© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International Journal of Integrated Engineering

Automatic Urban Road Users' Tracking System

Ma'moun Al-Smadi[1*], Khairi Abdulrahim[2], Kamaruzzaman Seman[2], Rosalina Abdul Salam[3]

[1]Electrical Engineering Department,
Al- Balqa' Applied University, JORDAN

[2]Faculty of Engineering and Built Environment,
Universiti Sains Islam Malaysia Negeri Sembilan, Nilai, 71800, MALAYSIA

[3]Faculty of Science and Technology,
Universiti Sains Islam Malaysia Negeri Sembilan, Nilai, 71800, MALAYSIA

*Corresponding Author

DOI: https://doi.org/10.30880/ijie.2021.13.01.014
Received 2 June 2020; Accepted 28 December 2020; Available online 30 January 2021

Abstract: This paper presents a Dynamic Gradient Pattern (DGP) based on Kalman filtering technique for urban road users tracking. DGP technique is proposed to enhance rigid object descriptive ability for improved verification. DGP descriptor along with weighted centroid was integrated with a Kalman filtering framework to enhance data association robustness and tracking accuracy. To handle multiple objects tracking, a DGP verification approach is addressed based on normalized Bhattacharyya distance. The proposed technique achieves a closer trajectory for rigid body movement. The DGP descriptor can discriminate the objects correctly, and it overcomes the partial occlusion and misdetection by verifying object location using the normalized Bhattacharyya distance between DGP features. Experimental evaluation is performed on urban videos that include a slow-motion temporary stop and partial occlusion. The experimental results demonstrate that the detecting and tracking accuracy are above 98.08% and 97.70% respectively.

Keywords: Motion detection, dynamic gradient pattern, object tracking, kalman filtering

1. Introduction

Traffic monitoring is an important component of the Intelligent Transportation System (ITS). Conventional techniques use active sensors, such as radar, infrared or underground sensors. Due to rapid advances in modern technologies with falling costs and massive computing power and storage, computer vision techniques have gained extensive use in ITS. However, it is difficult to localize and track targets precisely in urban road videos due to slow motion, temporary stop and partial occlusion [1]. The data association of fast isolated objects is feasible for tracking, and it is somehow reliable in some slow-motion situations [2]. In recent decades, computer vision attracts many researchers to contribute effectively to ITS applications. Vision-based traffic surveillance provides significant advantages with a non-intrusive installation with wide coverage and many applications. This paper focuses on applications of the proposed system in urban traffic videos.

The framework of vision-based surveillance usually consists of three major phases: detection, recognition and tracking [1].Most existing techniques perform detection using motion segmentation that includes frame differencing, background subtraction, or optical flow[2, 3]. Accuracy and robustness of segmentation have great importance in recognition, tracking, and higher-level processing [4]. Background subtraction can be classified into parametric, nonparametric, and predictive technique [1, 5]. Parametric techniques use a single uni-modal probability density function to model background like running Gaussian average, temporal median filter, sigma-delta filter or Gaussian Mixture

**Corresponding author: ma_smadi@bau.edu.jo*

Model (i.e., GMM) [6, 7]. Nonparametric techniques like Kernel Density Estimation (KDE) and codebook model can handle arbitrary density functions[8]. Finally, predictive techniques employ predictive procedures to predict the state dynamic of each background pixel. Kalman filtering, Wiener filter, autoregressive models and eigen-background are examples of such techniques [9-11].

Recognizing moving objects can be performed using the appearance-based technique [12, 13] that require prior knowledge and high computation. It uses visual information like color, texture, and shape in detecting vehicles. Feature-based [] techniques use coded descriptors to characterize the visual appearance of the vehicles. A variety of features have been used in vehicle detection like Scale Invariant Feature Transformation (SIFT) [10], speeded up Robust Features (SURF), Histogram of Oriented Gradient (HOG), and dynamic Gradient Pattern (DGP)[14-16].

After detection, object tracking associate vehicles in consecutive frames. It can be classified into three main categories: model-based, region-based, and feature-based tracking. Model-based tracking uses prior knowledge to model the target [1, 17]. Multi-view and deformable template model were used in[17, 18]. 2-D geometrical features, edges, image intensity, or gradient are used for3-D model reconstruction[19]. Deformable model-based tracking[19, 20]and constrained multiple-kernel tracking[18] are examples of dynamic 3-Dmodelling. Region-based Tracking detect objects silhouette as simple geometric shape characterized by area, coordinates, centroids, edges, contour, or intensity histogram etc. Feature-based tracking performs matching using representative features in transformed space. Earlier techniques used corner and edge features[21]. Several techniques propose feature descriptors like SIFT [22], SURF [23], HOG [24] and Haar-like features [25] for vehicle tracking.

