JOIN (Jurnal Online Informatika)
Volume 6 No. 2 | December 2021: 138-144
DOI: 10.15575/join.v6i2.671

p-ISSN: 2528-1682
e-ISSN: 2527-9165

Vehicle Tracking to Determine Position in The Parking Lot Utilizing CCTV Camara

Adi Suheryadi[1], Willy Permana P[2], Anis Al Hilmi[3], Reza PY[4], A Sumarudin[5], Firdaus[6]
[1-5] Department of Informatics, Politeknik Negeri Indramayu Indramayu, Indonesia
[6]Department of Electrical Engineering, Universitas Islam Indonesia Yogyakarta, Indonesia

Article Info

Article history:

Received December 03, 2020
Revised February 09, 2021
Accepted July 14, 2021
Published December 26, 2021

Keywords:

Computer vision
Image
Parking position
Vehicle tracking

ABSTRACT

Traveling to a place using a private vehicle is an activity that many people do when visiting an area. The visitors leave their cars in several parking lots within a certain period. The resulted, them having difficulty finding vehicles in the parking lot. This study aims to assist parking service users in finding cars parked at the parking location using CCTV cameras. Apart from being used as a security system, cameras have installed in the parking lot can also be used to track the visitor's cars to the point where they park. The proposed method consists of three large blocks: background subtraction, vehicle recognition, and vehicle tracking. Results this study obtained in the test include the accuracy for the vehicle tracking process of about 91.5%, with a true positive rate of approximately 81.12%, and vehicle recognition about 70%.

Corresponding Author:

Adi Suheryadi,
Department of Informatics,
Politeknik Negeri Indramayu,
Jl. Loh Bener lama No. 08 Indramayu, Indonesia
Email: adisuheryadi@polindra.ac.id

1. INTRODUCTION

One of the critical problems in the Intelligent Transportation System (ITS) is the Smart Parking (SPS) system [1]. SPS is implemented in many environments and has a variety of problem-solving capabilities, including time efficiency while drivers are looking for a parking space and others. Along with developing parking management system technology, vehicle production is constantly increasing yearly, as presented by OPEC. The number of vehicles worldwide is predicted to increase significantly from 841 million cars in 2008 to more than 1.6 billion cars in 2035 [2]. Therefore, free space to provide adequate parking areas is essential. On the other hand, the use of parking lots that are not adequately arranged has a very significant effect on the arrangement of the urban regions.

In big cities with limited parking space, many drivers do not get a parking space for their vehicles, so they use sidewalks to park their cars. It inflicts several traffic problems such as congestion and narrowing the space for pedestrians. The government seeks to provide several centralized parking points to resolve vehicle parking problems and organize smart city development. The areas that have centralized parking points include office or government areas and tourism areas.

With a large and centralized parking area, finding an available parking location is a difficult job, as is finding a parking space for vehicles parked at that location, especially in public transportation such as tourist buses which carry many passengers. Therefore, we need a parking system management to assist users in determining parking locations and finding the place of their vehicles.

Various studies related to SPS have been conducted which discuss the search for parking slots and parked vehicles. Kennecth et al. [3] tracked vehicles using particle filters to overcome occlusion and maneuvers. Linsin le et al., [4] propose tracking parking slots based on learning, using Parking-Slot Detection based on Learning (PSD). Another study came from Lin Zang et al. [5]. They developed a deep convolutional neural network (DCNN) based slot detection approach, which takes the surrounding image as input by identifying all the marking points on the input image and classifying the local images formed by the tagging point pairs. Vargeshe et al. [6] developed a machine learning approach to detect free space in parking spaces that regulates the mana boundary of each predetermined area, extending to the more general scenario of unregulated parking spaces. From the various studies above, each researcher focuses on automatically determining available parking spots and has not yet completed finding vehicles in that parking lot.

Finding vehicles in public parking lots is solved using embedded sensors [7], [8]. In Lee et al. [9], a wireless sensor network involves motion sensors under space. In our study, we propose a method of finding a vehicle in a parking lot through a vehicle tracking system to the parking point using pattern recognition and a machine learning process by utilizing existing CCTV cameras.

2. METHOD

In the proposed method, there are three significant steps, consisting of background subtraction, vehicle recognition, and vehicle tracking. Each of these steps has several processes, as shown in Figure 1.

