of the 100 test images from each dataset are illustrated in Fig.8. We attain the most accurate and stable detection result on the virtual scene dataset, and the mean and standard deviation of errors are 1.071 and 0.64, respectively. On the mimetic dataset, the orientation of the movable platform is unstable when it is pushed to move, causing the standard deviation of errors to increase to 1.07, and

TABLE I  
QUANTITATIVE EVALUATION RESULTS OF OUR METHOD ON THREE DATASETS.

Datasets	F1↑	Accuracy↑	AvgError↓	DSR↑
Virtual scene	95.4	99.5	1.071	100%
Mimetic scene	95.0	99.1	2.166	100%
Real scene	89.6	97.9	2.051	100%

&gt; REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) &lt;

TABLE II  
ABLATION EXPERIMENT RESULTS.

Adaptive image enhancement	Adaptive IPM	Soft Kalman filter	Adaptive confidence weighting	Real scene dataset			
				F1↑	Accuracy↑	AvgError↓	DSR↑
n.a.	n.a.	n.a.	n.a.	84.8	96.8	4.881	71%
√	n.a.	n.a.	n.a.	87.2	97.4	3.436	92%
√	√	n.a.	n.a.	88.2	97.7	3.270	94%
√	√	√	n.a.	88.8	97.8	2.253	100%
√	√	√	√	89.6	97.9	2.051	100%

the maximum error is only 5.03 pixels. For the real scene dataset, during RTG moving for 25 meters, the centerline position error varies in [0.73, 4.23] pixel, and the standard deviation is 0.73, verifying our method's ability to maintain detection accuracy and tracking stability during the RTG's movement.

The visual results on the three datasets are illustrated in Fig. 9. Fig. 9(a) shows the detection results of virtual lane lines in the morning, noon, dusk, and night. In turn, the detection of normal and discontinuous lanes during daytime and lanes with pedestrian and obstacle occlusions at night are shown in Fig. 9(b). Fig. 9(c) illustrates the detection performance under contamination, edge blur, shadow, and nighttime conditions in a real container yard. The enhanced gradient features accurately detect edges under different colors and lighting conditions. The detection problems of the lane with occlusions and breakages are well addressed by the adaptive sliding windows method. Although the lane lines have obvious differences in the three datasets, our method applies to cross-dataset detection without fine-tuning parameters. Therefore, our method also has outstanding generalization ability.

#### D. Ablation Experiments

To verify the impact of different components in our method on lane detection, we conduct ablation experiments on adaptive image enhancement, adaptive IPM, soft Kalman filter, and adaptive confidence weighting. The results are shown in TABLE II.

1) *Adaptive Image Enhancement*: Due to the gradient-based edge detection and adaptive sliding, our baseline method can identify the location of lane lines in an image with an F1 score of 84.8 and an accuracy of 96.8%. However, the precision of detection in detail is low, the mean error of the centerline position is up to 4.881 pixels, and the detection success rate is only 71%. The adaptive image enhancement can intensify the image contrast and improve the effect of gradient feature extraction. In contrast to the baseline, the average error is

decreased by 1.4 pixels, and the detection success rate grows by 21%.

2) *Adaptive IPM*: The adaptive IPM (AIPM) is designed for situations where the RTG direction changes significantly. The improvement of AIPM is insignificant in the real-scene datasets collected when RTG moves straight. The F1 score is increased by 1.0, and the average error is reduced by 0.2 pixels. We manually control the movable platform to change direction when making the mimetic dataset. The comparison of the fixed IPM and the adaptive IPM is shown in Fig. 10. By using the fixed IPM, the original region cannot follow the lane lines change, and the transformed edges are not perpendicular to the X-axis of an image. The AIPM method can dynamically adjust the original region according to the position of lane lines. Thus, the transformed edges are always perpendicular to the X-axis.

3) *Soft Kalman Filter*: A soft Kalman filter can increase the detection success rate to 100% and achieve stable lane tracking. Fig. 11 reports the width change of the detected left lane-line bottoms in 800 images from the real scene dataset. If the Kalman filter considers a soft constraint of lane-line width, the average line width predicted is 12.46 pixels, very close to the actual line width of 12.5 pixels, and then the standard deviation is only 0.315. However, the average line width decreases to 12.28 pixels, and the standard deviation grows quickly to 2.670 without considering the soft line-width constraint. By introducing the soft constraint, the variation range of line width significantly decreases, and the centerline position error drops from 3.270 to 2.253 pixels.