Tracking techniques require prediction and data association that use Kalman filter or Particle filter [25]. Kalman filtering is used to estimate the object location in the new frame [26].Extended Kalman-filtering was used in [27] to track the 3-D vehicle model. In [26],Kalman filter was used to track vehicle shape based on its location, speed, and length. Kalman filtering was used to integrate vehicle parts tracking in image plane [28]. In [29] Kalman filtering was adopted using vehicle coordinates and unit displacement of center of mass together with the dimensions and unit displacement of tracking region. Detection by tracking was used in [30], they estimate vehicles trajectories by Kalman filter.

Particle filter is a sequential Monte Carlo technique that estimates the latent state variables of a dynamical system [31]. In [29], projective particle filter was combined with a mean-shift algorithm to track the color histogram of the vehicle. A hybrid mean-shift and particle-filtering approach was developed in [32]to deal with partial occlusions and background clutter. The work in [33] employed particle filter in Bayesian estimation for vehicle tracking in urban environments. Vehicle tracking in [34] fuse several cues in particle filter, which include color, edge, texture, and motion constrained. The tracking technique in [35] is based on spatial and temporal coherence of particles. Vehicle tracking in [36] uses the similarity between color histogram to identify vehicle particle. In [24], RDHOG was integrated with the particle filter framework (RDHOGPF) to improve the tracking robustness and accuracy.

2. Proposed Framework

The proposed tracking system shown in Fig. 1, utilize weighted centroid and Dynamic Gradient(DGP) [16] verification to improve tracking accuracy. Initially, Weighted Sigma Delta Estimation-Cumulative Frame Differencing (WSDE-CFD) bimodal is used to extract foreground motion [2]. It models both background and foreground separately and combines them spatially. Morphological post-processing [17] is applied to enhance the detection mask. Next, the objects within the Region of Interest (ROI), with sufficient sizes are extracted. Then the binary mask and its grayscale image are used to find the weighted centroid and the dynamic gradient pattern. Finally, a separate Kalman filter is used to track each object, while DGP descriptor is used to verify miss detection and partial occlusion.

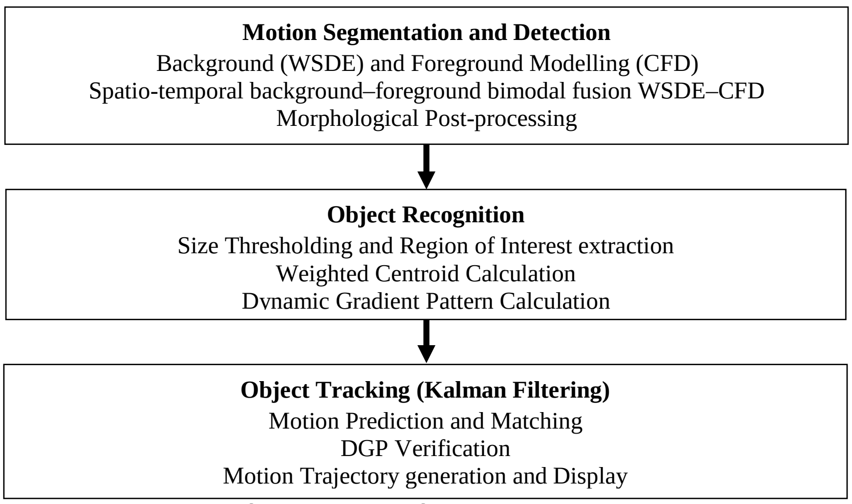


Fig. 1 - The flow diagram of the proposed tracking system

Kalman filter is used to predict the object location in the current frame based on its location and dynamics in the previous frame. To improve tracking procedure, objects are represented by their weighted centroid and DGP descriptor that provide better localization and discrimination. Miss detected objects due to size limitation or partial occlusion are verified by minimizing the normalized Bhattacharyya between DGP descriptor through consecutive frames.

2.1 Urban Road User's Detection

Motion segmentation is an application dependent task that requires several important considerations. For tracking purpose, it is necessary to generate a single track for each object to describe its motion path within the ROI. The recognition and tracking process must operate whenever an object enters the tracking zone from any street end, to work on all directions of traffic flow, also tracking initialization should work whenever detections are clearly visible without occlusion.