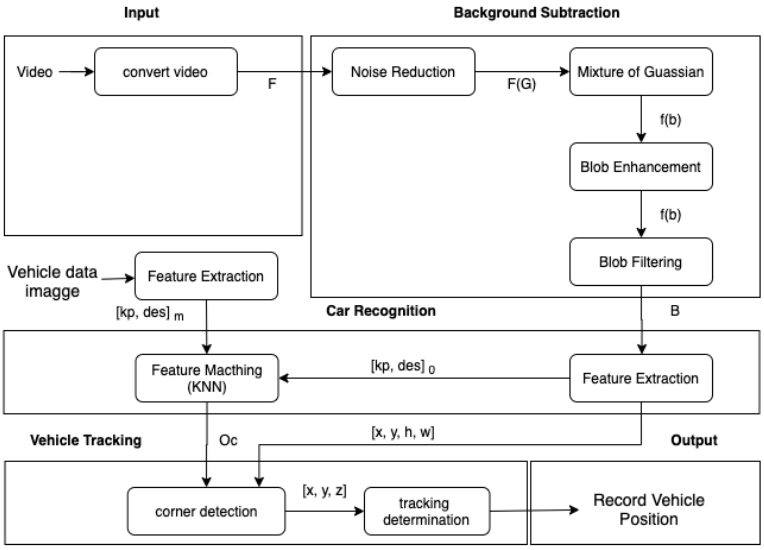

Figure 1. Block Diagram of vehicle tracking

2.1. Background Subtraction

The first process performed on the background subtraction stage is noise reduction. In this process, the noise in the frame (F) resulting from video conversion is reduced; before this process, F needs to be converted into a gray image first. The noise reduction process aims to prevent the background reduction process from detecting multiple failure blobs, wherein its implementation, noise reduction at F, uses a Gaussian filter [10]. The following process after getting the noise reduction results is a Mixture of Gaussian (MoG). The MoG can isolate the background and objects by modeling each pixel in the image based on a mixed gaussian distribution. The background is assumed to be a color with the highest value distribution [11], [12]. The MoG process leaves interesting objects in Figure F (b) as in Figure 2. However, so far, the MoG results still go noise in the image, so there are still failed blobs. Hence it is necessary to remove the failed blob using a morphological filter [13].

$$F(b) \cdot G = (F(b) \oplus G)\, \Theta\, G \tag{1}$$

Where G is the masking matrix of the morphological filter dilation - closing, the shape of the matrix is an ellipse.

The last process is blob filtering. In this process, an external scanning of the blob contours produced from the blob enhancement process is carried out, as seen in equation (4). Where B is a group of lumps resulting from increasing the blob $(F(b))$. B has four types of attributes, consist of$[Xo_i, Yo_i, Wo_i, Ho_i]$.$[Xo_i, Yo_i]$ are the coordinates of the blob location in the corner point while $[Wo_i, Ho_i]$ is the size of the blob. In contrast, thW and thH are threshold values as the object of boundary evaluation.

$$\prod_{i=0}^{n} B_i = \prod_{i=0}^{n} getBlobArea(\,F(b)_i) \tag{2}$$

$$\prod_{i=0}^{n} B_i = \prod_{i=0}^{n} [Xo_i, Yo_i, Wo_i, Ho_i] \tag{3}$$

$$B' = \begin{cases} (Wo_i \geq thW) \wedge (Ho_i \geq thH), [Xo_i, Yo_i, Wo_i, Ho_i] \\ \qquad\qquad [\emptyset] \end{cases} \tag{4}$$

