E-ISSN: 2798-4664

Early Detection and Tracking of Distant Incoming Traffic using Improved Detection on Road Vanishing Point Reference for Adaptive Traffic Light Signaling

Yoanda Alim Syahbana[1)] and Yokota Yasunari [2)]
[1] *Computer Engineering Technology, Information Technology Department, Politeknik Caltex Riau, Pekanbaru, Indonesia**
[2] *Department of Electrical, Electronics and Computer Engineering, Faculty of Engineering, Gifu University, Gifu, Japan*

E-mail: [*1)]yoanda@pcr.ac.id

Abstract: Real-time monitoring is essential and influences the decision-making process of adaptive traffic light systems. During temporary road closures, only one side of the lane can be accessed, increasing the need to recognize and track oncoming vehicles. Therefore, it is crucial to detect oncoming vehicles that are far away as early as possible, as waiting for an oncoming vehicle near a traffic light may delay the signal, leading to sudden braking or an accident. The purpose of this study was to improve traffic detection and tracking, even when the traffic is still far from the traffic lights. Vanishing point as detection reference is estimated, and Region of Interest (RoI) is calculated. An evaluation is performed based on how quickly the proposed method detects oncoming traffic compared to the R-CNN method. The results show that the proposed method requires an average of 17.75 frames to detect the target vehicle, while R-CNN requires an average of 63.36 frames to detect the target vehicle. The results show that the accuracy of the proposed method depends on the number of pixel orientations when estimating the vanishing point and how accurately the RoI is defined. Therefore, the proposed method reliably supports the safety and reliability of adaptive traffic light systems.

Keywords: adaptive traffic light; Region of Interest; vanishing point; distant incoming traffic detection and tracking

1. Introduction

Temporary roadblocks that only allow traffic in one lane direction at a time require traffic controllers to manage traffic. However, traffic controllers are prone to traffic accidents. Alternatively, timing-based traffic light can be positioned to control the traffic. Nevertheless, timing-based traffic light are ineffective because they do not depend on actual traffic conditions. Adaptive traffic lights based on real-time traffic conditions are therefore recommended [1,2]. Adaptive traffic lights are connected to cameras via computers to capture real-time traffic conditions. The decision to signal depends on the traffic conditions on both sides of the road captured by the camera.

Early detection of oncoming traffic with adaptive traffic lights is a significant challenge. Early detection provides information about the decision-making process of the traffic light system to signal the appropriate traffic lights. If the judgment is delayed, the signal may be delayed, causing the oncoming vehicle to stop suddenly, which may lead to an accident. However, the scene captured by the camera is subject to perspective projection. Far inbound traffic appears at a smaller size. Detecting this small object is difficult because there are objects that are not vehicles. Furthermore, using artificial intelligence such as deep learning to detect this small object is limited due to the low resolution of vehicle images [3–5].

In this study, we proposed a method to detect oncoming vehicles by referring to the position of the vanishing point in the perspective projection. Vanishing points are estimated based on texture-based Weber Orientation Descriptor (WOD) methods. A region of interest (ROI) is then defined to focus the detection only on the lane area. Vanishing point and RoI information are used to detect and track oncoming vehicles. In this study, background subtraction is used to detect incoming traffic and Kalman filtering is used to track detected objects. We use 12 videos retrieved from In-Luck Company to experiment with the proposed method. The performance of the proposed method is compared with that of the area using convolutional neural networks (R-CNN) in terms of how quickly the method can detect incoming traffic.

This paper is organized as follows. In Section 2, video test materials and designs of proposed method are introduced. In Section 3, the experiment results are discussed. In Section 4, evaluation of the proposed method is provided. Finally, the conclusion is summarized in Section 5.