Morphological post-processing is applied through opening and closing followed by a morphological filling. After that, the objects are detected if more than 50% of the object is within the tracking ROI, and their size is more than a specific threshold. The threshold is set to 550 pixels to make a tradeoff between small object size and being insensitive to segmentation noise. The detected objects are represented by their weighted centroid and DGP descriptor. Fig.2illustrates the detection steps for i-LIDS datasets. It shows the frame, background model and foreground detection result.

a b c

Fig. 2 - Detection and Tracking Stages for Easy Sequence from i-LIDS Dataset; (a) Current Frame, (b) Background Estimation and (c) Foreground Mask

2.1.1 Weighted Centroid

In general, a fast localization for object tracking use center of mass (i.e., centroid) of a binary mask [26]. The basic centroid is the average of the binary mask values and their respective positions. In this work, the weighted centroid is used to further enhance the localization by incorporating the grayscale values of the detected objects in the centroid calculation as follows:

$$x = \frac{\sum_{r=1}^{m}\sum_{c=1}^{n} M(r,c) \times I(r,c) \times c}{\sum_{r=1}^{m}\sum_{c=1}^{n} M(r,c) \times I(r,c)} \tag{1}$$

$$y = \frac{\sum_{r=1}^{m}\sum_{c=1}^{n} M(r,c) \times I(r,c) \times r}{\sum_{r=1}^{m}\sum_{c=1}^{n} M(r,c) \times I(r,c)} \tag{2}$$

Where *x, y*are the center of mass coordinates, *r,c* are the row and column index respectively, *M(r, c)*is the detection mask binary value and *I(r, c)* is the grayscale intensity value at location*(r, c)*. The use of weighted centroid provides a more accurate localization, especially under pose and orientation variation, by taking into account the grayscale distribution of the tacked object. The output of motion segmentation is a binary mask enclosed by a bounding box that will be used to perform data association and tracking through consecutive frame sequence.

2.2 Urban Road Users Tracking

The tracking process establishes temporal consistency between consecutive frames. As the video frame rate increases, the motion between consecutive frames is limited. Thus, the velocity of moving objects can be assumed constant [24], allowing the use of linear discreate-time (constant speed) Kalman filter to predict motion on the image plane. The formulation of the process and measurement models for constant speed Kalman filter is defined as follow:

$$x_k = A_{k-1}x_{k-1} + Bu_k w_{k-1} \tag{3}$$

$$y_k = H_k x_k + v_k \tag{4}$$

Where, x_k and y_k are the k^{th} state vector and measurement vector. A_k and H_k are the transition matrix and the measurement matrix. w_k and v_k are the process noise and the measurement noise that are assumed to be independent white Gaussian noise with zero mean. The old state $x_{k-1}$is propagated into the current state x_k through the transition matrix A. The input matrix B specifies the effect of input on the state update, however, the system input is u_k=0 (i.e., no input). On the other hand, the system state x_k is transformed into the output measurement y_k through the measurement matrix H.

Kalman Filter updates the system dynamics recursively by estimating the process state. It performs a two-phase prediction and correction cycle to update time and measurement, respectively. The state variables that are integrated into the Kalman filter are the weighted centroid, (x,y) and velocity (v_x, v_y) in the image plane. This will give the following state and measurement vectors:

$$x_k = [x, y, v_x, v_y]^T \tag{5}$$

$$y_k = [x, y]^T \tag{6}$$

Based on the above state and measurement vectors, the transition matrix A and measurement matrix H for the Kalman filter can be constructed as follows, where Δt is the time diff erence between two consecutive frames.

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{8}$$

Kalman filter can track a single moving object. To track multiple objects, a separate Kalman filter is initiated for each moving object in the tracking zone. The data association problem is solved by comparing the predicted location from the previous frame with the detected location in the current frame. The tracks are updated with the closest centroid location. In multiple objects tracking the number of detections may vary due to partial occlusion, size limitation or segmentation failure. Thus, an additional verification step is required to improve the data association. In this work, DGP is used to verify data association and improve tracking quality.

The motion trajectory over consecutive frames will be continuous for sufficient frame rate and the detections within multiple consecutive frames must have spatial relation. In order to match new detections with a previous track's predictions, a likelihood calculation is performed. That means finding a correct relation between observation and prediction using a similarity measure. Then previous detections are assigned to the closest predicted track according to the confidence value that takes into account the covariance of the predicted state of location and the process noise.

2.3 Dynamic Gradient Pattern Verification

To improve tracking efficiency in different challenging situations, it is necessary to combine both low-level location correspondences with high-level feature correspondences. The discriminative information using DGP descriptor utilizes a high-level object correspondence by verifying data association to tackle the shape representation of rigid body even under partial occlusion, pose and orientation variations, and enhance motion tracking.

