MODEL BASED TECHNIQUE FOR VEHICLE TRACKING IN TRAFFIC VIDEO USING SPATIAL LOCAL FEATURES

Arun Kumar H. D.[1] and Prabhakar C. J.[2]

[1]Department of Computer Science, Kuvempu University, Shimoga, India

ABSTRACT

In this paper, we proposed a novel method for visible vehicle tracking in traffic video sequence using model based strategy combined with spatial local features. Our tracking algorithm consists of two components: vehicle detection and vehicle tracking. In the detection step, we subtract the background and obtained candidate foreground objects represented as foreground mask. After obtaining foreground mask of candidate objects, vehicles are detected using Co-HOG descriptor. In the tracking step, vehicle model is constructed based on shape and texture features extracted from vehicle regions using Co-HOG and CS-LBP method. After constructing the vehicle model, for the current frame, vehicle features are extracted from each vehicle region and then vehicle model is updated. Finally, vehicles are tracked based on the similarity measure between current frame vehicles and vehicle models. The proposed algorithm is evaluated based on precision, recall and VTA metrics obtained on GRAM-RTM dataset and i-Lids dataset. The experimental results demonstrate that our method achieves good accuracy.

KEYWORDS

Traffic vehicle tracking, Vehicle model, Spatial local features, CS-LBP, Co-HOG, Shape features, Texture features,

1. INTRODUCTION

Real-time vehicle tracking in traffic video is an essential component of an intelligent video traffic surveillance system. Accurate and real-time vehicle tracking will greatly improve the performance of vehicle classification, road vehicle density estimation, vehicle activity analysis and high-level abnormal events analysis like lane crossing, sudden and long time vehicle stop. The aim of a vehicle tracker is to generate the trajectory of the vehicle over time by locating its position in every frame of traffic video. Development of a robust tracking method for vehicles is challenging because of: complex vehicle appearances like pose and scale variations, occlusion (the vehicle may be occluded by the background or other moving vehicles), and complex vehicle motion.

The features-based vehicle tracking algorithms (Perez, P. et al., 2002; Avidan, S., 2007; Ross, D. et al., 2008; Grabner, H. et al., 2006; Wang, S. et al., 2011) are most promising and tracking is performed based on tracking of features such as distinguishable points or lines on the vehicle. Selecting the right features plays an important role in order to increase the accuracy of features-based tracking algorithms (Beymer, et al., 1997). In general, the desirable property of a visual feature is its uniqueness so that the vehicle can be easily distinguished in the feature space. For example, color, texture, intensity, and pixel-based features are the spatial appearance features widely used to track the vehicle. Su, X. et al. (2007) have proposed rule based multiple objects tracking system for traffic surveillance using a collaborative background extraction algorithm.

DOI:10.5121/mlaij.2016.3401

Jung, Y.K. et al. (2001) have proposed features-based vehicle tracking system, which extracts corner features of the vehicle and tracks the features using a linear Kalman filter.

Babaei, P. et al. (2010) have proposed the tracking system which is based on a combination of a temporal difference and correlation matching in defined traffic zones. The system effectively combines simple domain knowledge about vehicle classes with time domain statistical measures to recognize target vehicles in the presence of partial occlusions. Gao, et al. (2008) have proposed particle filtering based tracking method. A moving vehicle is detected by redundant discrete wavelet transforms method (RDWT), and the key points are obtained by scale-invariant feature transform (SIFT). The matching of key points in the follow-up frames is obtained by the SIFT method and are used as the first particles to improve the tracking performance. Dahlkamp, et al. (2004) proposed Edge-Element Association (EEA) and Marginalized Contour (MCo) approaches for 3D model-based vehicle tracking in traffic scenes.

Based on usage of global and local features, features-based tracking algorithms can be further classified into two categories: Global methods(Ha, et al., 2011), and Local methods (Grabner, H. et al., 2006; Yu, Q. et al., 2008; Tran, S. et al., 2007; He, W. et al., 2009; Wang, S. et al., 2011). The global methods work in many practical applications, but have several basic limitations. First, it is very difficult to capture the small changes in illumination variation and difficult to represent the local details like scale and shape variations. Second, global representations are not robust for partial occlusion. Once the vehicles are occluded, the whole feature vector of vehicle representation is affected. Third, global representations are hard to update. Hence, global methods are not efficient for vehicle tracking in traffic video. Recently, local methods have opened a promising direction to solve these problems by representing a vehicle as a set of local parts or sparse local features. Part-based trackers generally use sets of connected or visual local properties. The parts used for vehicle representation are updated during tracking by removing the old parts that exhibit signs of drifting and adding new ones for easy accommodation of appearance changes.

In order to solve the problems associated with the global features-based algorithms, researchers have developed model-based tracking algorithms for vehicle tracking (Liu, X. et al., 2011; Cehovin, L. et al., 2013; Kwon, J. et al., 2009). In the model-based technique, there are two key components: vehicle representation and dynamics. Vehicle representation tries to model the vehicle as correctly as possible so that the tracking algorithm can correctly describe the complex vehicle appearance. The vehicle dynamics model represents how the vehicle appearance evolves over time to be able to handle appearance variations. These two problems are usually coupled together. The vehicle representation should be designed to simply update the model based on appearance variations, while the vehicle dynamics should be able to take advantage of the characteristics of vehicle representation for model update.

In this paper, we proposed vehicle tracking in traffic video using model-based strategy combined with spatial local features. We construct a vehicle model which captures the variation in vehicle scale, vehicle pose, and complex vehicles occlusion based on spatial local features such as shape and texture features extracted using Co-HOG and CS-LBP operator respectively. After constructing the vehicle model for the current frame, the vehicle features are extracted from each foreground mask of vehicle region and then vehicle model is updated. Finally, the vehicles are tracked based on the similarity measure between current frame vehicles and vehicle model.

2. RELATED WORK

Tracking is used to measure vehicle paths in video sequences. The tracking generally follows two steps: in the first step, features for the vehicle regions are generated in every video frame, and in the second step, a data association step has to provide correspondences between the regions of consecutive frames based on the features and dynamic model. The vehicle tracking is mainly used in two types of traffic videos such as highway and urban traffic scenes videos. Vehicles tracking on highways are easier than in urban traffic as there are few types of objects (one motorized vehicle of various sizes), little change in the orientation of the vehicles and few known entry and exit points. Cameras are also usually located much higher than in urban scenes, which reduce the vehicle occlusions. Tracking vehicles on highways are more challenging when the traffic is slower because the inter-vehicle space is significantly reduced, increasing the occlusion between vehicles. In urban areas, traffic includes pedestrians, motorcycles, and vehicles, and more complicated trajectories, with vehicles turning at intersections, stopping and parking, and many more entry and exit points in the scene. Different computer vision methods have thus been developed for these two applications.