2. Methods

This study formulates the proposed method as three steps. The first step is the initialization step that has two processes. The first process is to enclose region where the main movement of traffic exist in captured scene. In this process, this study uses background subtraction method to detect foreground object followed by frame difference method to detect movement of the object. Then, the object movement is aggregated to define RoI. The second process is aimed to estimate coordinate of vanishing point from the video frame. This study uses WOD method [6] that calculates differential excitation of the frame texture features. Then, Gabor filter is used to calculate pixel orientation. The information of differential excitation and orientation from every pixel is used for voting scheme to estimate vanishing point coordinate.

The second step is aimed to focus the detection of traffic within the RoI. Similar foreground object detection in the initialization step is used to detect the traffic. Following this, the third step is aimed track the detected traffic. This step associates the detected traffic based on its movement from frame to frame. Kalman filter is used track the detected traffic based on likehood of each detection to each motion track. In addition, the motion track is selected for the traffic that getting distant from the vanishing point. Finally, the selected motion track and its detected traffic are categorized as the incoming traffic. Figure 1 shows the design of the proposed method and the following section describes details of the steps that are used in the proposed method.

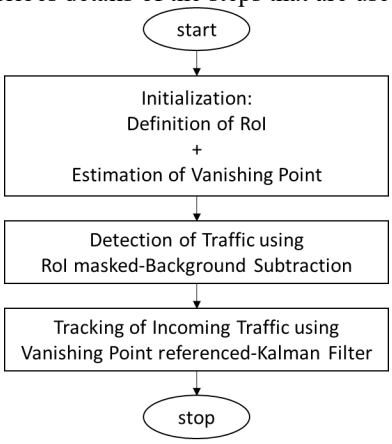


Figure 1. Design of proposed method.

2.1 Initialization Step

This study designs the initialization step as a preparation before conducting detection and tracking the incoming traffic. The first process in initialization step is defining the RoI. For RoI definition process, sequence of $I(x, y, t)$ with existence of traffic is used for RoI definition process. RoI definition process that is used in this study is similar to previously published works in [7].

The second process in initialization step is estimation of vanishing point coordinate. The vanishing point in this step is estimated from single $I(x, y, t)$ with t is preselected manually by visual observation. This study recommends selecting t without existence of traffic to obtain precise vanishing point. In this process, $I_{VP}(x, y)$ is $I(x, y, t)$ for the preselected t. In addition, $I_{VP}(x, y)$ is also convolve with median filter with size of 5x5 pixels to reduce noise.

The estimation of vanishing point is started by calculating two components of WOD: differential excitation and orientation at each pixel location. Differential excitation is calculated based on difference between center pixel intensity and average intensity of all neighbors pixel in a k x k kernel size. Differential excitation is calculated using

$$\xi_{wod}(p_{center}) = \begin{cases} \sqrt{G(p_{center})}, G(p_{center}) \geq 0 \\ 0, otherwise \end{cases}, \quad (1)$$

in which,

$$G(p_{center}) = arctan\left(\frac{p_{center} - \overline{p_{neighbor}}}{p_{center}}\right), \quad (2)$$

where $\xi_{wod}(p_{center})$, p_{center}, $\overline{p_{neighbor}}$, and $G(p_{center})$ are differential excitation for p_{center}, intensity of center pixel, average intensity of all neighbors pixel, and the intensity difference, respectively. This study predefined k=25 as size of kernel.

$\xi_{wod}(p_{center})$ is further processed by thresholding to minimize the noise that exists in the frame texture features. This study defines T=0.05 as thresholding value. Normalized value of $\xi_{wod}(p_{center})$ that is larger than T is used to estimate the vanishing point.

This study uses Gabor filter to estimate dominant orientation at each of pixel location. Kernel of Gabor filter g that is centered at (x, y) for orientation φ_n and radial frequency $\omega = 2\pi/\lambda$ is defined as

$$g_{\varphi_n}(x, y) = e^{-\frac{1}{8\sigma^2}(4a^2+b^2)}.(ia\omega - e^{c^2/2}), \quad (3)$$

where $a = x\cos\varphi_n + y\sin\varphi_n$ and $b = -x\sin\varphi_n + y\cos\varphi_n$. In this study, $\sigma = k/9$, c=2.2, and $\lambda = k\pi/10$ are a constant, similar to parameter setting in [6]. φ_n is calculated using

