Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 54-20 (2021) 147–152

Real-Time Adaptive Threshold Adjustment for Lane Detection Application under Different Lighting Conditions using Model-Free Control

Yujing Zhou, Zejiang Wang, and Junmin Wang

Walker Department of Mechanical Engineering

The University of Texas at Austin

Austin, TX 78712, USA (e-mail: yz009@utexas.edu, wangzejiang@utexas.edu, jwang@austin.utexas.edu)

Abstract: Lane detection and tracking technologies are critical to automated vehicles and the lane departure warning (LDW) feature in advanced driver assistance systems (ADAS). The color difference between lane marks and their background is an important source for drivers to detect lanes. However, most existing lane detection techniques neglect the importance of color features contained within lane marks because the color is subject to lighting variations. The hue, saturation, and value (HSV) color space is commonly used for color identification, as it separates the color, saturation, and brightness information into the H, S, and V channels, respectively. Colors can be identified with specified thresholds in H, S, and V channels, but fixed thresholds can hardly treat the disturbances caused by variation of lighting conditions. This paper leverages the color information of lane marks to perform the lane detection task by adaptively changing the thresholds in the HSV space to handle lighting condition variations. In doing so, the vehicle is assumed to follow a straight lane with a reference steering angle of zero degree under varying lighting conditions. It is hard to model the relationship between the thresholds in the HSV space and the vehicle steering angle in automated driving, so we adopt the model-free control (MFC), which is a data-driven approach and does not require an explicit model for the system. Simulink-QUARC joint offline experiments demonstrate the effectiveness of the proposed real-time HSV threshold adjustment method for better lane detections.

Keywords: Autonomous vehicles, Detection algorithms, Threshold selection, Adaptive systems, Self-adjusting systems, Model-free control

1. INTRODUCTION

Driving safety is receiving increasing attention worldwide. As fully automated vehicles are still in the development stage, more and more advanced driver assistance systems (ADAS) have been put in use, aiming to assist human drivers for safety enhancement on the roads. Unintended lane departure is among the leading causes of vehicle accidents (Narote *et al.* 2018). Consequently, lane departure warning systems (LDW) and other active lateral motion control systems (Wang *et al*., 2017, Jin *et al*., 2018, Wang *et al*., 2015, Zhu *et al.*, 2015, Chen *et al*., 2021, Zhou *et al*., 2021) have been developed to help a driver keep the vehicle in its lane. The prerequisite of most, if not all, LDW systems is the correct identification of lane markings. A vehicle with a LDW system typically has a camera mounted to the front windshield. It captures the real-time images that contain lane marks with complex environment information. LDW system has to separate the lane features from the background. Existing approaches can be generally divided into three tasks: image pre-processing, lane modeling, and lane detection (Narote *et al.* 2018).

Image pre-processing aims to decrease the computational time and improve the identification results. Color processing and region of interest (ROI) selection are typically used. An image is originally represented in the red-green-blue (RGB) color space. The color information is correlated among red, green, and blue channels, and it is very sensitive to changes in illumination. Therefore, color processing converts images from RGB space to grayscale intensities (Wu *et al.* 2014; Borkar *et al.* 2009) or hue, saturation, and value (HSV) space (Li *et al.* 2018; Vitabile *et al.* 2001; Mammeri *et al.* 2016) to improve the image contrast and the robustness of color representation under various lighting conditions. Besides the lane features, an image from the camera may contain rich background information, which imposes a heavy computational burden on the subsequent algorithms. ROI selection algorithms (Wu *et al.* 2014; Zhao *et al.* 2012; Li *et al.* 2018) aim to reduce the image sizes by deducting irrelevant features from the images to alleviate computational expenses. Lane modeling aims to mathematically characterize lane marks using linear- (Wu *et al.* 2014) or higher-order (Wu *et al.* 2014; Zhao *et al.* 2012; Wang *et al.* 2004) models. On the other hand, the feature-based model localizes lanes by extracting low-level features. For instance, edge detectors, such as Canny (Wu *et al.* 2014), Gabor (Mehrotra *et al.* 1992), Sobel (Mu *et al.* 2014) are typically implemented.