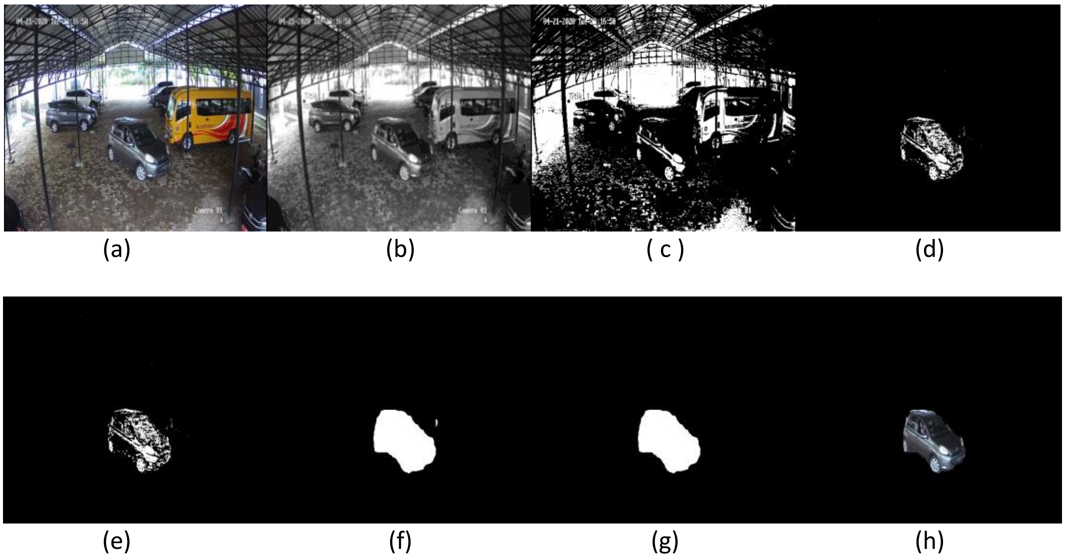

Figure 2. Background Subtraction Result: (a) original frame; (b) gray level; (c) image threshold; (d) MoG result without reduction noise; (e) MoG result with reduction noise; (f) blob enhancement; (g) blob filtering; (h) result of background subtraction

2.2. Vehicle Recognition

In the vehicle recognition stage, there are two main processes: feature extraction and feature matching. This stage aims to provide a label to the interest object so that the following process on the video no longer requires recognition but directly on the tracking object. The first process in introducing this vehicle is feature extraction using SIFT (Scale-Invariant Feature Transform) [14]. SIFT is based on the Difference-of-Gaussian (DoG) Operator which is a Laplacian-of-Gaussian (LoG) approach. The advantage of the SIFT method is that it can perform feature searches with different scales and sizes. Therefore, the method is considered the most accurate feature extraction method. The feature extraction result is $[kp, des]$, where kp is the key point from the extraction result, and des is the description of the object whose key point is taken. Object Descriptor is an array containing the nearest neighbor extraction value of 16x16 around the detected feature and segmenting the area into sub-blocks to produce 128 bin data. This value is helpful for the feature recognition process. The results of feature detection can be seen in Figure 3.

Figure 3. Feature Extraction with SIFT

Furthermore, to be able to provide the label, the feature matching process is carried out. The matching feature process will compare the key point data held by the current vehicle with the existing vehicle dataset. The feature matching process uses one of the Fast Approximate Neighborhood Neighbors (FANN) based on K-NN [15], with the nearest neighbor algorithm search method using kd-tree [16]. In this algorithm matching scheme, the nearest neighbor and second closest neighbor in each feature description (from the first image feature) are searched (in the second image feature). Furthermore, the ratio of the nearest neighbor to the second nearest neighbor is calculated for each feature description, then set the threshold value to filter the ratio value. The results of vehicle recognition can be seen in Figure 4.

Figure 4. Feature Matching with FANN

2.3. Vehicle Tracking

The last stage in this research is vehicle tracking. This stage aims to supervise objects towards the parking lot, and at the end of this stage, the vehicle's location in the parking lot is recorded. Vehicle Tracking is done by verifying the location of the candidate object (Op) and the object of interest in the next frame. The tracking results on the previous frame are used as information to determine the position of the next object. To get the centroid of Blob (B), that is shown at the formulation in equation 5.

$$cO = [x + \left|\frac{w}{2}\right|, y + \left|\frac{h}{2}\right|] \tag{5}$$

Figure 5. result of vehicle tracking

The procedures in the vehicle tracking process can be seen on algorithm 1

Algorithm 1. Procedure vehicle tracking

```
Procedure vehicleTracking (in/out Op : obj.Blob, in B: obj.Blob, F: Set of Frame )
{IS.  Op tracking result tracking in the previous blob, centroid and label, B is a current blob
FS. interest object set on Op}
Dictionary
[Xp, Yp, Xc, Yc] : centroid of previous and current blob
D : distance of Centroid\
Function FExtraction(in B: obj.Blob)
Function FMatching(in Kpb, train.Kp : key point)
Function BSubtraction(in F : frame)
Algorithm
If Op = 0 then
    [kpb, desb] ← FExtraction(B)
    Op ← FMatching(Kpb, train.Kp )
else
    [Xp, Yp] ← Op.getCoordinate()
    [Xc, Yc] ← getCentroid(B)
    D ← math.sqrt(|Xp^2-Xc^2|+|Yp^2-Yp^2|)
    if D <= thD then
        append([Xc, Yc], Op)
For each of F do
    B ← BSubtraction (F)
    vehicleTracking(Op , B)
```