4) *Adaptive Confidence Weighting*: The improved adaptive Kalman filter with detection confidence weighting can further reduce the centerline position error from 2.253 to 2.051 pixels

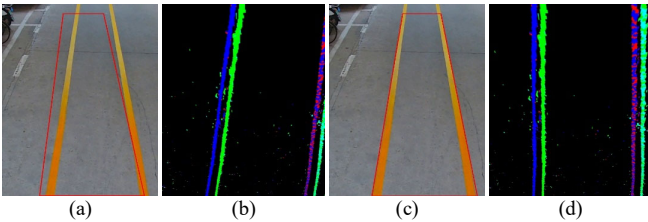


Fig. 10. Comparison between the fixed IPM and our adaptive IPM. (a) The original region and (b) transformed edges of the fixed IPM. On the contrary, (c) the original region and (d) transformed edges of the adaptive IPM.

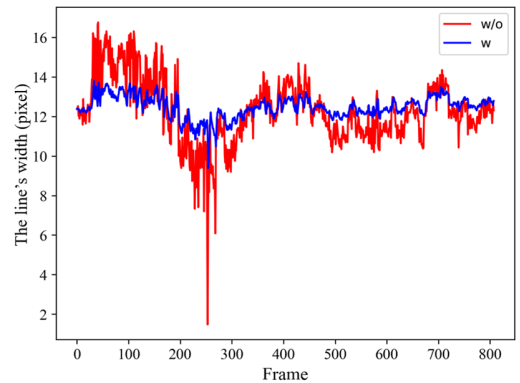


Fig. 11. The changing curve of the predicted lane-line width. Blue and red curves represent the results predicted by a filter with and without the soft constraint of lane-line width, respectively.



> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

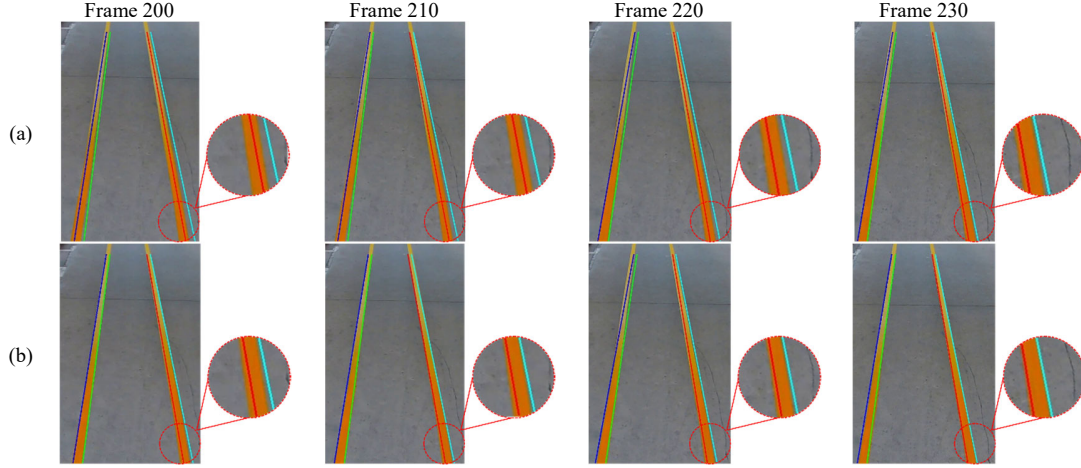


Fig. 12. Comparisons of different Kalman filters of (a) the standard Kalman filter and (b) the adaptive Kalman filter with detection confidence weighting.

and maintain a 100% detection success rate. Specifically, as illustrated in Fig. 12(a), the standard Kalman filter in the 200th frame cannot adapt promptly to the changes in lane line position. Namely, lane tracking falls behind the changes due to the influence of the previous predicted state. As a result, from Frame 200, the estimated lane lines deviate from the true positions till Frame 230. In this paper, our method can adjust the weights between the predicted states and measurements according to the detection confidence. Thus, the Kalman filter can quickly adapt to changes in RTG motion and obtain more accurate estimations of lane line positions.

In summary, our adaptive lane detection and tracking method has the following advantages and benefits. First, the CLAHE algorithm introduced has a better image enhancement effect under the condition of non-uniform brightness. Next, the adaptive inverse perspective mapping method can adapt to the subtle changes in the viewing angle by using a dynamically updated transformation matrix instead of a fixed one. Finally, compared with the ordinary Kalman filter, the adaptive Kalman filter we adopted can better balance the impact of the predicted and measurement on the final output.