However, the existing lane detection methods seldomly consider the importance of color features for lane marks, which have strong contrasts with the road color and can be good resources to separate lane regions from their background. It is largely because color identification is subject to changing lighting conditions, and robust lane detection is hard to achieve based only on color information (Banik *et al.* 2018; He *et al.* 2012).

10.1016/j.ifacol.2021.11.167

HSV color space is widely used for color identification because it separates the color, saturation, and brightness information in an image from the RGB space into the hue (H), saturation (S), and value (V) channels, respectively. It is a direct transformation from the RGB space with the following relations (Mammeri *et al.* 2016):

$$V = \max(R, G, B), \tag{1}$$

$$S = \begin{cases} \frac{V - \min(R,G,B)}{V} & if\ V \neq 0, \\ 0 & otherwise, \end{cases} \tag{2}$$

$$H = \begin{cases} \frac{60(G-B)}{V-\min(R,G,B)} & if\ V = R, \\ \frac{120+60(B-R)}{V-\min(R,G,B)} & if\ V = G, \\ \frac{240+60(R-G)}{V-\min(R,G,B)} & if\ V = B. \end{cases} \tag{3}$$

R, G, B values in (1-3) are normalized, and H, S, and V values have ranges: $H \in [0,360]$, $S \in [0,1], V \in [0,1]$. For each H, S, and V parameter in the HSV space, there is an upper bound and a lower bound associated with it. So there are totally six threshold values. A pixel in an image to be identified as part of a lane mark must have H, S, and V values within their respective ranges.

From previous studies, a dynamic hue (H) threshold method based on the peak value in the hue histogram was proposed in (Liang *et al.* 2011). Additional filters were required to eliminate the unwanted information, and this method was not robust enough to illumination variances. In (He *et al.* 2012), an adaptive threshold in the V channel was proposed for shadow detection based on pixels change velocity in the shadow area and the ratio of shadow pixels to moving object pixels. However, its application was limited to shadow detection for moving objects, and it requires specified video directions.

This paper proposes a novel way to adaptively change the thresholds in the HSV space from a control perspective. Lane detection under different lighting conditions is demonstrated to test its effectiveness. We assume that the vehicle travels along a straight lane, which is the most common vehicle operation, so that the steering angle of zero degree becomes the control reference. The designed controller adjusts the thresholds in the HSV space to let the real-time steering angle from the automated driving system follow the reference value.

Traditional model-based controllers typically rely on a pre-defined model to calculate the control commands. The plant model will have uncertain parameters, unmodeled dynamics, and disturbances in real applications. With physically unrelated variables in this application, which are vehicle steering angle and thresholds in the HSV color space, it is hard, if not impossible, to formulate a mathematical model to characterize their relations. So we adopted the model-free controller introduced in (Fliess *et al.* 2013). Model-free control (MFC) is a data-driven controller, which only requires a general understanding of the system for parameter tuning. It uses a local model to connect the system input and output, and it is updated continuously based on the input-output behaviors. In this paper, MFC is implemented to a scaled vehicle with a front camera to identify the lane marks under changing lighting conditions by following a reference steering angle. Simulink-QUARC joint offline experiments demonstrate the effectiveness of the proposed method.

The rest of the paper is organized as follows. Section 2 introduces the model-free control. Controller design and its application to this study are given in Section 3. Section 4 describes the system configuration. In Section 5, results from an offline experiment on a scaled car are shown. Finally, section 6 concludes the paper.