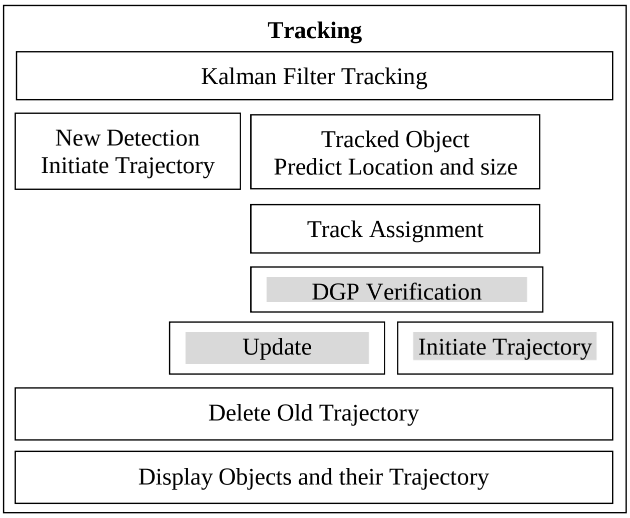
DGP descriptor performs tracking assessment based on the verification criterion. Any detected object with low confidence measurement that limits its assignment can be either a new detection entering the ROI or a tracked object that disappears due to size limitation or partial occlusion. In this case, the image of the tracked object and the predicted template location are resized into 32×32 pixels. Then, the similarity between their DGP descriptors is measured using the normalized Bhattacharyya distance as:

$$BD_N = \frac{\sum_{i=1}^{N} \sqrt{DGP_T(i) \times DGP_p(i)}}{\sum_{i=1}^{N} \max(DGP_T(i), DGP_p(i))} \tag{9}$$

where $DGP_T(i)$ and $DGP_P(i)$ are the i^{th} elements of the DGP descriptor for the tracked and predicted object, respectively, and N is the total length of the DGP descriptor. The value of BD_N will be in the range (0.0-1.0) according to the similarity between objects. Thus, predictions with low BD_N are considered as new detection and initialize a new track, while high BD_N predictions are considered as tracked objects and their tracks are updated. The flow diagram of the tracking procedure is shown in Fig.3.

3. Experimental Results and Discussion

The performance of the tracking system is evaluated by comparing it with a baseline tracker (MATLAB, Motion-Based Multiple Object Tracking). The MATLAB tracker uses Gaussian mixture model for background subtraction and constant velocity Kalman filtering for tracking based on centroid and bounding box size. The i-LIDS and MIT datasets are used for evaluation. i-LIDS Easy, i-LIDS Hard and MIT005.i-LIDS Easy sequence contains slow motion, stopped, and parked vehicles. On the other hand, i-LIDS Hard sequence also contains slow motion, stopped, and parked vehicles with sudden illumination variation. The performance is evaluated on a limited duration from three videos:



1. For each frame, if a new object enters the tracking zone, the object is labeled as detection and its track is initialized with an ID, bounding box, Kalman filter, label, weighted centroid, and track history.
2. Previous detections are assigned to the closest track according to the confidence value of measurement that takes into account the covariance of the predicted state of location and the process noise.
3. Objects within the tracking zone that have low confidence measure are either assigned to a previous track or initiated to a new track according to the DGP descriptor verification and normalized Bhattacharyya distance.
4. Any track that is not updated for a long period is considered as leaving the tracking zone and deleted.
5. Display all detections and tracks

Fig. 3 - The Flow Diagram of the Tracking Procedure

The outputs of the detection module were analyzed manually to count the correct detections when the bounding box covers more than 80% of the vehicle. The quantitative evaluation is done for both detection and tracking modules. The detection module is evaluated using the following object-based measures: Precision, Recall or True Positive Rate (TPR), False Positive Rate (FPR) and False Negative Rate (FNR). They are defined as follow:

$$Precision = \frac{T_p}{T_p + F_p} \times 100\% \tag{10}$$

$$Recall\ (TPR) = \frac{T_p}{T_p + F_n} \times 100\% \tag{11}$$

$$FPR = \frac{F_p}{T_p + F_p} \times 100\% \tag{12}$$

$$FNR = \frac{F_n}{T_p + F_n} \times 100\% \tag{13}$$

$$FNR = (100 - TPR)\% \tag{14}$$

where T_p (True Positives) represents the number of correctly detected vehicles, F_p (False Positives) represents the number of falsely detected vehicles, and F_n (False Negatives) is the number of missing vehicles.

For object tracking assessment, it is more natural to count objects track rather than matching track centroids. Therefore, the performance evaluation methodology proposed in[37] is used in this work. It consists of a rich set of metrics that provide an overview of the tracking performance like Correct Detected Tracks (CDT), False Detected Tracks (FDT), and Track Detection Failure (TDF) recognizing temporal and spatial coherence of tracks through track fragmentation (TF). Data association testing based on object ID change. First, a qualitative evaluation is described using an illustrative example. Next, a more accurate quantitative evaluation is performed.