#### E. Comparisons with State-of-the-art Methods

Two traditional image-processing-based methods, i.e., Hough transform (HT) [3] and sliding windows (SW) [3], and two deep-learning-based methods, i.e., SCNN [28] and RESA [29], are tested on the real scene dataset and compared with our method, AELD. The ROI, color threshold, and other parameters of the two traditional methods are set manually to meet the actual situation of lane lines in the dataset. The deep-

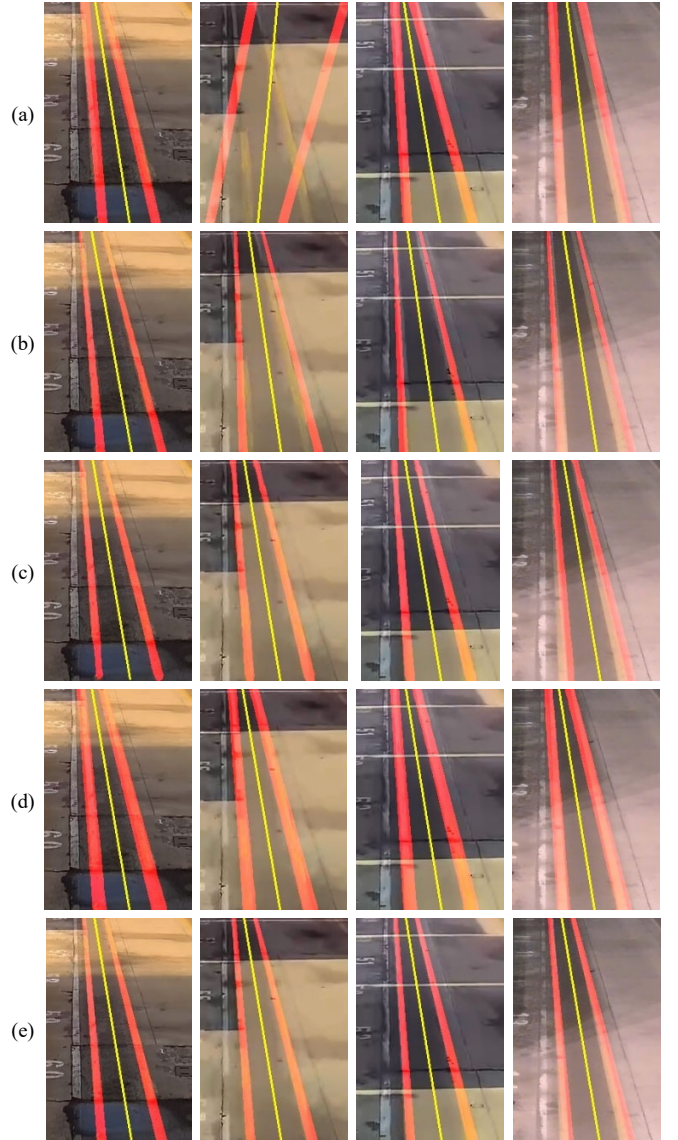


Fig. 13. Detection effects of other methods, i.e., (a) HT, (b) SW, (c) SCNN, (d) RESA, and (e) AELD.

TABLE III  
COMPARISONS OF OUR METHOD WITH OTHER METHODS ON REAL SCENE DATASET.

Methods	F1↑	Accuracy↑	AvgError↓	DSR↑
HT	44.4	87.4	9.108	26%
SW	57.2	92.3	6.979	75%
SCNN	3.3/90.4	90.1/98.1	10/3.008	0%/93%
RESA	5.5/78.7	90.2/94.6	10/2.368	0%/100%
<b>AELD</b>	89.6	97.9	2.051	100%

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

learning-based methods are pre-trained on the virtual scene dataset for 30 epochs and fine-tuned on the real scene dataset for 100 epochs with a learning rate of 0.016. The results of the five methods are listed in Table III. In Table III, lane detection using the Hough transform performs poorly and only reaches a detection success rate of 26%, much less than the actual demand. While using a sliding-window method, the detection success rate increases substantially to 75%, and the centerline's average position error is close to 7 pixels, resulting in erroneous position and azimuth deviations.