2. MODEL-FREE CONTROL

The idea behind MFC is to represent a system in an ultra-local model:

$$y^{(n)}(t) = F(t) + \alpha u(t), \tag{4}$$

where $y^{(n)}(t)$ represents the *n-th* order derivative of the system output, $F(t)$ is a piecewise constant encompassing the model information, α is an offline-tuned constant such that $y^{(n)}(t)$ and $\alpha u(t)$ are of the same magnitude, and $u(t)$ is the input to the system. $F(t)$ will be updated at each time step according to the input-output behaviors. Equation (4) is applied to a single-input-single-output (SISO) system in this study. Although the model order n is adjustable, it is demonstrated in the literature that low order, $n = 1$ or 2, is enough for many applications. Problems dealing with the vehicle's lateral dynamics (Wang and Wang 2020) and the quadrotor's attitude and position (Wang *et al.* 2016) demonstrate the need for a second-order model. However, the first-order model proves to be sufficient for most applications. In this study, $n = 1$ is used, and (4) can be written as

$$\dot{y}(t) = F(t) + \alpha u(t). \tag{5}$$

To calculate the system input, which is also the controller output, $u(t)$, $F(t)$ has to be approximated at each time step with:

$$F(t) \approx \hat{F}(t) = \hat{y}(t) - \alpha u(t - \tau), \tag{6}$$

where $u(t-\tau)$ is the controlled input to the system at the previous time step, and $\hat{y}(t)$ is the approximated first-order derivative of the system output. Due to noise in the real system, a direct derivative from the signal can drastically amplify the noise amplitude. So, algebraic derivative estimation (ADE) is used to perform the numerical differentiation (Mboup *et al.* 2009). ADE firstly estimates the raw signal $y(t)$ with a truncated Taylor expansion. Then, using the Laplace transform, it is expressed in the operational domain. A finite-integral form annihilator $\Pi_{k,\mu}^{N,n} = \frac{1}{s^{N+\mu+1}} \frac{d^{n+k}}{ds^{n+k}} \frac{1}{s} \frac{d^{N-n}}{dx^{N-n}} s^{N+1}$, where k, N, n, μ are integers, is applied to the truncated signal, and the *n-th* order derivative of $y(t)$ in the operational domain can be estimated. By applying the Cauchy formula for repeated integration to $\hat{y}^{(n)}(s)$, it is transferred back to the time domain with the following expression:

$$\hat{y}^{(n)}(t) = \frac{(\mu+k+2n+1)!}{(\mu+n)!(k+n)!} \int_0^T \frac{(T-\tau)^{\mu+n} \tau^{k+n}}{T^{\mu+k+2n+1}} y^{(n)}(\tau) d\tau, \tag{7}$$

where n equals 1 for the first-order derivative. Applying $k = \mu = 0$ to the above expression yields:

$$\hat{y}(t) = \frac{6}{T_{ADE}^3}\int_0^{T_{ADE}}(T_{ADE} - 2\tau)y(t-\tau)d\tau, \quad (8)$$

where $T_{ADE} = N\tau$ is the chosen sliding window size, and τ is the sample time. With this estimation, $F(t)$ can be calculated from (6) at each time step.

Closing the feedback loop with an intelligent proportional (iP) controller yields the following expression:

$$\dot{y}^*(t) = \dot{y}(t) + K_p e(t), \quad (9)$$

where $\dot{y}^*(t)$ is the reference system output, K_p is the controller gain, and $e(t)$ is the difference between actual and reference system outputs. After combining (5) and (9), the control output can be well expressed as:

$$u(t) = -\frac{F(t) - \dot{y}^*(t) + K_p e}{\alpha}. \quad (10)$$

3. PROBLEM FORMULATION AND CONTROLLER DESIGN

In this section, we study the behavior of H, S, and V threshold values under different lighting conditions and aim to choose the one that can best separate the lane marks from their backgrounds. Two assumptions are made below:

1. The vehicle is following a straight lane mark, and the reference steering angle is zero.
2. The adjustable threshold algorithm will only be applied to the straight-lane condition.

It is reasonable to state that the straight-lane following scenario is the most common operation in real life. For a curve road, as long as its radius is known, the reference steering angle can be calculated. The method presented in this study can be readily applied to such cases, but it will be left for future studies.