3.1 Qualitative Performance Evaluation

Illustrative examples that compare the typical tracking results of the proposed system with the baseline MATLAB tracker are shown in Fig.4 to Fig.6. The moving targets and their tracks are outlined with a yellow bounding box for the detection and a blue line for the motion trajectory.

The proposed system automatically detects all moving targets and tracks them through the consecutive frame sequence. Fig.4 shows sample tracking results for about 31 seconds from the i-LIDS Easy sequences. The detection bounding box of the proposed techniques fits more accurately around the tracked vehicle, owing to the improved accuracy of WSDE-CFD bimodal as compared to the GMM used in MATLAB tracker. This will provide more accurate and smooth tracks. The MATLAB tracker generates false alarm tracks, while the proposed tracker does not generate any false alarm tracks.

a b

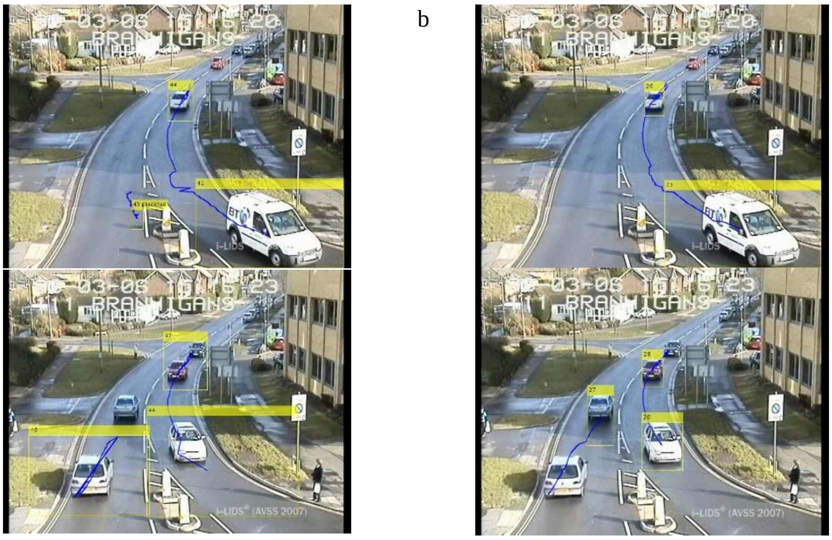


Fig. 4 - Tracking Results for i-LIDS Easy Sequence (a) MATLAB Tracker; (b) Proposed Tracker.

Figure 5 demonstrates the tracking results using i-LIDS Hard sequence. The proposed tracker achieves a smoother object trajectory, due to the use of weighted centroid together with DGP descriptor. The MATLAB tracker generate a false track, while the proposed system does not.

a b

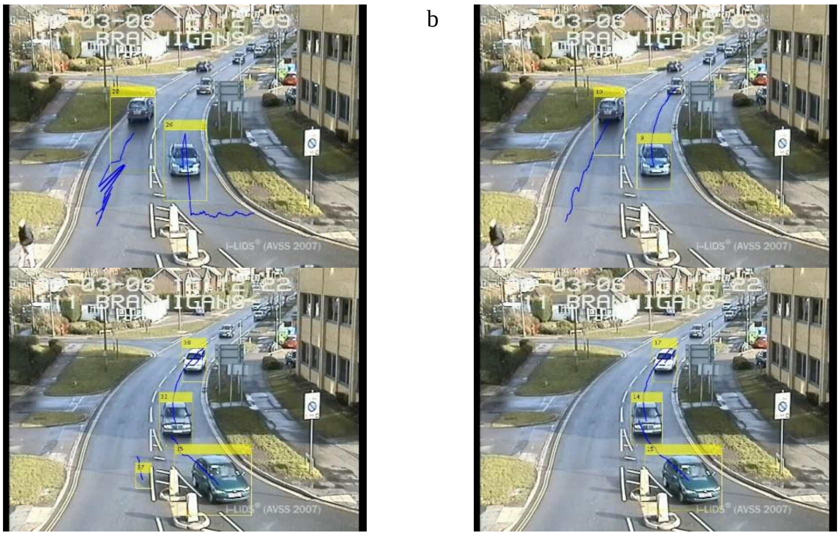


Fig.5 - Tracking Results for i-LIDS Hard Sequence (a) MATLAB Tracker; (b) Proposed Tracker.