3. RESULTS AND DISCUSSION

This study used 5766 frames from ten CCTV videos in the parking area to conduct vehicle recognition and vehicle tracking stages tests. Vehicle recognition and tracking testing use a confusion matrix [17], which follows equation 6.

$$Accuracy = \frac{(TN+TP)}{(TN+FP+FN+TP)}\ x\ 100 \quad (6)$$

$$TPR = \frac{TP}{(TN+FP+FN+TP)}\ x\ 100 \quad (7)$$

Where TP is True Positive, true positive occurs when the system detects a vehicle movement, and in the actual condition, the vehicle is moving too. TN is True Negative when the system does not detect vehicle movement, but there is no moving vehicle in actual condition. FP is the False Positive, which is when the system detects a car moving, but there is no moving vehicle in actual conditions. FN is False Negatives, that is when the system does not detect the car's movement, and in actual condition, there is a vehicle moving.

3.1. Vehicle Tracking

The results of the experiment vehicle tracking in the parking area are as follows in table 1.

Table 1. Vehicle tracking testing results

No	Video	TN	FP	FN	TP	Number of Frames
1	Video1	0	0	65	153	218
2	Video2	0	0	60	234	294
3	Video3	271	0	128	530	929
4	Video4	7	0	97	520	624
5	Video5	39	0	74	237	350
6	Video6	84	0	78	404	566
7	Video7	57	0	154	210	421
8	Video8	75	0	127	426	628
9	Video9	34	0	161	609	804
10	Video10	34	0	179	754	933
Total		600	0	1088	4677	5766

Based on the experimental results in table 1, the accuracy of successful vehicle tracking can be calculated as True positive rate (TPR) is 81.12 %, from 4677 frames is True Positive divide of all frame 5766. While accuracy in vehicle tracking is 91,5 %, the accuracy calculation is following equations 6 and 7. Based on the experiment above, it can be concluded that the vehicle tracking was successful, but there is still a possibility of failure, shown in figure 6. One of the leading causes is that the result of the background reduction process is less than perfect, so it leaves noise or objects from one to another frame that cannot be adequately detected. This can occur due to rapid changes in lighting, movement of objects other than a vehicle, and vehicle objects that have stopped for a long time.

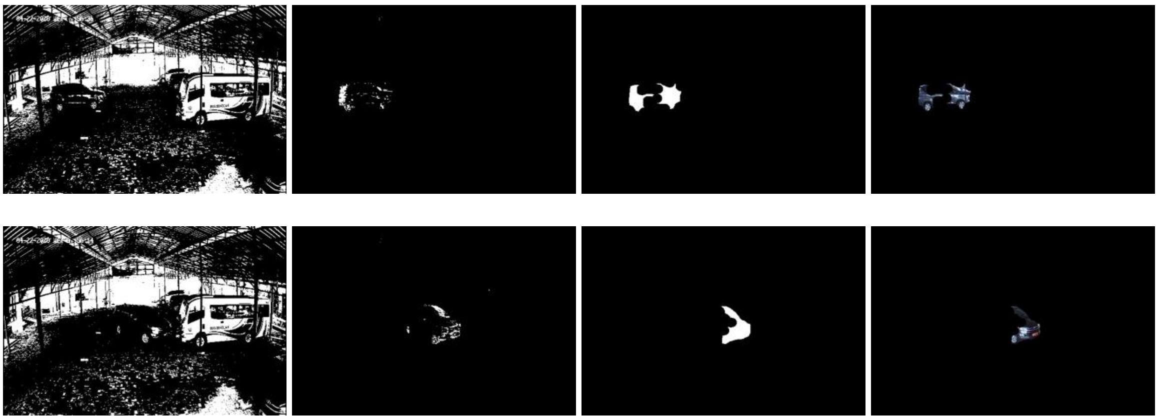

Figure 6. Failure of vehicle tracking

3.2. Vehicle Recognition

The results of the experiment vehicle tracking in the parking lot are as follows in table 2.

Table 1. Vehicle recognition results

No	Video	Recognition Result	No	Video	Recognition Result
1	Video1	True	6	Video6	False
2	Video2	True	7	Video7	True
3	Video3	True	8	Video8	False
4	Video4	True	9	Video9	True
5	Video5	True	10	Video10	False

Based on the experimental results in table 2, the accuracy of successful vehicle tracking can be calculated as True positive rate (TPR) is 70 %. The calculation of accuracy is following equation 2. Based on the above experiment, it can be concluded that the results of vehicle recognition are pretty, but there is still a possibility of failure, shown in figure 7. Like vehicle tracking cases, the cause of recognition failure is that the background reduction process is less than perfect, so that the result of the key point cannot be compared with the key point in the data set.