For the two deep-learning-based methods, SCNN [28] and RESA [29], the blue and black numbers represent the validation results of the pre-trained and fine-tuned models, respectively. Although images in the virtual scene datasets are similar to those in the real scene, two learning-based methods cannot bridge the gap between the two datasets. As a result, the model pre-trained in the virtual environments does not apply to actual container yards. Thanks to the high similarity between the training set and test set, the fine-tuned SCNN [28] achieves good segmentation results with an F1 score of 90.4 and an accuracy of 98.1%. But the prediction precision of the centerline is still less than our AELD, i.e., an average error of more than 3 pixels and a detection success rate of 93%. RESA [29] obtains a better prediction precision of the centerline and a 100% detection success rate, but its F1 score is only 78.7. Compared with the four methods, AELD significantly improves the accuracy of lane line segmentation and centerline location. In particular, AELD's average position error of the centerline is reduced to 2.051 pixels, 32% and 13% less than those of SCNN [28] and RESA [29], respectively.

Figure 13 illustrates the effects of lane line segmentation and centerline location using different methods. Specifically, in the Hough transform-based method, lane detection fails if the edges are blurred, and the absence of IPM causes segmentation accuracy to deteriorate. In comparison, the detection performance of the sliding-window method with IPM is improved notably. However, the detection results still have a large deviation from the true position of the lane line, especially in the cases of blurred edges and nighttime. SCNN exhibits

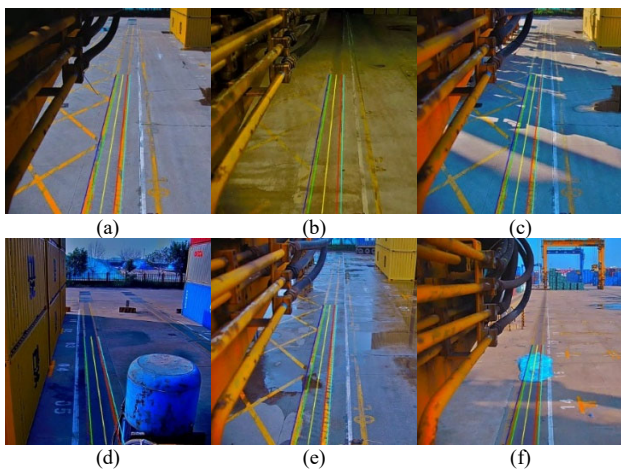


Fig. 14. Test effects under different conditions in a real container yard. (a) Daytime, (b) nighttime, (c) complex shadow, (d) serve shadow, (e) surface water, and (f) occlusion.

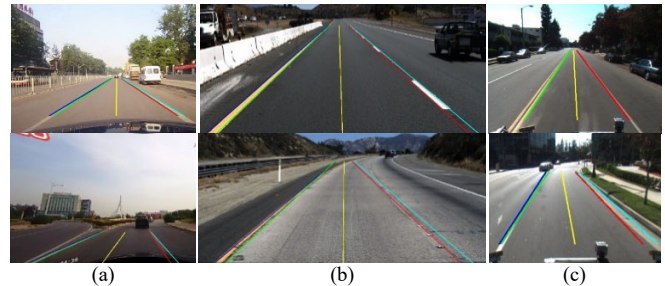


Fig. 15. Effects of lane detection on public datasets of (a) CULane, (b) Tusimple, and (c) Caltech. Pictures are cropped and scaled for illustration.

better performance than the above two methods, but the prediction of centerlines produces large errors when the lane lines are contaminated or at nighttime. The segmentation results of RESA almost cover the regions where lane lines locate, while they are too coarse at the detailed edges. Our AELD has excellent environmental adaptability and outperforms lane detection in multiple complex scenarios.

#### F. Real Test in a Container Yard

We conduct the all-weather tests in a manual container terminal to verify our method. The results are shown in Fig. 14. The test results reveal that our method can perform robust lane detection and tracking under complex lighting and weather conditions, i.e., daytime, nighttime, sunny, and rainy days, and suffering from shadows, contamination, surface water, and occlusions. Furthermore, our method can maintain a high detection precision and success rate in the operating container yard and provide accurate position and azimuth deviation for RTG's autonomous driving.

#### G. Expanded Experiments in Road Scenes

Although our AELD is designed for container yard scenes, we can also easily extend it to other scenarios where lane detection accuracy and robustness are required by adjusting a few parameters, e.g., ROI and lane-line width. We test it on the lane detection datasets of the road scenes such as CULane [28], Tusimple [30], and Caltech [36], and satisfactory effects are achieved, as shown in Fig.15. The performance of AELD is slightly reduced at the turn of lane lines because we fit the lines by a first-order polynomial for container yard scenes. Improving AELD to suit road scenes is our possible research direction.

