Dynamic Multi-Vehicle Detection and Tracking from a Moving Platform

Chung-Ching Lin *and Marilyn Wolf
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332
cclin@gatech.edu, wolf@ece.gatech.edu

Abstract

Recent work has successfully built the object classifier for object detection. Most approaches operate with a predefined class and require a model to be trained in advance. In this paper, we present a system with a novel approach for multi-vehicle detection and tracking by using a monocular camera on a moving platform. This approach requires no camera-intrinsic parameters or camera-motion parameters, which enable the system to be successfully implemented without prior training. In our approach, bottom-up segmentation is applied on the input images to get the superpixels. The scene is parsed into less segmented regions by merging similar superpixels. Then, the parsing results are utilized to estimate the road region and detect vehicles on the road by using the properties of superpixels. Finally, tracking is achieved and fed back to further guide vehicle detection in future frames. Experimental results show that the method demonstrates significant vehicle detecting and tracking performance without further restrictions and performs effectively in complex environments.

1. Introduction

Detecting and tracking objects are very important topics for surveillance and monitoring technologies. Nowadays, a camera on a moving platform is pervasive. More and more researchers are working on detecting and tracking objects using moving cameras. The task we address in this paper is dynamic scene analysis for detecting and tracking multiple vehicles from a moving, camera-equipped platform. We seek to detect other traffic participants in the environment. Such capability has obvious applications in car-mounted, wearable, and hand-held camera systems to ensure people's safety.

Recent work has successfully built the object classifier for object detection. Most approaches operate with a predefined class and require a model to be trained in advance. Such work remain trapped in a "closed universe" recognition paradigm, but a much more exciting paradigm of "open universe" datasets, promises to become dominant in the very near future [10].

Some approaches of scene analysis from a moving vehicle require multi-viewpoint, multi-category object detection [6]. Those approaches use 3D depth information as a reference to analyze the scene. Further, some approaches develop systems to detect and track objects by combining information from multiple types of sensors, e.g., laser, sonar, etc. However, multiple cameras and multiple sensors may not be always available for a moving platform. An approach to be applied on a monocular camera can enable universal application. In addition, the scene from a moving vehicle is changing all the time and has great variety. Thus, off-line training methods have some limit on those applications. An analysis method that does not need off-line training provides a good alternative in case where the trained classifier fails.

In order to overcome such problems and generate effective results without the above-mentioned restrictions, we present a system with a novel approach for multi-vehicle detection and tracking using a monocular camera on a moving platform without knowing camera-intrinsic parameters or camera-motion parameters. We propose an analysis method for detecting and tracking vehicles on the road using superpixel properties.

1.1. Related Work

While a great research effort has been dedicated to detection algorithms using steady cameras, it is difficult to generally apply existing methods to detect vehicles from videos captured by a mono camera on a moving platform. Yamaguchi et al. [12] propose a road region detection method by estimating the 3D position of feature points on the road. The feature points and epipolar lines are utilized to detect moving objects. This method assumes that there is no moving obstacle in the initial frame and that the road region in the initial frame is decided according to the height of

*Chung-Ching Lin was with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA. He is now with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598.

the camera that is measured when the vehicle is stationery. However, when these assumptions are violated, the application of this method would be restricted due to presence of moving obstacles in the initial frame or change of camera height. Kang et al. [4] use multiview geometric constrains to detect objects. However, the approach is non-causal since future information is required in this approach. Ess et al.[1] develop a robust algorithm for detecting and tracking pedestrians from a mobile platform. However, this algorithm is developed for a stereo rig, and the calibration of the stereo rig is required in order to use depth information in this algorithm. Leibe et al. [6] estimate structure-from-motion (SfM) and scene geometry with stereo rig. Then, multiple trained models are used to obtain 3D localization and trajectory. The classifiers used in those methods need to be trained off-line. One of the main disadvantages of off-line training method is the need to collect and train data in advance for a specific application.

Sivaraman and Trivedi [9] presented a comparative study about on-road vehicle detection. For some object detection work introduced in [9], active learning is used for object detection to train good classifiers with less data and to minimize human annotation time. In our work, with the same benefits, we develop an integrated algorithm that does not require prior training, and the algorithm can be generally applied to detect and track multiple vehicles.