Instead of controlling all six thresholds in H, S, and V channels, we aim to find the most sensitive variable to the varying lighting conditions that can best separate the lane mark from its background. A straight lane marks can be modeled by two lines, one for each boundary, and the region in between is of interest. This process was done offline, and the result is shown in Fig. 1. The image to the left was originally from the camera, and the identified lane region was masked by red color for illustration purposes in Fig. 1(b).

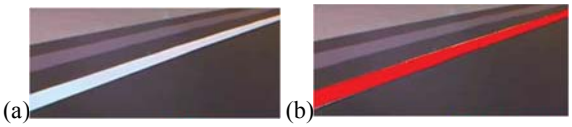
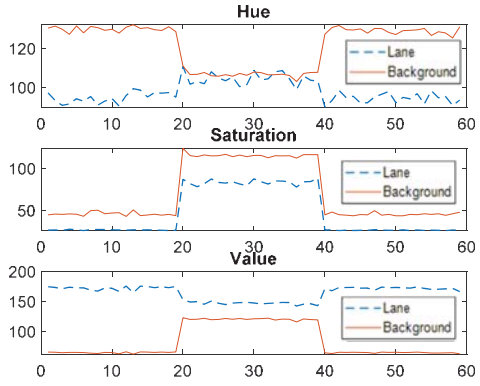
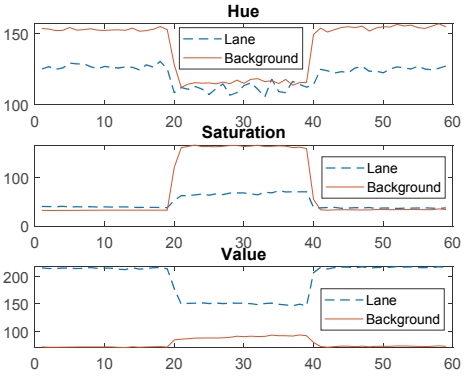


Fig. 1. (a) Original image captured by the camera, and (b) processed image with identified lane mark.

An LED light source was used in this study, and it can be set to two different brightness levels. To create varying lighting conditions, we initially turned on the light, then turned it off, and finally turned it back on. Two datasets were collected: average H, S, and V values for all pixels in the lane mark region (the red highlighted region in Fig. 1(b)) and in the background region (locations except the red region in Fig. 1 (b)). In parallel, representative results are shown in Fig. 2.



(a)



(b)

Fig. 2. Results from LED with two lighting conditions.

In Fig. 2, dotted lines and solid lines denote the average H, S, and V values for pixels in the lane mark region and background region, respectively. The results demonstrate that the average hue (H) and saturation (S) values for the two lines have intersections under different lighting conditions. It will be hard for the camera to distinguish a lane mark from its background based on H and S channels. Indeed, a well-separation of one channel out of the three is enough to distinguish the lane mark from its background, and, as shown from Fig. 2, clear separations between the solid and dashed lines can be observed from the V channel. It needs to be emphasized that this good separation in the V channel does not mean that a fixed threshold can effectively separate all pixels in the lane mark region from its background, because the information in Fig. 2 only represents the average behavior over all pixels. However, Fig. 2 gives us confidence that we can apply MFC to adjust the threshold in the V channel to accurately detect lane marks. As seen from the plots, V values of the lane mark region are higher than the background region and close to the default limit, which is 255. Thus, the upper bound of the V channel is set to 255, and the lower bound is chosen as the control variable.

Similar to what we found, the V channel information was used in literature to increase the brightness and contrast of an image

(Banik *et al.* 2018; Guo *et al.* 2014) and for shadow detection (He *et al.* 2012).

4. SYSTEM CONFIGURATION

A scaled car manufactured by Quanser, shown in Fig. 3, is used for the experimental study. It is equipped with a front camera and front lights.

Fig. 3. The scaled car used in this study.

For simplicity, only identifying the white lane (left lane) is performed in this study. The current system setup is shown in Fig. 4 with the proposed MFC enclosed in a dotted block.