$$\varphi_n = \frac{(n-1)\pi}{N_\varphi} \quad n = 1,2,\ldots,N_\varphi, \quad (4)$$

where N_φ is total number of orientations. Dominant orientation for $I_{VP}(x, y)$ for each p_{center} is calculated using

$$\hat{I}_{\varphi_n}(p_{center}) = I_{VP}(p_{center}) * g_{\varphi_n}(p_{center}), \quad (5)$$

where $\hat{I}_{\varphi_n}(p_{center})$ is result of convolution between video frame and kernel of Gabor filter, and $*$ denotes convolution operator. $\hat{I}_{\varphi_n}(p_{center})$ as convolution result has real part and imaginary part. These two parts are used to calculate Gabor energy for each pixel in $\hat{I}_{\varphi_n}(p_{center})$. Gabor energy is calculated as

$$E_{\varphi_n}(p_{center}) = \sqrt{Re\left(\hat{I}_{\varphi_n}(p_{center})\right)^2 + Im(\hat{I}_{\varphi_n}(p_{center}))^2}, \quad (6)$$

where $E_{\varphi_n}(p_{center})$ is magnitude of Gabor energy at p_{center}. Finally, orientation at each of pixel location is defined as

$$\theta_{wod}(p_{center}) = Argmax_{\varphi_n} E_{\varphi_n}(p_{center}), \quad (7)$$

where $\theta_{wod}(p_{center})$ is p_{center} orientation.

The vanishing point is estimated based on result of Line-Voting Scheme (LVS). Firstly, LVS sets accumulator space with the same size as $I_{VP}(x, y)$ with initial zero value. Secondly, $\xi_{wod}(p_{center})$ and its counterpart $\theta_{wod}(p_{center})$ act as a voter that draws rays in the accumulator space. The corresponding accumulator space is increased by 1 if the rays lies over it. Finally, maximum value in the accumulator space is defined as vanishing point coordinate, (x_{vp}, y_{vp}).

2.2 Object Detection using Background Subtraction

The second step uses defined RoI from the initialization step to mask the $I(x, y, t)$. The masking process conducted by

$$I_{masked}(x, y, t) = \begin{cases} I(x, y, t), I_{RoI}(x, y) = 1 \\ 0, otherwise \end{cases}, \quad (8)$$

where $I_{masked}(x, y, t)$ is the masking result. Then, similar process of background subtraction as used in [7] is applied for all frame in $I_{masked}(x, y, t)$. $I_{FO}(x, y, t)$ is obtained from the process.

2.3 Object Tracking using Kalman Filter

$I_{FO}(x, y, t)$ result from second step is further processed to track its movement. The process is to associate detected $I_{FO}(x, y, t)$ based on its movement from frame to frame. Since this study uses video from a stationary video camera, the Kalman filter [8] can predict object tracks in each frame and determine the likehood of each detection to each track. Following this, track maintenance is also applied to update any new object or vanishing object from the video frame.

Mainly, the motion estimation process in this step follows Matlab documentation [9]. Configuration is applied for minimum detection of the blob area for 100 pixels to cope with the condition in detecting small resolution of incoming traffic in the study case.

Motion estimation step results coordinate of the detected object from each frame. However, because there is also a possibility to capture the outgoing traffic approaching the vanishing point, the result is filtered for detection that gets further from the vanishing point only.

Distance of vanishing point to detected object is calculated using Euclidean distance. Euclidean distance is calculated as

$$d(t) = \sqrt{(x_{obj}(t) - x_{vp})^2 + (y_{obj}(t) - y_{vp})^2}, \quad (9)$$

where $d(t)$ and $(x_{obj}(t), y_{obj}(t))$ are Euclidean distance of object and coordinate of detected object at frame t, respectively. Filtering incoming traffic is conducted by

$$incoming(t) = \begin{cases} 1, d(t) > d(t-1) \\ 0, otherwise \end{cases}, \quad (10)$$

where $incoming(t)$ is label for incoming traffic.