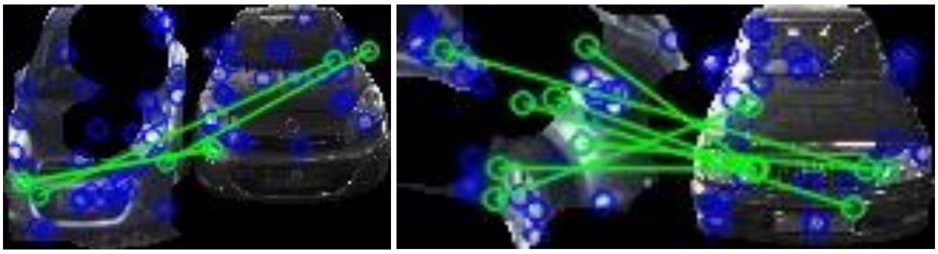

Figure 7. Failure of vehicle recognition

4. CONCLUSION

This paper provides an alternative method of locating a vehicle in a parking lot using a tracking approach. We have several block processes consisting of background reduction, vehicle recognition, and tracking. The methods used include MoG for background reduction, shift as a feature extraction method, and FANN as feature machining. We added a few steps to improve the pre-processing and tracking processes. The results obtained in the test show that the accuracy for the vehicle tracking process is around 91.5%, with a true positive rate of about 81.12% and vehicle recognition around 70%.

5. REFERENCES

[1] A. O. Kotb, Y. Shen, and Y. Huang, "Smart parking guidance, monitoring and reservations: a review," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 2, pp. 6–16, 2017.

[2] B. Turner, "The Statesmans Yearbook: The Politics, Cultures and Economies of the World 2012," *Organ. Pet. Export. Ctries.*, pp. 70–71, 2012.

[3] K. T. K. Teo, R. K. Y. Chin, N. S. V. K. Rao, F. Wong, and W. L. Khong, "Vehicle tracking using particle filter for parking management system," in *2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology*, 2014, pp. 193–198.

[4] L. Li, L. Zhang, X. Li, X. Liu, Y. Shen, and L. Xiong, "Vision-based parking-slot detection: A benchmark and a learning-based approach," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, 2017, pp. 649–654.

[5] L. Zhang, J. Huang, X. Li, and L. Xiong, "Vision-based parking-slot detection: A DCNN-based approach and a large-scale benchmark dataset," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5350–5364, 2018.

[6] A. Varghese and G. Sreelekha, "An Efficient Algorithm for Detection of Vacant Spaces in Delimited and Non-Delimited Parking Lots," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4052–4062, 2019.

[7] T. Lin, H. Rivano, and F. Le Mouël, "A survey of smart parking solutions," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3229–3253, 2017.

[8] F. Al-Turjman and A. Malekloo, "Smart parking in IoT-enabled cities: A survey," *Sustain. Cities Soc.*, vol. 49, p. 101608, 2019.

[9] S. Lee, D. Yoon, and A. Ghosh, "Intelligent parking lot application using wireless sensor networks," in *2008 International Symposium on Collaborative Technologies and Systems*, 2008, pp. 48–57.

[10] F. Chen and J. Ma, "An empirical identification method of Gaussian blur parameter for image deblurring," *IEEE Trans. signal Process.*, vol. 57, no. 7, pp. 2467–2478, 2009.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

[11] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-based surveillance systems*, Springer, 2002, pp. 135–144.
[12] S. S. Mohamed, N. M. Tahir, and R. Adnan, "Background modelling and background subtraction performance for object detection," in *2010 6th International Colloquium on Signal Processing & its Applications*, 2010, pp. 1–6.
[13] P. T. Jackway and M. Deriche, "Scale-space properties of the multiscale morphological dilation-erosion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 1, pp. 38–51, 1996.
[14] S. A. K. Tareen and Z. Saleem, "A comparative analysis of sift, surf, kaze, akaze, orb, and brisk," in *2018 International conference on computing, mathematics and engineering technologies (iCoMET)*, 2018, pp. 1–10.
[15] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2227–2240, 2014.
[16] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, 1977.
[17] W. Zhu, N. Zeng, N. Wang, and others, "Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations," *NESUG Proc. Heal. care life Sci. Balt. Maryl.*, vol. 19, p. 67, 2010.