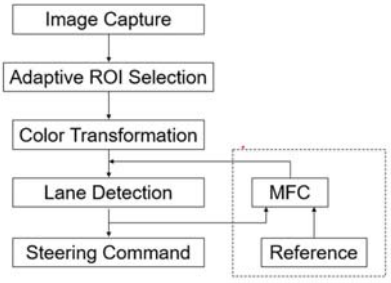


Fig. 4. Block diagram of lane detection and steering control algorithm.

An image was captured by the front camera, and only the bottom half rows and the left one-third columns of the image were selected as the ROI (the adaptive feature was not used in this study). Then, the selected region of the image was transformed from RGB space to HSV space, and only pixels within the specified thresholds in H, S, and V channels were identified as a lane mark region. The processed image from the "Lane Detection" stage is in binary form, with detected pixels having a value of 1, and other pixels having a value of 0. Finally, in the "Steering Command" stage, the identified pixels are fitted with a linear polynomial:

$$y = mx + b, \tag{11}$$

where x and y are the positions for each pixel, and m and b are the fitted polynomial parameters. Nominal parameters m_n and b_n were found offline by positioning the vehicle at the center of two lane marks. Note that the real-time fitted parameters and the offline tuned nominal parameters are from the vehicle's ego coordinate system. Their differences indicate the vehicle's deviation from the lane center. The steering angle is then calculated from the following equation:

$$\delta = \frac{K(\psi - \psi_n)}{C}, \tag{12}$$

where K is a constant gain, C is the number of columns of the binary image, and ψ and ψ_n are calculated from the actual linear fitted parameters and the nominal values, respectively. Detailed calculations for ψ and ψ_n are omitted here for brevity. This steering angle calculation method is very sensitive to pixels outside of the lane mark region. With improperly localized pixels, the linear polynomial fit cannot model a line that coincides with the actual lane mark so that the calculated steering angle command from (11, 12) will not be desirable.

With the dotted block shown in Fig. 4, the adaptive threshold method is included. As shown in Section 3, inputs to the controller are the actual steering angle command and the reference steering angle, and the controller output is the lower bound for the V channel. The constant α in the MFC was determined in a brutal-force manner. To let $\alpha u(t)$ and $\dot{y}(t)$ have close magnitude order, α is anticipated to be very small. α was tuned between 0.001 and 1 to minimize the square-mean-difference between the actual steering command and the reference steering angle, and it was determined to be 0.004. The lower bound for the V channel is the control variable, and other thresholds were chosen as the default values. System and controller parameters are summarized in Table 1.

Table 1. System and controller parameters.

Parameters	**Value**	
	Lower Threshold	Upper Threshold
Hue (H)	80	170
Saturation (S)	0	120
Value (V)	Adaptive	255
$\boldsymbol{\alpha}$	0.004	
$\boldsymbol{K_p}$	-0.001	
$\boldsymbol{m_n}$	0.51	
$\boldsymbol{b_n}$	0.54	
$\boldsymbol{\tau}$	12 Hz	
$\boldsymbol{N}$	3	
$\boldsymbol{y^*}$	0	
$\boldsymbol{K}$	0.75	

5. OFF-LINE EXPERIMENTAL RESULTS

A video from the scaled-car camera with a resolution of 410×820 was recorded to MATLAB at 12 Hz. Its length is 60 seconds, with the background light turned on for roughly 20 seconds, then turned off for roughly 20 seconds, and finally turned on again for roughly 20 seconds. The light source is from an LED. When the LED was turned off, the scaled car's front lights were turned on. The video was input offline to a Simulink-QUARC joint experiment. The resulting steering commands are shown in Fig. 5. The curves in dashed and solid lines represents the fixed threshold and the adaptive threshold outcomes, respectively. The proposed adaptive threshold method demonstrates its effectiveness in stabilizing the steering command very close to the reference value of zero degree. The fixed threshold method generates similar steering command compared to the adaptive threshold method under bright lighting conditions. However, when the light was turned

off, the steering command from the fixed threshold method became unstable and away from the reference value. The spikes shown in Fig. 5 are due to the sudden change in lighting conditions, and they only last for a short amount of time.