3. Result and Discussion

3.1 Initialization Step

Experiment result shows that the defined RoI can cover main road lane where most of traffic activity is existing. The defined RoI is also affected by perspective projection that which one side of the RoI become smaller as the road lane getting distant. Even though some of RoI shape irregular, the objective to exclude movement of non-vehicle object that is unpredictable in the captured scene such as shrub, grass and tree, and flag can be minimized.

Qualitative comparison shows that the estimated vanishing point from the initialization step can almost match the ground truth in straight road. It is because the LVS mainly defines the vanishing point based on major orientation of straight edge object that exists in the $I_{VP}(x, y)$. Existence of road line, road fence, pavement, and aerial utility cable influences accuracy of the estimated vanishing point. Observation of experiment result also shows that existence of straight edge of bridge and other visible roadway also makes the vanishing point is shifted in curved road.

As quantitative comparison, estimation error of the vanishing point is calculated using normalized Euclidean distance [10]. The estimation error is defined as

$$\delta = \frac{\sqrt{(x_{vp} - x_{gt})^2 + (y_{vp} - y_{gt})^2}}{\sqrt{N_x{}^2 + N_y{}^2}}, \quad (11)$$

where δ and (x_{vp}, y_{vp}) are estimation error and ground truth coordinate, respectively. δ near to 0 represents close estimation of the vanishing point to the ground truth; otherwise, δ near to 1 represents inaccuracy of estimation.

Table 1 tabulates δ for each video test material with variation of $N_\varphi \in \{9,12,18,36,180\}$. Variation of N_φ influences how accurate the estimation of vanishing point based on the orientation in Gabor filter calculation. In general, error of estimation is relatively low with maximum value of 0.1635. Increasing N_φ means increasing resolution of φ_n precision. Thus, the error estimation can be lowered. However, the processing time for high resolution orientation is also increased.

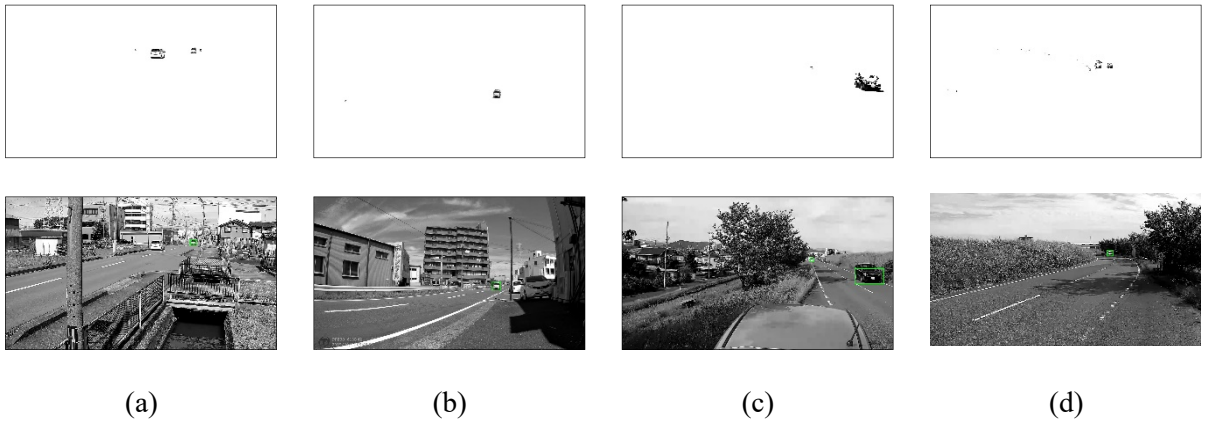
Table 1. Quantitative comparison of vanishing point estimation error (δ) for variation of N_φ.