1.2. Main Contributions

The paper consists of the following main contributions: 1) A novel online system is presented for dynamic scene analysis on videos acquired from an uncalibrated monocular camera on a moving platform. The system requires neither prior training nor multi-camera reference information. Thus, it is capable of adapting to a variety of environments. 2) An efficient method based on superpixel properties is proposed and demonstrates significant performance on challenging video sequences. The method reduces analysis complexity and the challenge of not using pixel-level analysis does not weaken the effectiveness of detection performance. 3) Not only the bounded boxes of the detected objects are provided but also the segmentation of the detected objects is generated. 4) An integrated algorithm is developed for multi-vehicle detection and tracking. The algorithm relaxes the stringent requirement of prior work and requires no information of camera intrinsic or motion parameters. The combination of these strengths enables the system to be generally applied to real-world problems.

1.3. System Overview

In our approach, bottom-up segmentation [2] is applied on the input images to get the superpixels first. The segmentation results provide regional information and make the analysis more efficient than pixel-level analysis. The scene is then parsed into less segmented regions by merging similar superpixels and the parsing result is used to estimate the road region. Next, a classification process is performed to detect the outlier of superpixels on the road region. Superpixel outliers and the lines detected in the scene are used to detect vehicles on the road. Finally, tracking is achieved and fed back to further guide vehicle detection in future frames.

2. Merging

We want to analyze the video streams based on the content of the images. Analyzing video streams using superpixels is more efficient than the pixel-based analysis. Superpixels that are generated by bottom-up segmentation can provide spatial information for aggregating pixels that could belong to a single object and reduce analysis complexity. Usually, some segmentation methods, like [2, 8], generate over segmented results. Segmentations that are similar and belong to the same object are separated by the object's or other vehicles' edges. To achieve efficiency, the segmentations of the same objects should be merged together. Here, we propose a method to merge the similar segmentations that belong to the same vehicle. The likelihood $L_g(S_i, c_j)$ is presented to evaluate the similarity between superpixel S_i and the superpixel group c_j.

$$L_g(S_i, c_j) = \omega_1 L_c(S_i, S_j) + \omega_2 L_f(S_i, S_j) \, \forall S_i \in \xi_{c_j}, \quad (1)$$

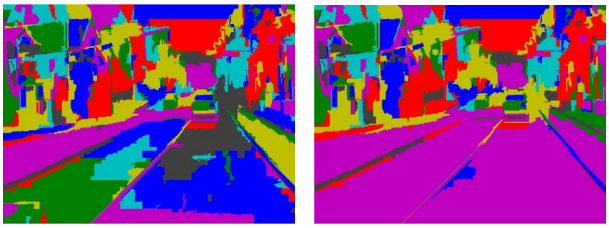
where $L_c(.)$ is the color likelihood, $L_f(.)$ is the feature likelihood, S_i is one of the neighbor superpixels of group c_j, S_j is the initial superpixel in group c_j, ξ_{c_j} is the set of neighbor superpixels of group c_j, and ω_i is the weighting. When computing the likelihood, superpixel S_j is used to compare with superpixel S_i. The merging process for c_j is initiated from the largest superpixel and preformed in descending size order. Simply using an RGB image model cannot deal with shadow problems. Therefore, the HSV image model is also used in the color likelihood to help the merging process. The means and variances of the RGB and HSV values in superpixels are calculated. The means and variances are taken as Gaussian random numbers. The logarithm of the probability $P_c(S_i|S_j)$ is used as the color likelihood $L_c(S_i, S_j)$. $P_c(S_i|S_j)$ is modeled by a normal distribution. For the feature likelihood $L_f(.)$, the feature used here is the coefficients of the Walsh-Hadamard (WH) transform, and the size of the WH transform is determined by the superpixel's size. The logarithm of the conditional inference probability for the WH feature is used as the feature likelihood and defined as follows:

$$-log\left(P_{WH}(S_i|S_j)\right) = \sum_i \left\| f_{WH}(S_i) - f_{WH}(S_j) \right\|_1, \quad (2)$$

where $f_{WH}(S_i)$ is the mean of the WH feature in superpixel S_i and $\|\cdot\|_1$ is 1-norm. The popular histogram of the

oriented gradients (HOG) feature is not utilized to compare the similarity of superpixels for merging because the road region does not have rich texture. The merging process is recursive. After every round of merging, the new neighbors of c_j will be inspected until no more merging can be done. Since superpixels are bounded by edges, the superpixels are expanded for multiple pixels by morphological operation in order to find similar neighbors. When the likelihood $L_g(S_i, c_j)$ is high, we merge the superpixel S_i into group c_j. After this process, the similar neighboring superpixels are merged together.

Fig. 1(a) shows the segmentation results of superpixels. As one can see, the segmentation results are over segmented. After the merging process, better segmentation results are obtained and are shown in Fig. 1(b).