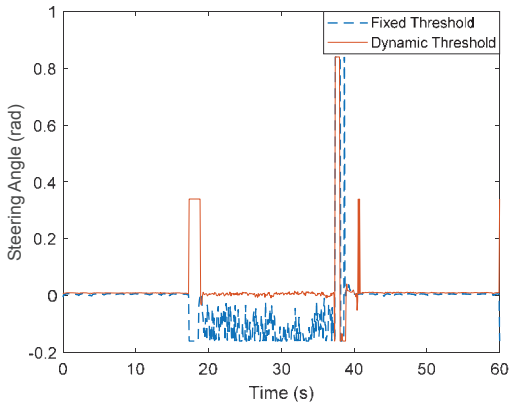


Fig 5. Steering command with and without MFC.

The stable steering command shown in Fig. 5 is a result of the adaptive threshold adjustment. The control effort from MFC can be inferred in Fig. 6. When the light was on, both methods shared the same threshold value of 140, but when the light was turned off, the adaptive threshold was increased to 210. Under this dark condition, we suspect that the front light of the scaled car, which is in white color, causes the background region in proximity of the car to appear to be brighter and have a higher value in the V channel. Thus, the lower bound of the V channel has to be higher to filter out the unwanted region. When the LED light was turned on again, the adaptive threshold went down to the value around 150, close to the initial value. In contrast, the fixed threshold stayed unchanged for the entire period. As discussed previously, it cannot handle the varying lighting conditions well.

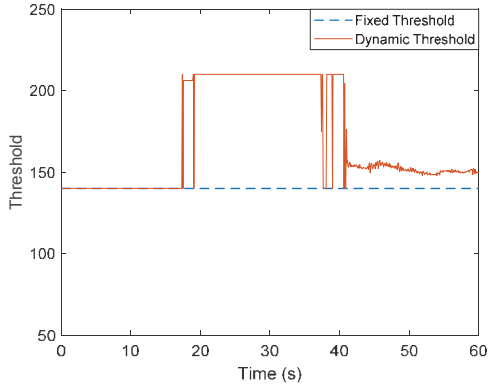


Fig. 6. Adaptive thresholds from MFC and fixed threshold.

To further demonstrate the improvement of adaptive threshold under different lighting conditions, we show real-time video captures in Fig. 7. The top three captures are from the adaptive threshold method, and the bottom three captures are with fixed thresholds. For illustration purposes, pixels being identified as lane features are shown in red. Under good lighting conditions, the red region shown in Fig. 7 accurately covered the white lane mark for both fixed and adaptive thresholds. However, with fixed thresholds, when the LED light was turned off, there were red regions outside of the white lane, and the outliers were highlighted with circles in the figure. On the other hand, the red region exactly coincided with the white lane with the adaptive threshold control, and there was nowhere in the background with red color from visual inspection.

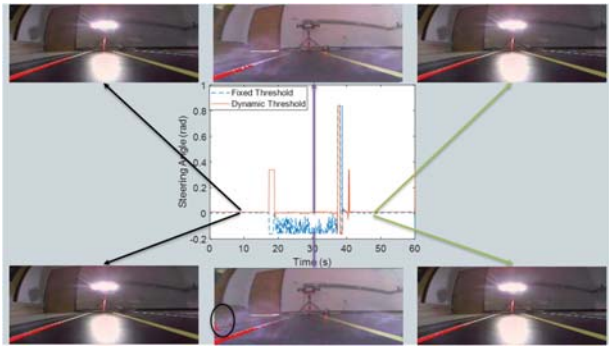


Fig. 7. Captured video frames from different lighting conditions.