The tracking results for MIT dataset is shown in Fig 6. The detection accuracy of the proposed techniques is better than that for the MATLAB tracker, where the detection bounding box fits more closely around the tracked vehicle and the stopped vehicles are still detected. The tracking accuracy of the proposed techniques is better than that for the MATLAB tracker, where the tracking smoothness is better.

3.2 Quantitative Performance Evaluation

The quantitative evaluation results shown in Table 1 and Table 2, indicate that the proposed tracking system outperforms the MATLAB tracker for all evaluation metrics. For evaluation metrics that evaluate the motion segmentation and detection, the results are shown in Table 1. The proposed system has a better performance which can be explained by the improved background estimation technique (WSDE-CFD bimodal) as compared to the standard GMM used in the baseline MATLAB tracker. The TPR of the proposed system is 100%, 98.08% and 98.85% for i-LIDS

Easy, i-LIDS Hard and MIT005 respectively, which is better than the MATLAB tracker that achieves a lower TPR of 97.44%, 92.31 and 96.55 for i-LIDS Easy, i-LIDS Hard and MIT005 respectively. This indicates more robust and accurate results against background clutter and illumination variations.

a

b

Fig.6 - Tracking results for MIT_005 sequences (a) MATLAB tracker; (b) Proposed tracker

The FPR for the MATLAB tracker is much higher than that for the proposed tracking system especially in i-LIDS videos. This is due to the presence of camera vibration, sudden illumination variation, and background clutter. The FPR for MATLAB tracker is 40.16%, 37.66% and 20.00% for i-LIDS Easy, i-LIDS Hard and MIT005 respectively, which is much worse as compared to the proposed system that achieves a FPR of 8.24%, 15.00% and 6.25% for i-LIDS Easy, i-LIDS Hard and MIT005 respectively. Moreover, the miss detection for the proposed system (FNR) is less than 2% for all videos, while the miss detection for the baseline MATLAB tracker is greater than 2.5%.

Table 1- Detection Results for i-LIDS Easy, i-LIDS Hard and MIT005

	i-LIDS Easy		i-LIDS Hard		MIT005	
	MATLAB	Proposed	MATLAB	Proposed	MATLAB	Proposed
Number of Vehicles	78	78	52	52	87	87
Detected Vehicles	76	78	48	51	84	86
False detection	51	7	29	9	21	6
Miss detection	2	0	4	1	3	1
TPR	97.44%	100%	92.31%	98.08%	96.55%	98.85%
FPR	40.16%	8.24%	37.66%	15.00%	20.00%	6.25%
FNR	2.56%	0%	7.69%	1.92%	3.45%	1.15%

The tracking results shown in Table 2 indicate that the proposed system outperforms the baseline MATLAB tracker on all high-level tracking evaluation metrics. The proposed system detected 93.36%, 90.39% and 97.70% of the tracks, which is better as compared to the MATLAB tracker that detected 76.92%, 78.85% and 82.76% of the tracks for i-LIDS Easy, i-LIDS Hard and MIT005 respectively. Moreover, the prior information used in the tracking module correct some confusing detections resulting from orientation variation and partial occlusion.

Table 2 - Tracking Results for i-LIDS Easy, i-LIDS Hard and MIT005

Evaluation Metrics	i-LIDS Easy		i-LIDS Hard		MIT005	
	MATLAB	Proposed	MATLAB	Proposed	MATLAB	Proposed
Number of Ground truth Tracks	78	78	52	52	87	87
Number of detected Tracks	129	116	85	73	123	104
Correct Detected Tracks (CDT)	60	73	41	47	72	85
False Detected Tracks (FDT)	29	7	18	4	10	5
Track Detection Failure (TDF)	17	5	11	5	15	3
Track Fragmentation (TF)	18	16	9	9	13	7
ID Change	28	21	16	12	21	11

The MATLAB tracker has a high FDT of 29, 18, and 10 tracks for i-LIDS Easy, i-LIDS Hard, and MIT005, respectively, owing to the background clutter and illumination variations in these videos, but the proposed system that uses WSDE-CFD bimodal has much less false detected tracks of 7, 4 and 5 tracks. The MATLAB tracker also failed to detect 17, 11, and 15 tracks for i-LIDS Easy, i-LIDS Hard, and MIT005, respectively due to the slow motion and the temporary stops in these videos, while the proposed tracker that adapt slow motion and temporary stops have a lower TDF of 5, 5, and 3, tracks. Track fragmentation occurs when a tracked object is miss detected within the track. This is similar for both MATLAB tracker and the proposed system because it depends on object size and partial occlusion. However, the proposed system generates a lower ID change with better track continuity and linking.

Finally, the proposed system can detect and track vehicles under slow motion, temporary stop, and parking condition. While failure or miss detections are due to size limitation or occlusion before entering the tracking zone.