Site	δ					$\bar{\delta}$
	9	**12**	**18**	**36**	**180**	
1	0.0134	0.0204	0.0051	0.0040	0.0080	0.0111
2	0.0947	0.0501	0.0053	0.0027	0.0171	0.0369
3	0.1026	0.0161	0.0025	0.0040	0.0149	0.0288
4	0.0282	0.0188	0.0166	0.0087	0.0059	0.0151
5	0.0538	0.0074	0.0144	0.0099	0.0149	0.0208
6	0.0328	0.0301	0.0428	0.0174	0.0251	0.0312
7	0.0364	0.0163	0.0365	0.0352	0.0115	0.0242
8	0.0221	0.0148	0.0278	0.0154	0.0180	0.0195
9	0.0081	0.0199	0.0222	0.0246	0.0290	0.0205
10	0.1635	0.0278	0.0270	0.0283	0.0398	0.0587
11	0.0390	0.0946	0.0271	0.0375	0.0444	0.0500

12	0.0241	0.0067	0.0724	0.0132	0.0150	0.0259

3.2 Detection and Tracking of Incoming Traffic

Figure 1 shows sample result of the traffic detection from the second step. The first row shows detection of $I_{FO}(x, y, t)$ inside the RoI. To achieve the study purpose, small object that appear inside the RoI is categorized as candidate of the incoming traffic. Then, the detection is further processed by Kalman filter to calculate motion estimation. The second row of Figure 1 shows detected object that getting distant from the vanishing point. These objects are defined as result of the proposed method.



(a) (b) (c) (d)

Figure 1. Sample detection of $I_{FO}(x, y, t)$ (first row) and final result of incoming traffic detection (second row): (a) Site 2; (b) Site 4; (c) Site 7; (d) Site 12.

4. Evaluation

The performance of the proposed method is benchmarked to the performance of Regions with R-CNN [11] object detector. The evaluation is conducted based on how early the method can detect the incoming traffic. First, the CNN in R-CNN is created using image input layer, 2D convolution layer for Convolutional Neural Networks, Rectified linear unit (ReLU) layer, Max pooling layer, Fully connected layer, Softmax layer, and Classification output layer for a neural network.

R-CNN processed a CIFAR-10 data set that contains 50,000 training images that will be used to train a CNN. The training images have ten categories, including automobile, that suitable for the study case. The training is conducted using Stochastic Gradient Descent with Momentum (SGDM) with an initial learning rate of 0.001. The initial learning rate is reduced every eight epochs for a total of 40 epochs training.

After ensuring R-CNN is working well for the CIFAR-10 data set, the network is also trained using self-generated data set. The data set is generated based on the video frame from the first and the second camera. Then, each image is labeled manually based on visual observation from the frame. Forty vehicle images are selected from the video frames. The image shows the front part of the vehicle that represents incoming traffic. The image is also varied in terms of size, position in the roadway, and grayscale intensity. The training is conducted using the same Stochastic Gradient Descent with Momentum (SGDM) with an initial learning rate of 0.001 for 100 epochs training. The entire implementation of R-CNN used in this study follows Matlab documentation [12] with some modifications to accommodate study case conditions.

The benchmarking process is started by selecting sequential frames as object of evaluation. The sequential frames are selected manually from each the video test material, started at t_{start}. The sequential frames show the movement of incoming traffic from the ground truth vanishing point for duration of 10 seconds that is defined by t_{end}. The frontmost traffic is defined as detection target vehicle. These sequential frames are processed by the proposed method and R-CNN. The benchmarking process compares how early both methods detect the incoming traffic by confirming the detection result visually that is defined by t_{detect}.

Table 2 summarizes detection result for the benchmarking process. In general, the proposed method has earlier detection of incoming traffic than the R-CNN method. For twelve the video test materials in this study case, the proposed method requires average of 17.75 frames to detect the target vehicle while the R-CNN requires average of 63.36 frames to detect the target vehicle. For Site 9, R-CNN method cannot detect the target

vehicle because until t_{end}, the target vehicle is too small to be detected. The result shows that R-CNN requires larger size vehicle image to ensure its recognition as a vehicle. For example, in Site 1, the vehicle is detected if its minimum size is 96x200 pixels. In Site 8, vehicle is detected if its minimum size is 141x176pixels. For Site 12, R-CNN method has earlier detection than the proposed method. It is because the detected vanishing point from the proposed method exists in the center of roadway due to S-curved characteristic of the roadway. As shown in Figure 3: (l), incoming car needs to pass the vanishing point first before can be detected as incoming traffic.