(a) (b)

Figure 1. Superpixel grouping (a) before grouping, (b) after grouping

3. Classification

In this section, the grouping results and the lines detected in the image are utilized to detect the road region. After the grouping process, the groups on the bottom of the field of view (FOV) are taken as the candidate of road region with the assumption that the camera is facing forward in the car. The lines around the boundary of superpixels that are close to the bottom FOV are detected by using dilate/erode morphological operations. The line detection module is based on the approach in [11], which is more reliable than using the normal Hough transform detection method. In Fig. 2(a), red lines show the lines around the boundary of superpixels that are close to the bottom FOV. If lines around the superpixel boundary are detected, the intersections are calculated and the mode of the density is taken as the horizon of the FOV (as shown in Fig. 2(b)). If the mode is outside the FOV or no lines are found, the boundary of the superpixel groups near the bottom FOV is used to calculate the intersections and the mode is used as the horizon.

Next, we are able to identify the road region by using the horizon position and the extreme values of the superpixel boundary group near the bottom FOV. After the road region is defined, the superpixels on the road region are examined and classified as inliers or outliers of the road with the following classification likelihood $L(.)$.

$$L(S_i, R_k) = max_j\{L_{RGB}(S_i, S_j)\},\ \forall S_j \in R_k, \quad (3)$$

where S_i is the i^{th} superpixel in the detected road region, R_k is the superpixel group near the bottom FOV, $L(S_i, R_k)$ is the likelihood value for S_i and R_k, and $L_{RGB}(S_i, S_j)$ is the likelihood value computed by summing the absolute difference of the RGB mean and variance of S_i and S_j. R_k contains multiple superpixels that are grouped together using Eq. 1. With the highest computed $L_{RGB}(S_i, S_j)$, $L(S_i, R_k)$ is used to find the superpixel S_j in R_k that matches best with superpixel S_i on the road region. Then, the chroma-luminance (CL) relation of S_i and S_j is applied to classify superpixel S_i as a outlier or inlier of the road.

$$S_i \text{ and } S_j \text{ are CL-similar iff }, \max \begin{pmatrix} |\bar{R}_i - \bar{R}_j| \\ |\bar{G} - \bar{G}_j| \\ |\bar{B} - \bar{B}_j| \\ |\sigma_{R,i} - \sigma_{R,j}| \\ |\sigma_{G,i} - \sigma_{G,j}| \\ |\sigma_{B,i} - \sigma_{B,j}| \end{pmatrix} \leq \varepsilon \quad (4)$$

The CL-similar relation defines the required CL affinity between same-region superpixels. The CL threshold, ε, constrains the region member superpixel equivalence.

(a) (b)

Figure 2. Horizon detection (a) lines around the boundary of superpixels that are close to the bottom FOV, (b) density of intersections

4. Detection

This section presents the method to detect vehicles on the road region. With the information of the horizon and superpixels on the bottom of the FOV, the road region can be defined. Given the observations Z, the probability of the hypothesis h with vehicle positions and sizes can be expressed as follows:

$$p(h|Z) = p(h|R, l) \quad (5)$$
$$= p(h|S_i, \psi) \quad (6)$$
$$= p(h|S_o, \zeta) \quad (7)$$

where R is the detected road region, l is the set of lines detected on the image, S_i is the superpixels on the road region,

ψ is the set of lines on the road region, S_o are the superpixels that are outliers of the road, and ζ is the grouped lines. As shown in Eq. 7, computing the probability h given Z is equal to computing the probability of h given the road region R and detected lines l. Also, computing $p(h|Z)$ is equal to computing the probability of h given the superpixels S_i and lines ψ on the road. In our system, we compute the probability of h given the superpixels that are outliers S_o and grouped lines ζ. The length and dispersion of the lines in ζ provide information of the vehicle size and position. Next, for every hypothesis h, we compute the probability of the outlier ratio by using the validation potential. The following sections describe the method to group lines and the computation of the validation potential.

4.1. Line Grouping

We use the detected lines to provide cues for detecting vehicles on the road. If a scene is classified geometrically, like [3], we make the reasonable assumption that each segmented region is belong to either the vertical or horizontal category. Road region is classified in the horizontal class and detected vehicles, like cars, on the road are classified in the vertical class. Since cars are symmetric, the detected lines that belong to a car have the same angle and the centroid of each detected line is on the same vertical y-axis. That means the centroids have the same x position but different y positions. Therefore, the lines with the centroid at a similar horizontal x position are grouped and the regions of grouped lines are considered as possible vehicles on the road. In addition, the length and dispersion of the grouped horizontal lines provide information about the bounded boxes of the vehicles.