To quantify the enhancement from the adaptive thresholds, the total number of identified pixels that fall out of the lane mark region were calculated. A normalized result for identified pixels that are outside of the lane mark region is shown in Fig. 8. It shows that except for the spike region due to the overexposure effect, the number of pixels outside of the lane feature is very close to zero with the dynamic thresholds. It is consistent with the observation in Fig. 7 that there is no red region in the background. On the contrary, the percentage of pixels outside of the lane feature is quite high with the fixed threshold, and these outlier pixels deteriorate the performance of autonomous vehicle steering control as shown in Fig. 5. At the overexposure moment, the image appears to be washed out. The top image shown in Fig. 8 was from an overexposed video frame, and the ROI was constrained to be the bottom left corner. It shows that nearly all the pixels were highlighted in red and wrongly considered to be the lane mark region, which is because that the overexposure effect makes the entire image appear to be close to the white lane color.

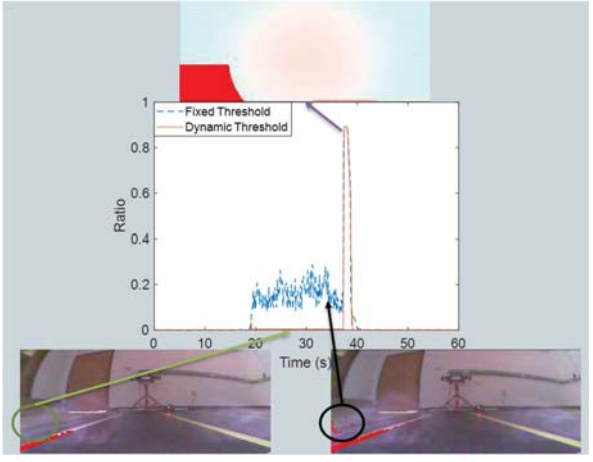


Fig 8. The ratio of pixels outside the lane region with pictures showing lane detection results and the overexposure effect.

6. CONCLUSIONS

This paper presents a novel way to adaptively adjust the V channel threshold in the HSV space under different lighting conditions for better lane detection in automated driving applications. The proposed method enables the scaled car to follow a reference steering angle under changing lighting conditions, while the fixed-threshold method failes to do so. Yellow lane detection under more lighting conditions will be performed in future studies. Testing on curved roads with known radii will also be studied in the future.

REFERENCES

Banik, P.P., Saha, R. and Kim, K.D., 2018, January. Contrast enhancement of low-light image using histogram equalization and illumination adjustment. In *2018 International Conference on Electronics, Information, and Communication (ICEIC)* (pp. 1-4). IEEE.

Borkar, A., Hayes, M. and Smith, M.T., 2009, November. Robust lane detection and tracking with ransac and kalman filter. In *2009 16th IEEE International Conference on Image Processing (ICIP)* (pp. 3261-3264). IEEE.

Chen, Y., Zhang, X., and Wang, J., 2021, Robust Vehicle Driver Assistance Control for Handover Scenarios Considering Driving Performances, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 51, Issue 7, pp. 4160 – 4170.

Fliess, M. and Join, C., 2013. Model-free control. *International Journal of Control*, *86*(12), pp.2228-2252.

Guo, J.K., Sung, C.C. and Chang, H.H., 2014, April. Improving visibility and fidelity of underwater images using an adaptive restoration algorithm. In *OCEANS 2014-TAIPEI* (pp. 1-6). IEEE.He, Y., Li, J., Wang, H., Pu, H. and Li, R., 2012, October. Adaptive vehicle shadow detection algorithm in highway. In *2012 Fifth International Symposium on Computational Intelligence and Design* (Vol. 2, pp. 240-243). IEEE.

Jin, X., Yu, Z., Yin, G., and Wang, J., 2018. Improving Vehicle Handling Stability Based on Combined AFS and DYC System via Robust Takagi-Sugeno Fuzzy Control. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, Issue 8, pp. 2696 – 2707.

Li, M., Li, Y. and Jiang, M., 2018. Lane detection based on connection of various feature extraction methods. *Advances in Multimedia*, *2018*.