Table 2. Benchmarking result on detection of incoming traffic.

Site	t_{start}	t_{end}	t_{detect}	
			Proposed method	**R-CNN**
1	870	1070	880	890
2	360	660	370	437
3	120	420	139	202
4	4200	4500	4211	4276
5	1110	1410	1121	1321
6	600	900	644	672
7	1800	2100	1810	1817
8	1080	1280	1090	1120
9	3780	4080	3792	*
10	1080	1380	1094	1132
11	4500	4800	4538	4545
12	690	990	714	695

*until t_{end}, R-CNN cannot detect the target vehicle

5. Conclusions

The primary purpose of the research was successfully achieved. Improvements to detecting and tracking distant oncoming vehicles based on the vanishing point criterion have been proposed. We showed that the proposed method could define RoI and estimate the vanishing point. The proposed method achieves earlier detection compared to using R-CNN. The results also show that the performance of the proposed method depends on the RoI definition and the number of pixel directions in the Gabor filter computation, which affects the vanishing point accuracy.

Acknowledgement

Special thanks to In-Luck Company for data collection support.

Reference

[1] Sahar Araghi, Abbas Khosravi, Douglas Creighton, "A review on computational intelligence methods for controlling traffic signal timing", Expert Systems with Applications, Vol. 42, No. 3, pp. 1538–1550, Feb. 2015.

[2] Fushi Lian, Bokui Chen, Kai Zhang, Lixin Miao, Jinchao Wu, Shicao Luan, "Adaptive traffic signal control algorithms based on probe vehicle data", Journal of Intelligent Transportation Systems, Vol. 25, No. 1, pp. 41–57, 2021.

[3] Michal Koziarski, Boguslaw Cyganek, "Impact of low resolution on image recognition with deep neural networks: An experimental study", International Journal of Applied Mathematics and Computer Science, Vol. 28, No. 4, pp. 735–744, Dec. 2018.

[4] Lixia Xue, Xin Zhong, Ronggui Wang, Juan Yang, Min Hu, "Low - resolution vehicle recognition based on deep feature fusion", Multimedia Tools and Applications, Vol. 77, pp. 27617–27639, 2018.

[5] Wilman W. W. Zou, Pong C. Yuen, "Very low resolution face recognition problem", IEEE Transactions on Image Processing, Vol. 21, No. 1, pp. 327-340, Jul. 2011.

[6] Weibin Yang, Xiaosong Luo, Bin Fang, Daiming Zhang, Yuan Yan Tang, "Fast and accurate vanishing point detection in complex scenes", 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 93-98, Oct. 2014.
[7] Yoanda Alim Syahbana, Yasunari Yokota, "Detection of Congested Traffic Flow during Road Construction using Improved Background Subtraction with Two Levels RoI Definition", International ABEC, pp.71-76, 2021
[8] Greg Welch, Gary Bishop, "An Introduction to the Kalman Filter", 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 1–16, 2001.
[9] Matlab Motion-Based Multiple Object Tracking Available online: https://de.mathworks.com/help/vision/ug/motion-based-multiple-object-tracking.html.
[10] Peyman Moghadam, Janusz A. Starzyk, W. S. Wijesoma, "Fast vanishing-point detection in unstructured environments", IEEE Transactions on Image Processing, Vol. 21, No. 1, pp. 425–430, July 2011.
[11] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587, Sep. 2014.
[12] Matlab Train Object Detector Using R-CNN Deep Learning Available online: https://de.mathworks.com/help/vision/ug/object-detection-using-deep-learning.html.