An agglomerative hierarchical clustering algorithm is used for grouping lines with the following constraints:

$$\theta_k - \theta_{k'} < th_\theta \tag{8}$$

$$d_{cen,x} < th_{cen,x}, \tag{9}$$

where θ_k is the angle of the detected line k, and $d_{cen,x}$ is the distance between the x positions of line k and line k'' centroids. By clustering, the lines that are close to each other and that have similar horizontal x positions can be grouped together. The width and the height of each possible vehicle are decided by the maximum line length and maximum line dispersion in each grouping region respectively. We validate the accuracy of the vehicle detection results in the following section.

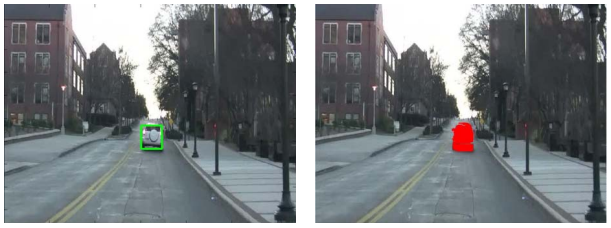
4.2. Validation

The detected outliers of superpixels are used to validate the detected line groups and reduce false detections. A validation potential $\Phi(.)$ is designed to measure the accuracy of the hypothesis h. The validation potential is expressed in terms of the ratio of outliers inside the line group ζ_j:

$$\Phi(h_j|S_o,\zeta_j) = \sum_i \frac{n_{S_{o_i},\zeta_j}}{\vartheta_{\zeta_j}}, \tag{10}$$

$n_{S_{o_i},\zeta_j}$ is the pixel number of the superpixel outliers S_{o_i} inside the area of ζ_j. ϑ_{ζ_j} is the overall pixel numbers in the area of line group ζ_j. We sum up the pixel numbers of outliers in the bounded box of line group ζ_j, and divide it by the overall pixel numbers ϑ_{ζ_j}. The ratio provides the information of the accuracy of the hypothesis h.

Fig. 3(a) shows the detection result. The vehicle's segmentation can be obtained using the superpixel outliers in the bounded box and shown in Fig. 3(b).



(a) (b)

Figure 3. Detection (a) detected vehicle, (b) vehicle segmentation.

5. Tracking

The tracking system is developed by recursive Bayesian estimation and implemented by the Markov Chain Monte Carlo (MCMC) technique [5, 7]. The state of the target and its observation at time t are denoted as $x_{i,t}$ and z_t respectively. $Z_t = \{z_1, \ldots, z_t\}$ represents the overall observations from time 1 to t. The posterior probability of $x_{i,t}$ given observation Z_t at time t can be derived as

$$p(x_{i,t}|Z_t) = \alpha \cdot p(z_t|x_{i,t}) \cdot \int p(x_t|x_{i,t-1}) \cdot p(x_{i,t-1}|Z_{t-1}) dx_{i,t-1}. \tag{11}$$

Eq. 11 is a recursive form of Bayesian estimation using prior probability $p(x_{i,t-1}|Z_{t-1})$ to estimate posterior probability $p(x_{i,t}|Z_t)$. The estimation also needs observation model $p(z_t|x_{i,t})$, which is a measurement of z_t given $x_{i,t}$. The state transition model $p(x_{i,t}|x_{i,t-1})$ is used as a motion model to predict current state $x_{i,t}$ given previous state $x_{i,t-1}$. This formula is valuable in the visual tracking area and has been used in a lot of work.

For computational efficiency, the MCMC sampling step is used to replace the sequential importance sampling (SIS) step on recursive Bayesian estimation. A set of unweighted samples $\{x^k_{i,t-1}\}^N_{k=1}$ generated by MCMC sampling is used for an approximation of $p(x_{i,t-1}|Z_{t-1}) \approx \{x^k_{i,t-1}\}^N_{k=1}$.

The Metropolis-Hastings (MH) algorithm is used to generate an unweighted sample set $\{x_{i,t}^k\}_{k=1}^N$ with posterior distribution $p(x_{i,t}|Z_t)$.

In the MH algorithm, a proposed move x' is generated by the proposal distribution $q(x', x)$. The move is accepted with an acceptance ratio α where

$$\alpha = \min\left\{1, \frac{\pi(x')q(x', x)}{\pi(x)q(x, x')}\right\}. \quad (12)$$

If rejected, the move x' is discarded and x remains unchanged. In this way, the distribution of the samples generated by MCMC will approximate the desired distribution π. In this paper, the desired distribution is defined as $\pi(x) = p(z_t|x_{i,t}) \sum_{k=1}^N p(x_{i,t}|x_{i,t-1}^k)$. The transition model $p(x_{i,t}|x_{i,t-1})$ is modeled by a Gaussian distribution, $p(x_{i,t}|x_{i,t-1}) \sim N(x_{i,t-1}, \Sigma_1)$, where Σ_1 is the variance.