### V. CONCLUSION

This paper proposes an adaptive edge-based lane detection and tracking algorithm (AELD) for RTG autonomous driving in a container yard. The adaptive image enhancement and gradient features extraction are utilized to realize lane-line edge detection and grouping in complex lighting and weather conditions. We design a vanishing-point-guided inverse perspective mapping approach to adapt to changes in RTG's orientation. The soft constraint of lane-line width is introduced into the Kalman filter. The Kalman gains can be adjusted dynamically according to the detection confidence, improving the stability of lane tracking. Finally, we build three scene datasets for the test. The results of extensive experiments and



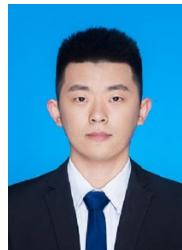
> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

tests show that AELD achieves outstanding performance with a detection success rate of 100% and the position error of the centerline less than 3 pixels, better than those by the compared state-of-the-art methods, e.g. SCNN and RESA.

Considering possible failure of our image-processing-based approach in some extreme cases, e.g., where lane lines are very blurred or even large segments of them invisible, based on it our ongoing work is to explore a low-cost learning mode to train deep networks for accurate and fast detection of RTG lane lines in more complex situations.

## REFERENCES

- [1] R. Linn, J. Liu, Y. Wan, C. Zhang, and K. G. Murty, "Rubber tired gantry crane deployment for container yard operation," *Comput. Ind. Eng.*, vol. 45, no. 3, pp. 429-442, Oct. 2003.
- [2] A. M. Reza, "Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement," *J. VLSI Signal Process. Syst. Signal Image Video Technol.*, vol. 38, no. 1, pp. 35-44, Nov. 2004.
- [3] R. Muthalagu, A. Bolimera, and V. Kalaichelvi, "Lane detection technique based on perspective transformation and histogram analysis for self-driving cars," *Comput. Electr. Eng.*, vol. 85, Jul. 2020.
- [4] C. Lee and J. H. Moon, "Robust lane detection and tracking for real-time applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 4043-4048, Feb. 2018.
- [5] Y. Feng, and J. Li, "Robust lane detection and tracking for autonomous driving of rubber-tired gantry cranes in a container yard," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2022, pp. 1729-1734.
- [6] H. Yoo, U. Yang, and K. Sohn, "Gradient-enhancing conversion for illumination-robust lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1083-1094, Apr. 2013.
- [7] V. Gaikwad, and S. Lokhande, "Lane departure identification for advanced driver assistance," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 910-918, Sep. 2014.
- [8] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679-698, Nov. 1986.
- [9] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE J. Solid-State Circuits*, vol. 23, no. 2, pp. 358-367, Apr. 1988.
- [10] A. Borkar, M. Hayes, and M. T. Smith, "Lane detection and tracking using a layered approach," in *Proc. ACIVS*, Oct. 2009, pp. 474-484.
- [11] A. Borkar, M. Hayes, and M. T. Smith, "A novel lane detection system with efficient ground truth generation," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 365-374, Dec. 2011.
- [12] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11-15, Jan. 1972.
- [13] M. A. Fischler, and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381-395, Jun. 1981.
- [14] A. Küçükmanisa, G. Tarım, and O. Urhan, "Real-time illumination and shadow invariant lane detection on mobile platform," *J. Real-Time Image Process.*, vol. 16, no. 5, pp. 1781-1794, Oct. 2017.
- [15] S. Bisht, N. Sukumar, and P. Sumathi, "Integration of hough transform and inter-frame clustering for road lane detection and tracking," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, May. 2022, pp. 1-6.
- [16] S. Luo, X. Zhang, J. Hu, and J. Xu, "Multiple lane detection via combining complementary structural constraints," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7597-7606, Jul. 2020.
- [17] D. Ding, C. Lee, and K. Lee, "An adaptive road ROI determination algorithm for lane detection," in *Proc. IEEE Reg. 10 Annu. Int. Conf.*, Oct. 2013, pp. 1-4.