4. Conclusions

This paper proposed an effective detection and tracking technique for urban road users. The technique incorporates WSDE-CFD bimodal and DGP descriptor. WSDE-CFD bimodal was used to detect foreground object even under slow or temporary stopped conditions. The object was represented by weighted centroid and DGP descriptor to improve shape discrimination, localization, and tracking. Kalman filter was used to estimate and update its likelihood position to validate tracked vehicle with the closest DGP descriptor before assigning it to a specific track. Data association with DGP verification provides an accurate mechanism to supervise the tracking robustness and maintain trajectory smoothness. Experimental results show that integrating motion segmentation, recognition and tracking systems can provide robust detection and tracking for urban road users. The proposed system tracks partially occluded objects and reduce trajectory drifts due to variations in shape, pose, and orientation on urban roads.

References

[1] Abdulrahim, Khairi, & Salam, Rosalina Abdul. (2016). Traffic surveillance: A review of vision based vehicle detection, recognition and tracking. International journal of applied engineering research, 11(1), 713-726

[2] Abdulrahim, Khairi, Seman, Kamaruzzaman, & Salam, Rosalina Abdul. (2019). A new spatio-temporal background–foreground bimodal for motion segmentation and detection in urban traffic scenes. Neural Computing and Applications, 1-17

[3] Liu, Ya, Lu, Yao, Shi, Qingxuan, & Ding, Jianhua. (2013). Optical flow based urban road vehicle tracking. Paper presented at the 2013 ninth international conference on computational intelligence and security

[4] Yaseen, Zaher Mundher, Sulaiman, Sadeq Oleiwi, Deo, Ravinesh C, & Chau, Kwok-Wing. (2019). An enhanced extreme learning machine model for river flow forecasting: State-of-the-art, practical applications in water resource engineering area and future research direction. Journal of Hydrology, 569, 387-408

[5] Zhong, Zuofeng, Zhang, Bob, Lu, Guangming, Zhao, Yong, & Xu, Yong. (2016). An adaptive background modeling method for foreground segmentation. IEEE Transactions on intelligent transportation systems, 18(5), 1109-1121

[6] Elgammal, Ahmed, Harwood, David, & Davis, Larry. (2000). Non-parametric model for background subtraction. Paper presented at the European conference on computer vision

[7] Al-Nawashi, Malek, Al-Hazaimeh, Obaida M, & Saraee, Mohamad. (2017). A novel framework for intelligent surveillance system based on abnormal human activity detection in academic environments. Neural Computing and Applications, 28(1), 565-572

[8] Kim, Kyungnam, Chalidabhongse, Thanarat H, Harwood, David, & Davis, Larry. (2005). Real-time foreground–background segmentation using codebook model. Real-time imaging, 11(3), 172-185

[9] Toyama, Kentaro, Krumm, John, Brumitt, Barry, & Meyers, Brian. (1999). Wallflower: Principles and practice of background maintenance. Paper presented at the Proceedings of the seventh IEEE international conference on computer vision

[10] Oliver, Nuria M, Rosario, Barbara, & Pentland, Alex P. (2000). A Bayesian computer vision system for modeling human interactions. IEEE transactions on pattern analysis and machine intelligence, 22(8), 831-843

[11] Obaida, M Al-Hazaimeh. (2015). Combining audio samples and image frames for enhancing video security. Indian Journal of Science and Technology, 8(10), 940

[12] Manzoor, Muhammad Asif, & Morgan, Yasser. (2017). Vehicle Make and Model classification system using bag of SIFT features. Paper presented at the 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)

[13] Al-Hazaimeh, Obaida M, Al-Nawashi, Malek, & Saraee, Mohamad. (2019). Geometrical-based approach for robust human image detection. Multimedia Tools and Applications, 78(6), 7029-7053

[14] Hsieh, Jun-Wei, Chen, Li-Chih, & Chen, Duan-Yu. (2014). Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition. IEEE Transactions on intelligent transportation systems, 15(1), 6-20

[15] Dalal, Navneet, & Triggs, Bill. (2005). Histograms of oriented gradients for human detection. Paper presented at the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)

[16] Ma'moun Al-Smadi, Khairi Abdulrahim, & Salama, Rosalina Abdul. (2019). Dynamic Features Descriptor for Road User Recognition Using Hierarchal Graph Dynamic Gradient Pattern. International Journal of Recent Technology and Engineering (IJRTE), 7(6S2)

[17] Ottlik, Artur, & Nagel, H-H. (2008). Initialization of model-based vehicle tracking in video sequences of inner-city intersections. International Journal of Computer Vision, 80(2), 211-225