Rad, R. (2005) has proposed real-time tracking of multiple vehicles on the highway. They used Kalman filter and background subtraction techniques. They extract the contour of the vehicle using morphological operations, and the algorithm has three phases, detection of pixels on moving vehicle, detection of a shape of interest in frame sequences and finally determination of relation among objects in frame sequences. Ma, C. et al. (2016) have proposed fusion based hashing method for visual object tracking. Nguyen, P.V. et al. (2008) have proposed Multi-modal Particle Filter (MPF) for tracking vehicles. The aim of this method is to build some most basic functions of a motorcycle surveillance system using MPF based on the color observation model. Babaei, P. et al. (2013) have method which addresses synchronizing the cameras for tracking vehicles simultaneously in overlapping fields of view. Arrospide, J. et al. (2008) have proposed multi object feature tracking strategy. It tracks specially selected points of the image based on computation of sparse optical flow. The tracking strategy includes a central outlier rejection stage, that ensures robustness of the tracker based on probabilistic techniques, and a kalman filtering stage to smooth out the trajectories.

Niknejad, H. et al. (2011) have proposed an embedded real time method for detection and tracking of multi objects including vehicles, pedestrians, motorbikes and bicycles in urban environment. The features of different objects are learned as a deformable object model through the combination of a latent support vector machine (LSVM) and histograms of oriented gradients(HOG). Laser depth data have been used as a priori to generate objects hypothesis regions and HOG feature pyramid level is used to reduce the detection time. Detected objects are tracked through a particle filter which fuses the observations from laser map and sequential images. Messelodi, S. et al. (2005) have proposed the system that uses a combination of segmentation and motion information to localize and track moving vehicles on the urban road plane, utilizing a robust background updating, and a feature-based tracking method. Strigel, E. et al. (2013) have proposed vehicle detection and tracking at intersections by fusing multiple camera views. Using this fusion map, the pose, width and height of the vehicles can be determined. After that, the detected vehicles are tracked by a Gaussian-Mixture approximation of the Probability Hypothesis Density filter. Zheng, Y. et al. (2012) have proposed model based vehicle localization and tracking for urban traffic surveillance using image gradient matching. The matching between the 3D model projection and 2D image data is a key technique for model based localization, recognition and tracking problems. Lee, K. H. et al. (2015) have proposed a model based 3D constrained multiple kernel tracking. This approach regards each patch of the 3D vehicle model as a kernel and tracks the kernels under certain constrains facilitated by the 3D geometry of the vehicle model. A kernel density estimator is designed to well fit the 3D vehicle model during tracking. Kim ,G. et al. (2011) proposed vehicle tracking based on Kalman filter. They detect cars based in Adaboost and the vehicles are tracked using Kalman filter. Barth, A. et al. (2010) have proposed real-time multi-filter approach for vehicle tracking at intersections. Both motion and depth information is combined to estimate the pose and motion parameters of an oncoming vehicle, including the yaw rate, by means of kalman filtering.

Niknejad, H. T. et al. (2012) have proposed multi-vehicle detection and tracking using vehicle mounted monocular camera. The features of vehicles are learned by Latent Support Vector Machine(LSVM) and Histograms of Oriented Gradients (HOG). The detection algorithm combines both global and local features of the vehicle as a deformable object model. Detected vehicles are tracked through a particle filter. Sivaraman, S. et al., (2011) have proposed stereo-monocular fusion approach to on-road localization and tracking of vehicles. Utilizing a calibrated stereo-vision rig, the proposed approach combines monocular detection with stereo-vision for on road vehicle localization and tracking for driver assistance. The system initially acquires synchronized monocular frames and calculates depth maps from the stereo rig. The system then detects vehicles in the image plane using an active learning-based monocular vision approach. Using the image coordinates of detected vehicles, the system then localizes the vehicles in real-world coordinates using the calculated depth map. The vehicles are tracked both in the image plane, and in real-world coordinates.

3. PROPOSED WORK

Our approach for tracking of vehicles in traffic video based on model-based strategy involves two steps. In the first step, background is subtracted and vehicles are detected in frame t. We subtract the background and obtained candidate foreground objects represented as foreground mask using our previous work (Arun Kumar, H.D. et al., 2015). The background subtraction reduces computation time and removes complex background. After obtaining foreground mask of candidate objects in frame t, vehicles are detected using Co-occurrence Histograms of Oriented Gradient (Co-HOG) descriptor. In the second step, we construct a vehicle model for each vehicle in frame t based on shape and texture features extracted from the foreground mask of vehicle regions using Co-HOG and CS-LBP operator. The vehicle model captures the variation in vehicle scale, vehicle pose, and complex vehicles occlusion. After constructing the vehicle model for the current frame, the vehicle features are extracted from each detected vehicle image and then vehicle model is updated. Finally, the vehicles are tracked based on the similarity measure between current frame vehicles and vehicle model. Vehicle position is located by integrating all matching in the vehicle model. Since, features may appear and disappear due to viewpoint changes and occlusions, our dynamic model is designed to be able to add a new feature model and remove expired ones adaptively and dynamically. The Hungarian algorithm is used to link detections for tracking. The flow diagram of proposed vehicle tracking approach is shown in Figure 1.

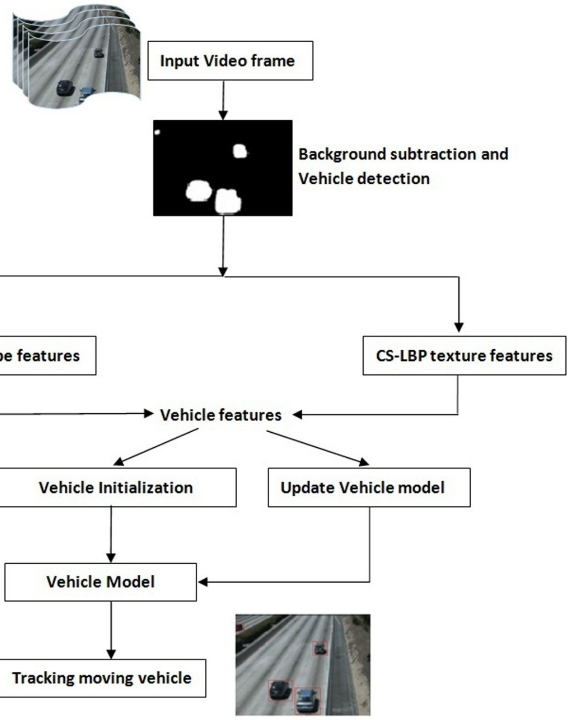


Figure 1. The flow diagram of our approach

3.1 Formulation of proposed approach

Traffic vehicle tracking can generally be formulated as a multi-variable estimation problem. Given the video sequence $\{V_1, V_2, ..., V_m\}$ as input, we use S_t^i to indicate the state of the i^{th} vehicle in the t^{th} frame. We use $S_t = (S_{1,t}, S_{2,t}, ..., S_{M_t t})$ to indicate the states of all the M_t vehicles in the t^{th} frame, $O = \{O_t\}, i = 1, 2, ..., t$ as all the t detections, $S_{i,1:t} = \{S_{i,1}, S_{i,2}, ..., S_{i,t}\}$ to indicate the sequential states of the i^{th} vehicle from the first frame to the t^{th} frame, and $S_{1:t} = \{S_1, S_2, ..., S_t\}$ to indicate all the sequential states of all the vehicles from the first frame to the t^{th} frame.

$$S_{1:t} = \arg\max P(S_{1:t} | O_{1:t}), \tag{1}$$

The state of the vehicle in the current frame t only depends on the state of the vehicle in previous frame $t-1$. When we process frame t, only the tracking in frame $t-1$ and the image in the t^{th} frame I_t are involved in the calculation.

$$P(S_t | O_{1:t}) = P(S_t | O_t, S_{t-1}) = \int P(S_t | O_t, I_t, S_{t-1}) P(O_t | I_t, S_{t-1}) dO_t, \tag{2}$$

where $P(O_t | I_t, S_{t-1})$ indicates how realistic the detection in frame t is, $P(S_t | O_t, I_t, S_{t-1})$ describes how well the detections in frame t matches tracking's in frame $t-1$.