Liang, M., Yue-ju, X., De-yun, K., Ke, H., Qi-Fu, L. and Kai, W., 2011, April. A hybrid H component histogram threshold and sparse field level set algorithm for litchi image automatic segmentation. In *2011 International Conference on Electric Information and Control Engineering* (pp. 1001-1004). IEEE.

Mammeri, A., Boukerche, A. and Tang, Z., 2016. A real-time lane marking localization, tracking and communication system. *Computer Communications*, *73*, pp.132-143.

Mboup, M., Join, C. and Fliess, M., 2009. Numerical differentiation with annihilators in noisy environment. *Numerical algorithms*, *50*(4), pp.439-467.

Mehrotra, R., Namuduri, K.R. and Ranganathan, N., 1992. Gabor filter-based edge detection. *Pattern recognition*, *25*(12), pp.1479-1494.

Mu, C. and Ma, X., 2014. Lane detection based on object segmentation and piecewise fitting. *TELKOMNIKA Indones. J. Electr. Eng. TELKOMNIKA*, *12*(5), pp.3491-3500.

Narote, S.P., Bhujbal, P.N., Narote, A.S. and Dhane, D.M., 2018. A review of recent advances in lane detection and departure warning system. *Pattern Recognition*, *73*, pp.216-234.

Srivastava, S., Lumb, M. and Singal, R., 2014. Improved lane detection using hybrid median filter and modified Hough transform. *Int. J. Adv. Res. Comput. Sci. Softw. Eng*, *4*(1), pp.30-37.

Vitabile, S., Pollaccia, G., Pilato, G. and Sorbello, F., 2001, September. Road signs recognition using a dynamic pixel aggregation technique in the HSV color space. In *Proceedings 11th International Conference on Image Analysis and Processing* (pp. 572-577). IEEE.

Wu, P.C., Chang, C.Y. and Lin, C.H., 2014. Lane-mark extraction for automobiles under complex conditions. *Pattern Recognition*, *47*(8), pp.2756-2767.

Wang, J., Wang, J., Wang, R., and Hu, C. 2017. A Framework of Vehicle Trajectory Replanning in Lane Exchanging with Considerations of Driver Characteristics. *IEEE Transactions on Vehicular Technology*, Vol. 66, Issue 5, pp. 3583-3596.

Wang, R., Zhang, H., Wang, J., Yan, F., and Chen, N., 2015. Robust Lateral Motion Control of Four-wheel Independently Actuated Electric Vehicles with Tire Force Saturation Consideration. *Journal of the Franklin Institute*, Vol. 352, Issue 2, pp. 645 – 668.

Wang, Y., Teoh, E.K. and Shen, D., 2004. Lane detection and tracking using B-Snake. *Image and Vision computing*, *22*(4), pp. 269-280.

Wang, H., Ye, X., Tian, Y., Zheng, G. and Christov, N., 2016. Model-free–based terminal SMC of quadrotor attitude and position. *IEEE Transactions on Aerospace and Electronic Systems*, *52*(5), pp. 2519-2528.

Wang, Z. and Wang, J., 2020. Ultra-local model predictive control: A model-free approach and its application on automated vehicle trajectory tracking. *Control Engineering Practice*, *101*, p.104482.

Zhao, K., Meuter, M., Nunn, C., Müller, D., Müller-Schneiders, S. and Pauli, J., 2012, A novel multi-lane detection and tracking system. In *2012 IEEE Intelligent Vehicles Symposium*. pp. 1084-1089.

Zhu, X., Zhang, H., Wang, J., and Fang, Z., 2015, Robust Lateral Motion Control of Electric Ground Vehicles with Random Network-induced Delays, *IEEE Transactions on Vehicular Technology*, Vol. 64, No. 11, pp. 4985 – 4995.

Zhou, X., Wang, Z., and Wang, J., 2021, Popov-H_∞ robust path-tracking control of autonomous ground vehicles with consideration of sector-bounded kinematic nonlinearity. *Journal of Dynamic Systems, Measurement, and Control*, 143(11), 111004.