[18] Lee, Kuan-Hui, Hwang, Jenq-Neng, & Chen, Shih-I. (2014). Model-based vehicle localization based on 3-D constrained multiple-kernel tracking. IEEE Transactions on Circuits and Systems for Video Technology, 25(1), 38-50

[19] Zhang, Zhaoxiang, Tan, Tieniu, Huang, Kaiqi, & Wang, Yunhong. (2011). Three-dimensional deformable-model-based localization and recognition of road vehicles. IEEE transactions on image processing, 21(1), 1-13

[20] Lin, Y.-L., Tsai, M.-K., Hsu, W. H., & Chen, C.-W. (2012). Investigating 3-D model and part information for improving content-based vehicle retrieval. IEEE transactions on circuits and systems for video technology, 23(3), 401-413

[21] Bragatto, T. A., Ruas, G. I., Benso, V. A., Lamar, M. V., Aldigueri, D., Teixeira, G. L., & Yamashita, Y. (2008). A new approach to multiple vehicle tracking in intersections using harris corners and adaptive background subtraction. Paper presented at the 2008 IEEE Intelligent Vehicles Symposium

[22] Wei, L., Xudong, X., Jianhua, W., Yi, Z., & Jianming, H. (2014). A SIFT-based mean shift algorithm for moving vehicle tracking. Paper presented at the 2014 IEEE Intelligent Vehicles Symposium Proceedings

[23] Gamage, T. D., Samarawickrama, J. G., & Pasqual, A. (2014). GPU based non-overlapping multi-camera vehicle tracking. Paper presented at the 7th International Conference on Information and Automation for Sustainability

[24] Wu, B.-F., Kao, C.-C., Jen, C.-L., Li, Y.-F., Chen, Y.-H., & Juang, J.-H. (2013). A relative-discriminative-histogram-of-oriented-gradients-based particle filter approach to vehicle occlusion handling and tracking. IEEE Transactions on Industrial Electronics, 61(8), 4228-4237

[25] Elkerdawi, S. M., Sayed, R., & ElHelw, M. (2014). Real-time vehicle detection and tracking using Haar-like features and compressive tracking. Paper presented at the ROBOT2013: First Iberian Robotics Conference

[26] Mandellos, N. A., Keramitsoglou, I., & Kiranoudis, C. T. (2011). A background subtraction algorithm for detecting and tracking vehicles. Expert Systems with Applications, 38(3), 1619-1631

[27] Lou, J., Tan, T., Hu, W., Yang, H., & Maybank, S. J. (2005). 3-D model-based vehicle tracking. IEEE Transactions on image processing, 14(10), 1561-1569

[28] Sivaraman, S., & Trivedi, M. M. (2013). Vehicle detection by independent parts for urban driver assistance. IEEE Transactions on Intelligent Transportation Systems, 14(4), 1597-1608

[29] Zhang, R., Ge, P., Zhou, X., Jiang, T., & Wang, R. (2013). An method for vehicle-flow detection and tracking in real-time based on Gaussian mixture distribution. Advances in Mechanical Engineering, 5, 861321

[30] Tian, B., Li, Y., Li, B., & Wen, D. (2013). Rear-view vehicle detection and tracking by combining multiple parts for complex urban surveillance. IEEE Transactions on Intelligent Transportation Systems, 15(2), 597-606

[31] Wang, N., & Yeung, D.-Y. (2013). Learning a deep compact image representation for visual tracking. Paper presented at the Advances in neural information processing systems

[32] Maggio, E., & Cavallaro, A. (2005). Hybrid particle filter and mean shift tracker with adaptive transition model. Paper presented at the Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005

[33] Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., . . .Huhnke, B. (2008). Junior: The stanford entry in the urban challenge. Journal of field Robotics, 25(9), 569-597

[34] Rezaee, H., Aghagolzadeh, A., & Seyedarabi, H. (2010). Vehicle tracking by fusing multiple cues in structured environments using particle filter. Paper presented at the 2010 IEEE Asia Pacific Conference on Circuits and Systems

[35] Bouvie, C., Scharcanski, J., Barcellos, P., & Escouto, F. L. (2013). Tracking and counting vehicles in traffic video sequences using particle filtering. Paper presented at the 2013 IEEE international instrumentation and measurement technology conference (I2MTC)

[36] Barcellos, P., Bouvié, C., Escouto, F. L., & Scharcanski, J. (2015). A novel video based system for detecting and counting vehicles at user-defined virtual loops. Expert Systems with Applications, 42(4), 1845-1856

[37] Yin, F., Makris, D., & Velastin, S. A. (2007). Performance evaluation of object tracking algorithms. Paper presented at the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Rio De Janeiro, Brazil