3.2 Background Subtraction and Vehicles Detection

In this section, we introduce the method for background subtraction and vehicles detection. The formulation for background subtraction (foreground detection) and vehicles detection process is $P(O_t | I_t, S_{t-1})$ defined in equation (2), which is described as:

$$P(O_t | I_t, S_{t-1}) = \begin{cases} P_{fg} P_{\det} & \text{if the category of object is moving vehicle}, \\ P_{bg} & \text{otherwise.} \end{cases} \tag{3}$$

We subtract the background and obtained the foreground mask of candidate moving objects using our approach proposed in the previous work (Arun Kumar, H.D. et al. 2015), which reduces the computation time and removes the complex background. Our previous approach has verified to be an efficient and effective background subtraction method. Our approach for background subtraction uses modified SXCS-LBP texture descriptor for finding foreground mask of candidate objects,

$$P_{fg}(O_t | I_t, S_{t-1}) = \begin{cases} 1 & \text{if } O_t \text{ is in foreground regions by background subtraction}, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

After background subtraction, each pixel is labelled as foreground or background using 0 or 1, 0 indicates background and 1 indicates foreground. Once the background subtraction process is completed, we obtain foreground mask for each candidate object.

Once foreground mask of each object is obtained, it is necessary to detect vehicles among candidate objects in the frame t. The detection of vehicles is described as $P_{\det}$. In order to detect vehicles, we utilize the approach proposed by Tomoki Watanabe, et al. (2009), in which Co-occurrence Histograms of Oriented Gradient (Co-HOG) features are extracted to represent moving vehicles.

$$P_{\det}(O_t | I_t, S_{t-1}) = \begin{cases} 1 & \text{if } O_t \text{ is obtained by the detection}, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

3.3 Vehicle Tracking

After detecting the vehicles in frame t, we track the moving vehicles in traffic video. There are two subsections. First, we initialize the tracking, in the second step, we calculate the similarity between tracking and observation, and then observation could be linked to being a tracking.

3.3.1. Initialization of tracking

In order to initialize tracking, vehicle model is constructed for each detected vehicle at time t. The appearance of the vehicle under tracking may change over time due to changes of vehicle scale, vehicle pose, complex vehicle occlusion, and the appearance variation would lead to losing track. Hence, vehicle model captures these changes occurred in vehicle while it is moving. In the starting stage of the vehicle tracking, the target vehicle model in the first frame is only initialized and all the remaining vehicle models are empty. Generally, the vehicle model is updated incrementally. Whenever a larger appearance variation is detected, the updated target vehicle models are stored in the empty models.

For M vehicles at time t, we construct M vehicle models represented by a shape and texture features obtained using Co-HOG and CS-LBP method respectively. In the following subsections, we present procedure involved in extracting the Co-HOG and CS-LBP features from each detected vehicle image.

Co-occurrence Histogram of Oriented Gradients (Co-HOG)

Co-occurrence Histogram of Oriented Gradients (Co-HOG) (Tomoki Watanabe, et al., 2009) descriptor is an extension of the original HOG shape descriptor that captures the spatial information of neighboring pixels. Instead of counting the occurrence of the gradient orientation of a single pixel, gradient orientations of two or more neighboring pixels are considered. For each pixel in an image block, the gradient orientations of the pixel pair formed by its neighbor and itself are examined. The Co-HOG has two important merits. One is the robustness against illumination variation because gradient orientations are computed from the local intensity difference. The other merit is the robustness against deformations because slight shifts deformations make small histogram value changes.

The co-occurrence matrix expresses the distribution of gradient orientations at a given offset over an image. The combinations of neighbor gradient orientations can express shapes in detail. Mathematically, a co-occurrence matrix K is defined at an each block $N \mathrm{X} M$ of an image *I*, parameterized by an offset (x, y) as:

$$K_{x,y}(i,j) = \sum_{p=1}^{N} \sum_{q=1}^{M} \begin{cases} 1 & \mathit{if}\ I(p,q) = i\ \mathit{and}\ I(p+x, q+y) = j, \\ 0 & \mathit{otherwise.} \end{cases} \tag{6}$$

We describe the process of Co-HOG calculation as follows. Initially, we compute gradient orientations from an image by

$$\theta = \arctan\left(\frac{I_y}{I_x}\right), \tag{7}$$

where $\arctan()$ returns the inverse tangent of the elements in degrees. I_y and I_x are vertical and horizontal gradient respectively calculated by Gaussian filter. We label each pixel with one of eight discrete orientations. In our approach, all 0^0 to 360^0 orientations are split up into eight orientations per 45^0. Then, we compute co-occurrence matrices using Eq. (6). We used 31 offsets, including a zero offset. In most of other applications, the authors proceed by dividing an image into a number of blocks and from each block extract co-occurrence matrices. We divide the vehicle image patch into non overlapping blocks of size $N \mathrm{X} M$, the co-occurrence matrices are computed for each block. Finally, the components of all the co-occurrence matrices are concatenated into a vector.

We divide the vehicle image into $2 \mathrm{X} 4$ (the accuracy of our approach is considerably better than for other number of blocks. Hence, in all the experiments, we divide the vehicle image into $2 \mathrm{X} 4$ blocks) blocks and Co-HOGs at each block are computed. Figure 2 gives an illustration of Co-HOG feature descriptor extraction process from a given vehicle image.

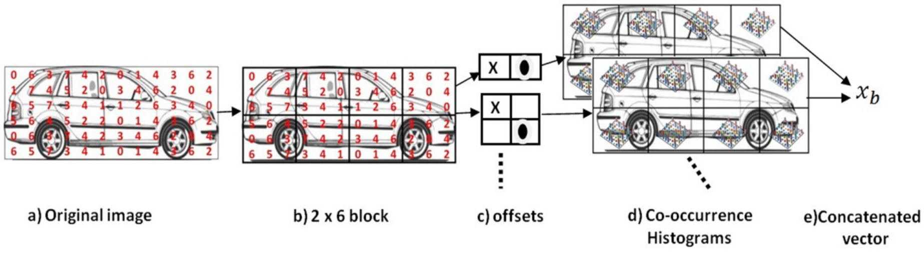


Figure 2. Illustration of Co-HOG feature descriptor extraction process

The dimension of Co-HOG feature vector for the given vehicle image is 15,424 when we divide the vehicle image into $2X4$ blocks. From one small region or block, Co-HOG obtains 31 co-occurrence matrices. A co-occurrence matrix has 64 components. Thus, Co-HOG obtains (64 × 30 + 8) × (2 × 4) = 15,424 components for each vehicle image.