- [18] Y. Su, Y. Zhang, T. Lu, J. Yang, and H. Kong, "Vanishing point constrained lane detection with a stereo camera," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2739-2744, Nov. 2017.
- [19] P. C. Wu, C. Y. Chang, and C. H. Lin, "Lane-mark extraction for automobiles under complex conditions," *Pattern Recognit.*, vol. 47, no. 8, pp. 2756-2767, Aug. 2014.
- [20] R. Jiang, K. Reinhard, V. Tobin, and S. Wang, "Lane detection and tracking using a new lane model and distance transform," *Mach. Vision Appl.*, vol. 22, no. 4, pp. 721-737, Jan. 2011.
- [21] S. Chen, L. Huang, H. Chen, and J. Bai, "Multi-lane detection and tracking using temporal-spatial model and particle filtering," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2227-2245, Mar. 2021.
- [22] P. Lu, S. Xu, and H. Peng, "Graph-embedded lane detection," *IEEE Trans. Image Process.*, vol. 30, pp. 2977-2988, Feb. 2021.
- [23] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane, "A review of recent advances in lane detection and departure warning system," *Pattern Recognit.*, vol. 73, pp. 216-234, Jan. 2018.
- [24] D. Liang, Y. Guo, S. Zhang, and T. Mu, "Lane detection: a survey with new results," *J. Comput. Sci. Technol.*, vol. 35, no. 3, pp. 493-505, May. 2020.
- [25] D. Neven; B. D. Brabandere; S. Georgoulis; M. Proesmans; and L. V. Gool, "Towards end-to-end lane detection: an instance segmentation approach," in *Proc. IEEE Intell. Veh. Symp.*, Sep. 2018, pp. 286-291.
- [26] Y. Ko; Y. Lee; S. Azam; F. Munir; M. Jeon; and W. Pedrycz, "Key points estimation and point instance segmentation approach for lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8949-8958, Jul. 2022.
- [27] S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T. Lee, H. S. Hong, S. Han, and I. S. Kweon, "Vpgnet: Vanishing point guided network for lane and road marking detection and recognition," in *Proc. IEEE Int. Conf. Comput. Vision*, Oct. 2017, pp. 1947-1955.
- [28] X. Pang, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, pp. 7276-7283.
- [29] T. Zheng, H. Fang, Y. Zhang, W. Tang, Z. Yang, H. Liu, and D. Cai, "Resa: Recurrent feature-shift aggregator for lane detection," in *Proc. AAAI Conf. Artif. Intell.*, May. 2021, pp. 3547-3554.
- [30] TuSimple. (2017). *Tusimple benchmark*. [Online]. Available: <http://benchmark.tusimple.ai/>.
- [31] Z. Qin, H. Wang, and X. Li, "Ultra fast structure-aware deep lane detection," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 276-291.
- [32] Y. Zhang, Z. Lu, X. Zhang, J. Xue, and Q. Liao, "Deep learning in lane marking detection: a survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 5976-5992, Apr. 2021.
- [33] K. Murahira, T. Kawakami, and A. Taguchi, "Modified histogram equalization for image contrast enhancement," in *Proc. Int. Symp. Commun., Control, Signal Process. (ISCCSP)*, Mar. 2010, pp. 1-5.
- [34] D. Zhang, B. Fang, and W. Yang, "Robust inverse perspective mapping based on vanishing point," in *Proc. IEEE Int. Conf. Secur., Pattern Anal., Cybern.*, Oct. 2014, pp. 458-463.
- [35] J. Jeong, and A. Kim, "Adaptive inverse perspective mapping for lane map generation with SLAM," in *Proc. Int. Conf. Ubiquitous Robots Ambient Intell.*, Aug. 2016, pp. 38-41.
- [36] M. Aly, "Real time detection of lane markers in urban streets," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2008, pp. 7-12.



**Yunjian Feng** received M.S. degree in vehicle electronic engineering from Wuhan University of Technology, Wuhan, China, in 2020. He is currently pursuing the Ph.D. degree in control theory and control engineering from Southeast University. His current research interests include machine vision, deep learning, and autonomous driving.



**Jun Li** (Senior Member, IEEE) received Ph.D. degree in control theory and control engineering from Southeast University (SEU), Nanjing, China, in 2007.

From 2008 to 2010, he was a Post-Doctoral Fellow at SEU. In 2014, he was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA. He is the Director of the Robotics and Intelligent Systems Laboratory at SEU. His current research interests include machine vision, logistics and construction robotics, machine learning, and operations research.

Professor Li is a Fellow of the Institution of Engineering and Technology (IET) and serves as an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.