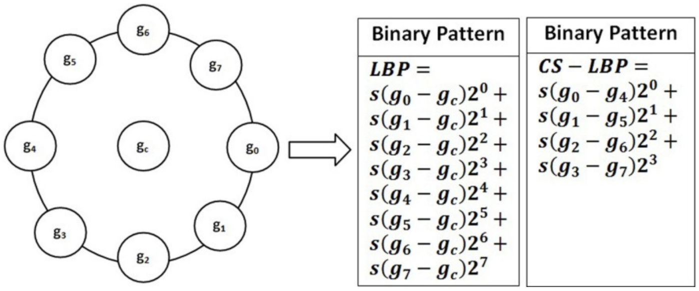
Center Symmetric Local Binary Pattern (CS-LBP)

Heikkila, et al. (2002) have proposed a novel interest region descriptor called as Center Symmetric Local Binary Pattern (CS-LBP) descriptor which is an extension of LBP texture operator. The CS-LBP descriptor has several advantages such as tolerance to illumination changes, robustness on flat image areas, and computational efficiency. The CS-LBP compares the gray values of pairs of pixels in the center-symmetric direction. For 8 neighbors, LBP produces 256 different binary patterns, whereas for CS-LBP produces only 16 binary patterns, Figure 3 gives an illustration of CS-LBP feature descriptor computation process, and CS-LBP is mathematically defined as follows:

$$CS-LBP_{R,P} = \sum_{p=0}^{\left(\frac{p}{2}\right)-1} S\left(g_p - g_{p+\left(\frac{p}{2}\right)}\right)2^p, \tag{8}$$

where g_p and $g_{p+\left(\frac{p}{2}\right)}$ correspond to gray values of the center-symmetry pair of pixels and the function $S(x)$ is threshold function defined as follows:

$$S(x) = \begin{cases} 1 & x \geq 0 \\ 0 & otherwise. \end{cases} \tag{9}$$



Binary Pattern	Binary Pattern
$LBP =$ $s(g_0 - g_c)2^0 +$ $s(g_1 - g_c)2^1 +$ $s(g_2 - g_c)2^2 +$ $s(g_3 - g_c)2^3 +$ $s(g_4 - g_c)2^4 +$ $s(g_5 - g_c)2^5 +$ $s(g_6 - g_c)2^6 +$ $s(g_7 - g_c)2^7$	$CS - LBP =$ $s(g_0 - g_4)2^0 +$ $s(g_1 - g_5)2^1 +$ $s(g_2 - g_6)2^2 +$ $s(g_3 - g_7)2^3$

Figure 3. LBP and CS-LBP features for neighborhood of 8 pixels

The concatenated Co-HOG and CS-LBP feature vector of each vehicle represents vehicle model for each vehicle and concatenated feature vector *(fv)* of each vehicle *(O_i)* is defined as

$$fv(O_i) = \left\{hv_j\right\}_{j=1}^{N}, \tag{10}$$

where N is the total number of feature vector bins and hv_j is the value of j^{th} bin.

For each trajectory, the vehicle model is updated frame by frame. The appearance in the current frame is the most important one, so we update as follows:

$$fv_{k+1} = (1-\alpha)\, fv_k + \alpha fv_c, \tag{11}$$

where fv_k is the feature vector after updating in k^{th} frame, α is a constant which is set to 0.8 and fv_c is the feature vector calculated in the current frame. Each vehicle is tracked based on minimum distance between feature vector of the vehicle in the current frame and its associated vehicle model. The Hellinger distance is used to compare the feature vectors

$$d\left(fv_1, fv_2\right) = \sqrt{1 - \frac{1}{\sqrt{\overline{fv_1}, \overline{fv_2} N^2}} \sum_{q=1}^{N} \sqrt{hv_{q,1} hv_{q,2}}}, \tag{12}$$

where fv_1 and fv_2 are two feature vectors, and

$$\overline{fv_k} = \frac{1}{N}\sum_{q=1}^{N} hv_{q,k,} \qquad k = 1, 2 \tag{13}$$

3.3.2. Similarity between tracking and observation

We simplify $P(S_t | O_t, I_t, S_{t-1})$ is probability for a tracking and a detection,

$$P(S_t | O_t, I_t, S_{t-1}) = P_a, \tag{14}$$

$$P_a = \frac{1}{\sqrt{2\pi}\sigma_a} \exp\left(-\frac{d\left(fv\left(D_t\right), fv_t\right)^2}{2\sigma_a^2}\right), \tag{15}$$

where $fv\left(D_t\right)$ is the feature vector of detected vehicle in frame t, fv_t is the feature vector of tracking after updating in frame t, $d\left(.\right)$ is the function to calculate Hellinger distance, and σ_a is a given threshold.

4. EXPERIMENTAL RESULTS

The performance evaluation of the proposed method for vehicle-tracking is a frame-by-frame evaluation process. We carry out experiments on challenging two traffic surveillance video datasets such as GRAM-Road Traffic Monitoring (GRAM-RTM) (Guerrero-Gómez-Olmedo, et al., 2013) and i-Lids. In order to measure the performance of our approach for vehicle-tracking, we used evaluation metrics such as precision, recall, and Vehicle-Tracking Accuracy (VTA)

(Smith, K. et al., 2005). The precision measures how much of the estimates (ε is tracker outputs are referred to as estimates) cover the ground truth (GT) vehicle and can take values between 0 (no overlap) and 1 (full overlap). It is possible to have high precision with poor quality tracking as depicted in Figure 4(a). The recall measures how much of the GT is covered by then ε and can take values between 0 (no overlap) and 1 (full overlap). It is possible to have a high recall yet have poor quality tracking (Figure 4(b)). Vehicle-Tracking Accuracy (VTA) is total position error for matching vehicle hypothesis pair over all frames, averaged by the total number of matches. The precision ($v_{i,j}$), recall ($\rho_{i,j}$), and VTA are defined as follows:

$$v_{i,j} = \frac{\varepsilon_i \cap GT_j}{\varepsilon_i}, \tag{16}$$

$$\rho_{i,j} = \frac{\varepsilon_i \cap GT_j}{GT_j}, \tag{17}$$

$$VTA = 1 - \frac{\sum_t \left(M_t + FP_t + MM_t \right)}{\sum_t \left(GT_t \right)} = 1 - \left(\overline{M} + \overline{FP} + \overline{MM} \right), \tag{18}$$

where GT_j is the ground truth for tracking target vehicles and indexed by j , ε_i is the tracker output are referred to as estimates and indexed by i , M_t is the number of missed vehicles at time t . FP_t is the number of false positives that correspond to detect vehicles and that do not overlap any real vehicles in the scene. MM_t is the number of mismatches at time t . GT_j is the total number of vehicles at time t . $\overline{M}$, $\overline{FP}$ and $\overline{MM}$ represent the corresponding ratio.

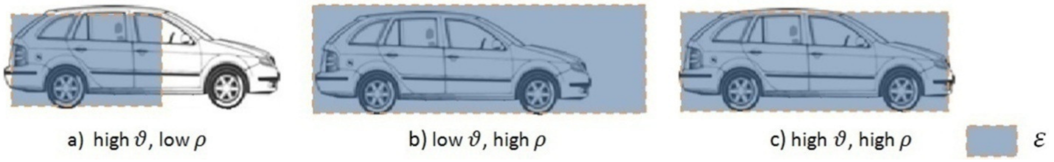


Figure 4. a) precision (v) b) recall (ρ) c) both precision and recall should have high values

There are four basic types of errors that our system can make. The first type of error may occur when a vehicle exists, but the system does not recognize it (False Negative: A ground truth object exists that is not associated with an estimate). The second type of error occurs when the system may indicate the presence of a vehicle which does not exist (False Positive: An estimate exists that is not associated with a ground truth object). The third type of error occurs when one vehicle is tracked by multiple estimates (Multiple Trackers: Two are more estimates are associated with the same ground truth). The last type of error occurs when multiple vehicles are tracked by one estimate (Multiple Objects: Two or more ground truth objects are associated with the same estimate). These errors are depicted in Figure 5.

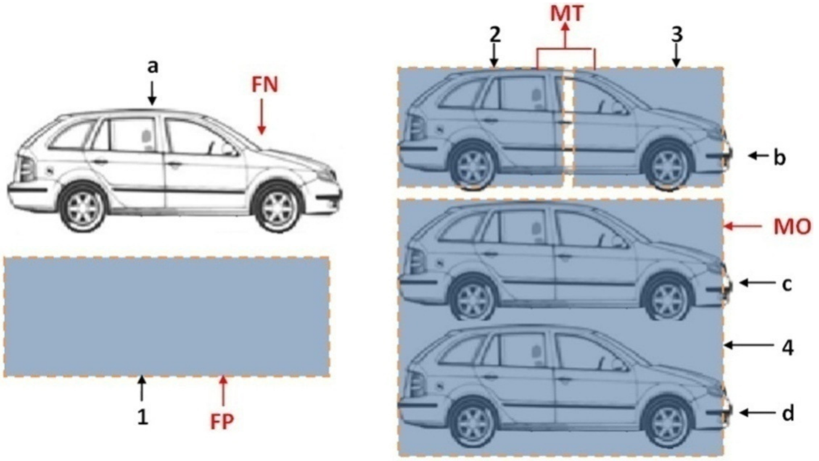


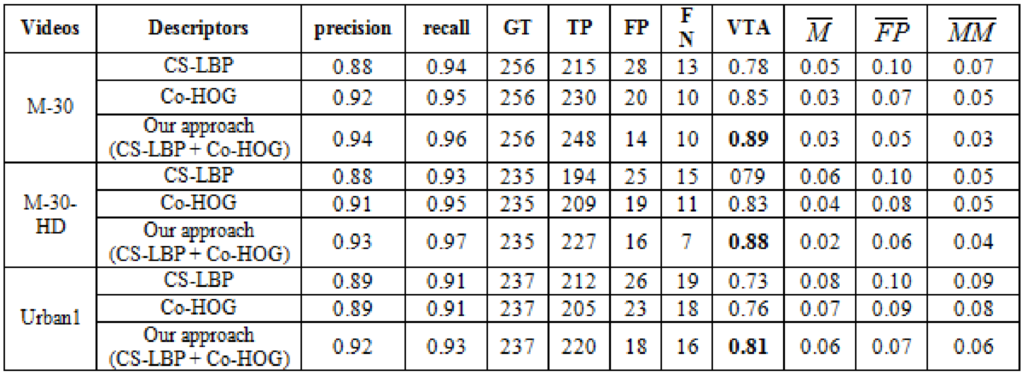
Figure 5. Types of configuration errors εs $(1,2,3,4)$ attempt to track four GTs (a,b,c,d)

4.1. Experiments on GRAM-RTM dataset

In the first set of experiments, we evaluated the performance of our approach quantitatively using GRAM Road-Traffic Monitoring (GRAM-RTM) dataset. The GRAM Road-Traffic Monitoring dataset consists of three traffic video sequences and these video sequences were recorded under different conditions and with different platforms. The first video sequence is M-30 video (7529 frames), was recorded on a sunny day and resolution for each frame is $800 X 480$. The second video sequence is M-30-HD (9390 frames), was recorded in the same location, but cloudy day and resolution for each frame is $1200 X 720$. The third video sequence Urban1 (2345 frames), was recorded at a busy urban intersection and the resolution of each frame is $600 X 360$. The ground truths of these three video sequences were manually obtained. We compared the performance of our approach with CS-LBP alone (Texture descriptor) and Co-HOG alone (Shape Descriptor).

Table 1 shows the comparative study of our approach (CS-LBP + Co-HOG) with CS-LBP alone and CO-HOG alone. All of the values represent the average of the considered criterion obtained for the whole three M-30, M-30-HD, and Urban1 video sequences. It is observed that our approach achieves the highest precision and highest recall compared to CS-LBP alone and CO-HOG alone for M-30, M-30-HD, and Urban1 video sequences. The VTA of our approach is 89%, 88% and 81% for the three challenging video sequences, which means that the tracking of almost all of the vehicles is detected. The low FP level of our approach (03%, 02%, and 06%) shows that almost all of the detected tracking's correspond to a real vehicle in the scene. 03% of the vehicles are totally missing from the M-30 video sequence, while 04% are missing from the M-30-HD video sequence, and 06% are missing from the Urban1 video sequence. The increase in FP for Urban1 video sequence is mainly due to some pedestrians located at the scene of the ROI. Our approach decreases the mismatch rate (03%, 02% and 06% for respective video sequences) compared to shape and a texture descriptor alone. The combination of CS-LBP and Co-HOG improves the VTA, precision, recall and reduce false detection, mismatch and missed vehicles. Figure 6 shows the qualitative performance of our approach for all three GRAM Road-Traffic Monitoring dataset video sequences. The red color windows in the sample video frames describe tracked vehicles.

Table 1. The results comparison of our approach with CS-LBP alone and Co-HOG alone based on precision, recall and VTA for M-30, M-30-HD, and Urban1 video sequences.



Videos	Descriptors	precision	recall	GT	TP	FP	F N	VTA	$\overline{M}$	$\overline{FP}$	$\overline{MM}$
M-30	CS-LBP	0.88	0.94	256	215	28	13	0.78	0.05	0.10	0.07
	Co-HOG	0.92	0.95	256	230	20	10	0.85	0.03	0.07	0.05
	Our approach (CS-LBP + Co-HOG)	0.94	0.96	256	248	14	10	**0.89**	0.03	0.05	0.03
M-30-HD	CS-LBP	0.88	0.93	235	194	25	15	079	0.06	0.10	0.05
	Co-HOG	0.91	0.95	235	209	19	11	0.83	0.04	0.08	0.05
	Our approach (CS-LBP + Co-HOG)	0.93	0.97	235	227	16	7	**0.88**	0.02	0.06	0.04
Urban1	CS-LBP	0.89	0.91	237	212	26	19	0.73	0.08	0.10	0.09
	Co-HOG	0.89	0.91	237	205	23	18	0.76	0.07	0.09	0.08
	Our approach (CS-LBP + Co-HOG)	0.92	0.93	237	220	18	16	**0.81**	0.06	0.07	0.06

4.2. Experiments on i-Lids dataset

In the second set of experiments, we evaluated the performance of our approach quantitatively using i-Lids dataset. The i-Lids dataset consists of seven traffic video sequences, among seven video sequences, we selected AVSS PV Easy video sequence which includes scenes of vehicle turning, illumination changes, and vehicles moving from far to near. The resolution for each frame is $720 X 576$. The ground truths of this video sequence were manually obtained. Table 2 shows the precision, recall, and VTA based quantitative comparison of our approach result with CS-LBP alone and Co-HOG alone for the AVSS PV easy video sequence. The proposed approach (CS-LBP + Co-HOG) has achieved highest precision, recall and VTA compared to CS-LBP alone and Co-HOG alone. The FP rate of our approach is very high because AVSS PV easy video sequence contains some pedestrians, motorcycles, and vehicles moving from far to near. Figure 7 shows the tracking results of our approach for sample frames of AVSS PV easy i-Lids dataset. The first row of Figure 7 shows sample original video frames where vehicles pose are changing because of curved lane. The second row shows tracking result represented using red color window. It is observed that our approach accurately track the vehicles even though vehicle pose changes. This is because, in each frame, the vehicle model is updated efficiently in order to capture the variation in pose of the vehicles.

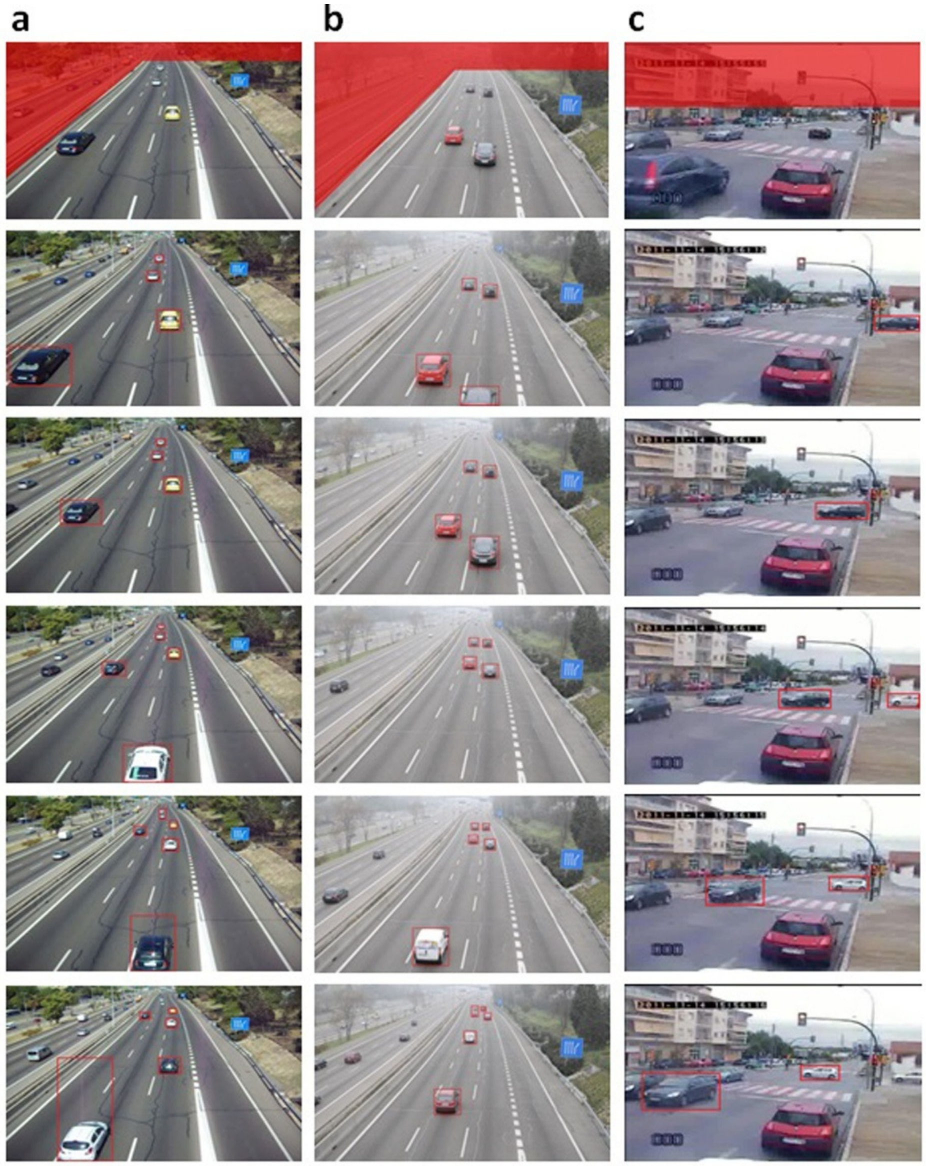


Figure 6. Vehicle tracking result of our approach for all three video sequences of GRAM Road-Traffic Monitoring dataset a) M-30 b) M-30-HD, and c) Urban1(first row is exclusion area shown using red color).

Table 2. The results comparison of our approach with CS-LBP alone and Co-HOG alone based on precision, recall and VTA for AVSS PV Easy video sequence.

Videos	Descriptors	precision	recall	GT	TP	FP	FN	VTA	$\overline{M}$	$\overline{FP}$	$\overline{MM}$
AVSS PV Easy	CS-LBP	0.88	0.90	80	74	10	8	0.60	0.10	0.13	0.17
	Co-HOG	0.89	0.87	80	73	9	10	0.59	0.13	0.12	0.16
	Our approach (CS-LBP + Co-HOG)	0.90	0.91	80	77	8	7	**0.69**	0.09	0.10	0.12

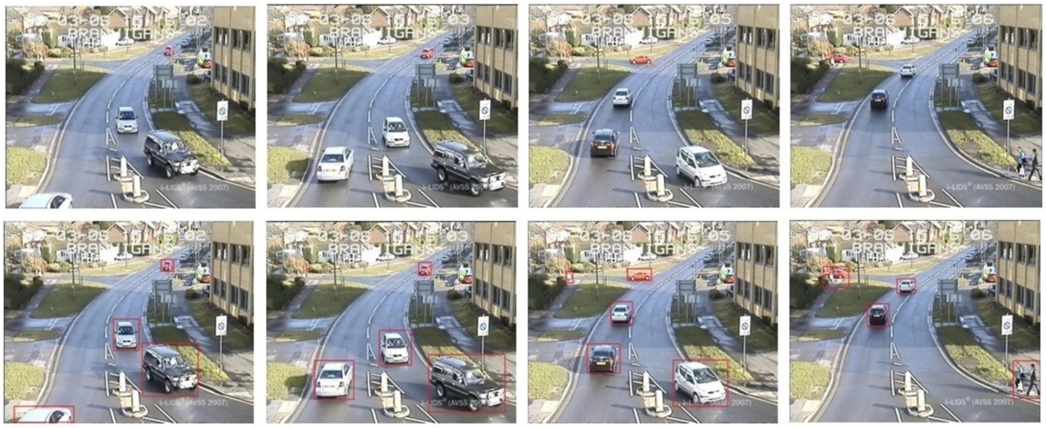


Figure 7. Vehicle tracking result of our approach for AVSS PV easy video sequence

4.3. Visual comparison with existing method

We compared tracking results of our approach obtained on i-Lids dataset visually with the tracking results of SIFT-based Mean Shift algorithm proposed by Liang et al. (2014). The first row of Figure 8 shows sample original video frames, such as Frame #26, Frame #55, Frame #75, and Frame #85. The second and third row shows the results of Mean Shift method and our proposed approach results for the sample frames. The tracking window of Mean shift method deviates at the Frame #55, Frame #75 and Frame #85. This is because the color histogram of the candidate template is changed when the moving vehicle is turning left, and the illumination is affected by the shadow of the building. For Mean Shift algorithm, when the illumination and shape of the vehicle changes, the number of matched points greatly increases and the performance of the method decreases, which records false tracking rate. Our approach results presented in the third row demonstrate that the moving vehicles are tracked more accurately for the Frame #26, Frame #55, Frame #75, and Frame #85. The increase in tracking rate of our approach is due to the fact that adaptation of CS-LBP descriptor, which is illumination invariant and extracts accurate vehicles texture features, and the Co-HOG descriptor gives accurate shape features even though the vehicle changes its pose. Hence, the combination of shape and texture descriptors increases the vehicle tracking result.

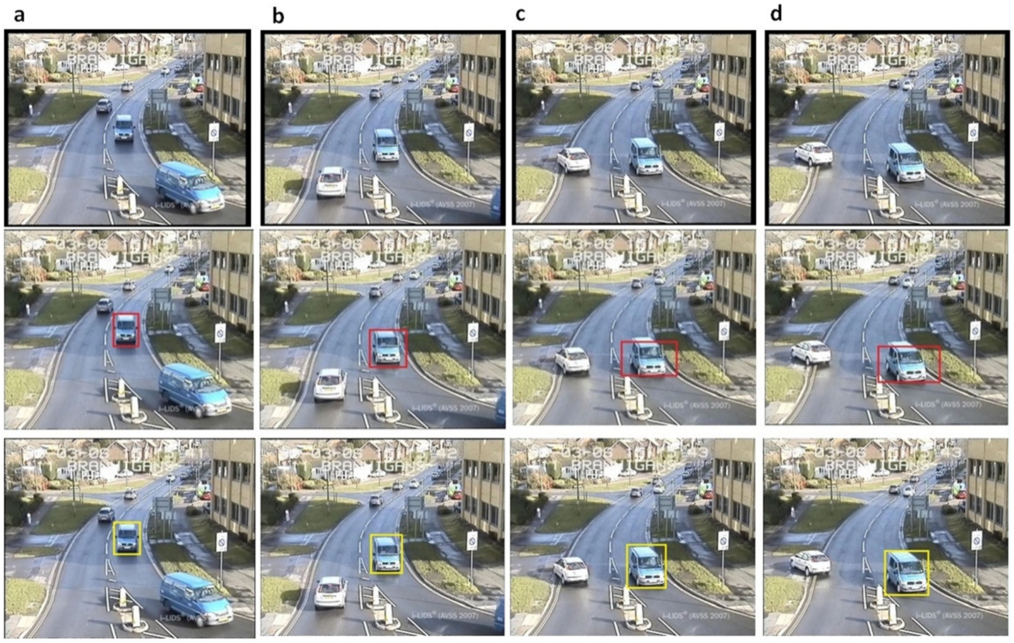


Figure 8. Vehicle tracking result for AVSS PV Easy video sequences for Mean Shift (red), and proposed method (yellow) a) Frame #26, b) Frame #55, c) Frame #75, and d) Frame #85.

5. CONCLUSION

In this paper, we proposed a model-based vehicle tracking technique using spatial local features such as shape and texture features. The shape descriptor such as Co-HOG is used for the representation of vehicle shape and CS-LBP texture descriptor is used for representation of vehicle texture. The vehicle model is constructed which captures the variation in illumination, vehicle scale, vehicle pose and complex vehicles occlusion. The evaluation process conducted on two popular datasets such as GRAM-RTM and i-Lids demonstrate that our approach achieves highest vehicle tracking accuracy. The visual comparison with existing method shows that our approach yields accurate tracking even vehicle pose and illumination changes. The drawback of our approach is that when the pedestrians and motorcycles are present in the video sequence, our approach tracks these objects as vehicles and it reduces the tracking accuracy.

REFERENCES

[1] Pérez, P., Hue, C., Vermaak, J., & Gangnet, M. (2002), “Color-based probabilistic tracking”, In Computer vision—ECCV 2002 (pp. 661-675). Springer Berlin Heidelberg.

[2] Avidan, S. (2007), “Ensemble tracking”, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 29(2), 261-271.

[3] Ross, D. A., Lim, J., Lin, R. S., & Yang, M. H. (2008), “Incremental learning for robust visual tracking”, International Journal of Computer Vision, 77(1-3), 125-141.

[4] Grabner, H., & Bischof, H. (2006, June), “On-line boosting and vision”, In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on (Vol. 1, pp. 260-267). IEEE.

[5] Grabner, H., Grabner, M., & Bischof, H. (2006, September), “Real-Time Tracking via On-line Boosting”, In BMVC (Vol. 1, No. 5, p. 6).

[6] Wang, S., Lu, H., Yang, F., & Yang, M. H. (2011, November), “Superpixel tracking”, In Computer Vision (ICCV), 2011 IEEE International Conference on(pp. 1323-1330). IEEE.

[7] Beymer, D., McLauchlan, P., Coifman, B., & Malik, J. (1997, June), " A real-time computer vision system for measuring traffic parameters", In Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on (pp. 495-501).

[8] Su, X., Khoshgoftaar, T. M., Zhu, X., & Folleco, A. (2007), "Rule-based multiple object tracking for traffic surveillance using collaborative background extraction", In Advances in Visual Computing (pp. 469-478). Springer Berlin Heidelberg.

[9] Jung, Y. K., & Ho, Y. S. (2001), "A feature-based vehicle tracking system in congested traffic video sequences", In Advances in Multimedia Information Processing—PCM 2001 (pp. 190-197). Springer Berlin Heidelberg.

[10] Babaei, P. (2010, December), "Vehicles tracking and classification using traffic zones in a hybrid scheme for intersection traffic management by smart cameras", In Signal and Image Processing (ICSIP), 2010 International Conference on (pp. 49-53). IEEE.

[11] Gao, T., Liu, Z. G., Gao, W. C., & Zhang, J. (2008), "Moving vehicle tracking based on SIFT active particle choosing", In Advances in Neuro-Information Processing (pp. 695-702). Springer Berlin Heidelberg.

[12] Dahlkamp, H., Pece, A. E., Ottlik, A., & Nagel, H. H. (2004), "Differential analysis of two model-based vehicle tracking 0020 approaches", In Pattern Recognition (pp. 71-78). Springer Berlin Heidelberg.

[13] Ha, S. W., & Moon, Y. H. (2011), "Multiple object tracking using SIFT features and location matching", International Journal of Smart Home, 5(4), 17-26.

[14] Yu, Q., Dinh, T. B., & Medioni, G. (2008), "Online tracking and reacquisition using co-trained generative and discriminative trackers", In Computer Vision–ECCV 2008 (pp. 678-691). Springer Berlin Heidelberg.

[15] Tran, S., & Davis, L. (2007, October), "Robust Object Tracking with Regional Affine Invariant Features", In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on (pp. 1-8). IEEE.

[16] He, W., Yamashita, T., Lu, H., & Lao, S. (2009, September), "Surf tracking", In Computer Vision, 2009 IEEE 12th International Conference on (pp. 1586-1592). IEEE.

[17] Liu, X., Lin, L., Yan, S., Jin, H., & Jiang, W. (2011), "Adaptive object tracking by learning hybrid template online", Circuits and Systems for Video Technology, IEEE Transactions on, 21(11), 1588-1599.

[18] Cehovin, L., Kristan, M., & Leonardis, A. (2013), "Robust visual tracking using an adaptive coupled-layer visual model", Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(4), 941-953.

[19] Kwon, J., & Lee, K. M. (2009, June), "Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping Monte Carlo sampling", In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 1208-1215). IEEE.

[20] Rad, R., & Jamzad, M. (2005), "Real time classification and tracking of multiple vehicles in highways", Pattern Recognition Letters, 26(10), 1597-1607.

[21] Ma, C., Liu, C., Peng, F., & Liu, J. (2016), "Multi-feature Hashing Tracking", Pattern Recognition Letters, 69, 62-71.

[22] Nguyen, P. V., & Le, H. B. (2008), "A multi-modal particle filter based motorcycle tracking system", In PRICAI 2008: Trends in Artificial Intelligence (pp. 819-828). Springer Berlin Heidelberg.

[23] Babaei, P., Fathy, M., & Berangi, R. (2013), "Consistent Vehicles Tracking By Using A Cooperative Distributed Video Surveillance System", International Research Journal of Applied and Basic Sciences, 4(10), 3658-3663.

[24] Arróspide, J., Salgado, L., Nieto, M., & Jaureguizar, F. (2008, October), "On-board robust vehicle detection and tracking using adaptive quality evaluation", In 2008 15th IEEE International Conference on Image Processing. IEEE.

[25] Niknejad, H. T., Takahashi, K., Mita, S., & McAllester, D. (2011, June), "Embedded multi-sensors objects detection and tracking for urban autonomous driving", In Intelligent Vehicles Symposium (IV), 2011 IEEE (pp. 1128-1135). IEEE.

[26] Messelodi, S., Modena, C. M., & Zanin, M. (2005), "A computer vision system for the detection and classification of vehicles at urban road intersections", Pattern analysis and applications, 8(1-2), 17-31.

[27] Strigel, E., Meissner, D., & Dietmayer, K. (2013, June), “Vehicle detection and tracking at intersections by fusing multiple camera views”, In Intelligent Vehicles Symposium (IV), 2013 IEEE (pp. 882-887). IEEE

[28] Zheng, Y., & Peng, S. (2012, September), “Model based vehicle localization for urban traffic surveillance using image gradient based matching”, In Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on (pp. 945-950). IEEE.

[29] Lee, K. H., Hwang, J. N., & Chen, S. I. (2015), “Model-Based Vehicle Localization Based on 3-D Constrained Multiple-Kernel Tracking”, Circuits and Systems for Video Technology, IEEE Transactions on, 25(1), 38-50.

[30] Kim, G., Kim, H., Park, J., & Yu, Y. (2011), “Vehicle tracking based on kalman filter in tunnel”, In Information Security and Assurance (pp. 250-256). Springer Berlin Heidelberg.

[31] Barth, A., & Franke, U. (2010, September), “Tracking oncoming and turning vehicles at intersections”, In Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on (pp. 861-868). IEEE.

[32] Niknejad, H. T., Takeuchi, A., Mita, S., & McAllester, D. (2012), “On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation”, Intelligent Transportation Systems, IEEE Transactions on, 13(2), 748-758.

[33] Sivaraman, S., & Trivedi, M. M. (2011, October), “Combining monocular and stereo-vision for real-time vehicle ranging and tracking on multilane highways” In 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC) (pp. 1249-1254). IEEE.

[34] Arun Kumar, H. D., Prabhakar, C. J. (2015), “Moving Vehicles Detection in Traffic Video Using Modified SXCS-LBP Texture Descriptor”, International Journal of Computer Vision and Image Processing (IJCVIP), Vol 5, No. 2, pp14-34.

[35] Watanabe, T., Ito, S., & Yokoi, K. (2009), “Co-occurrence histograms of oriented gradients for pedestrian detection”, In Advances in Image and Video Technology (pp. 37-47). Springer Berlin Heidelberg.

[36] Heikkilä, M., & Pietikäinen, M. (2002), “A texture-based method for modeling the background and detecting moving objects”, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28(4), 657-662.

[37] Guerrero-Gómez-Olmedo, R., López-Sastre, R. J., Maldonado-Bascón, S., & Fernández-Caballero, A. (2013, June), “Vehicle tracking by simultaneous detection and viewpoint estimation”, In International Work-Conference on the Interplay Between Natural and Artificial Computation (pp. 306-316). Springer Berlin Heidelberg.

[38] Smith, K., Gatica-Perez, D., Odobez, J. M., & Ba, S. (2005, June), “Evaluating multi-object tracking”, In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops (pp. 36-36). IEEE.

[39] i-Lids: http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html

Authors

Arun Kumar H. D. Received M.Sc. degree in computer Science from Kuvempu University, Karnataka, India in 2009, He is pursuing Ph.D. degree in Kuvempu University, Karnataka, India. His research interests are image and video processing, Computer Vision and Machine Vision

Prabhakar C.J. received Ph.D. degree in Computer Science in the year 2009 from Gulbarga University, Gulbarga, Karnataka, India. He is currently working as Assistant Professor in the department of Computer Science, Kuvempu University, Karnataka, India. His research interests are computer vision, Image and video processing.