Our observation likelihood $p(z_t|x_{i,t})$ is designed by using only superpixel information. Observation likelihood $p(z_t|x_{i,t})$ is defined as

$$p(z_t|x_{i,t}) = \sum_h \beta_h (r_{S_{h,t},x_{i,t}} - \hat{r}_{S_{h,t},x_{i,t}}) - \sum_k r_{S_{k,t},x_{i,t}} \quad (13)$$

$$r_{S_{h,t},x_{i,t}} = n_{S_{h,t},x_{i,t}}/n_{S_{h,t}} \quad (14)$$

$$\hat{r}_{S_{h,t},x_{i,t}} = (n_{S_{h,t}} - n_{S_{h,t},x_{i,t}})/n_{S_{h,t}} \quad (15)$$

$$r_{S_{k,t},x_{i,t}} = n_{S_{k,t},x_{i,t}}/n_{S_{k,t}} \quad (16)$$

where β_h is the weighting assigned to superpixel $S_{h,t}$. $S_{h,t}$ is a superpixel that is CL-similar with one of the superpixels $S_{q,t-1}$ belongs to the vehicle $x_{i,t-1}$ in the previous frame. $S_{k,t}$ is a superpixel that is not CL-similar with any superpixels belonging to the vehicle $x_{i,t-1}$ in the previous frame. $n_{S_{h,t},x_{i,t}}$ is the number of pixels that are in the area of $x_{i,t}$ and belonging to superpixels $S_{h,t}$. $n_{S_{h,t}}$ is the total number of pixels belonging to superpixels $S_{h,t}$. $n_{S_{k,t},x_{i,t}}$ is the number of pixels that are in the area of $x_{i,t}$ and belonging to superpixels $S_{k,t}$. $r_{S_{h,t},x_{i,t}}$ is the percentage of the superpixel $S_{h,t}$ covered by the area of state $x_{i,t}$. $\hat{r}_{S_{h,t},x_{i,t}}$ is the percentage of the superpixel $S_{h,t}$ not covered by the area of state $x_{i,t}$. $r_{S_{k,t},x_{i,t}}$ is the percentage of $S_{k,t}$ covered by the area of state $x_{i,t}$.

When there are more superpixels in the area $x_{i,t}$ that are CL-similar with the superpixels belonging to the vehicle in the previous frame, the value of the first summation gets larger. On the other hand, $p(z_t|x_{i,t})$ is penalized by the leaking ratio $\hat{r}_{S_h,x_{i,t}}$. And, the weighting β_h is the area of superpixel $S_{q,t-1}$ divided by the overall area of the vehicle in the previous frame. That means the weighting is proportional to the size of superpixel $S_{q,t-1}$. The second summation means that the value of $p(z_t|x_{i,t})$ is penalized by the pixel number of the outlier in the area of $x_{i,t}$.

6. Experiments

We test the proposed algorithm on a variety of real-world videos. The video streams were captured by a camera in a forward-moving car and the camera was held by a human hand. In Fig. 4, 5 and 6, the video streams are captured around the urban area. The car speed is about 10∼35 MPH. The videos are recorded at a frame rate of 30Hz and the resolution of 640x480 pixels. Because the road is uneven and the human hand is unstable, the captured video streams have a lot of sudden irregular movements. The relative movements between vehicles and the camera are complex and change rapidly. In Fig. 7, the video streams are captured on the highway with lower resolution of 384x288 pixels.

For the tracking system, 100 particles are used, and the length of the thinning interval is five ($N = 100$, $M = 5$). We model the proposal distribution $q(x, x')$ by a Gaussian distribution. All variances are set to be proportional to the vehicle size.

Fig. 4 and 5 show the multi-vehicle detection and tracking results of experiment 1 and 2. The detection results are shown in the top row, and the tracking results are shown in the bottom row. Both video streams were taken near the intersection, and there are road marks in the scenes. The figures in the top row show that the proposed method is able to perform significant detection performance regardless of the vehicles are moving forward or toward the camera. Fig. 5(a)(b)(c) show that our system can still successfully detect the vehicles even when the vehicles become smaller. Fig. 4(d)(e)(f) show the frames of tracking after the detection task. The trackers follow detected vehicles quite well after the detection task is terminated. Fig. 5(d)(e)(f) show the frames of tracking in the middle of the detection task. They present robust tracking performance with only small biases in vehicle position and size. In the middle of detection, the tracking can be associated with detected vehicles and enhance the detection power and robustness of the system.

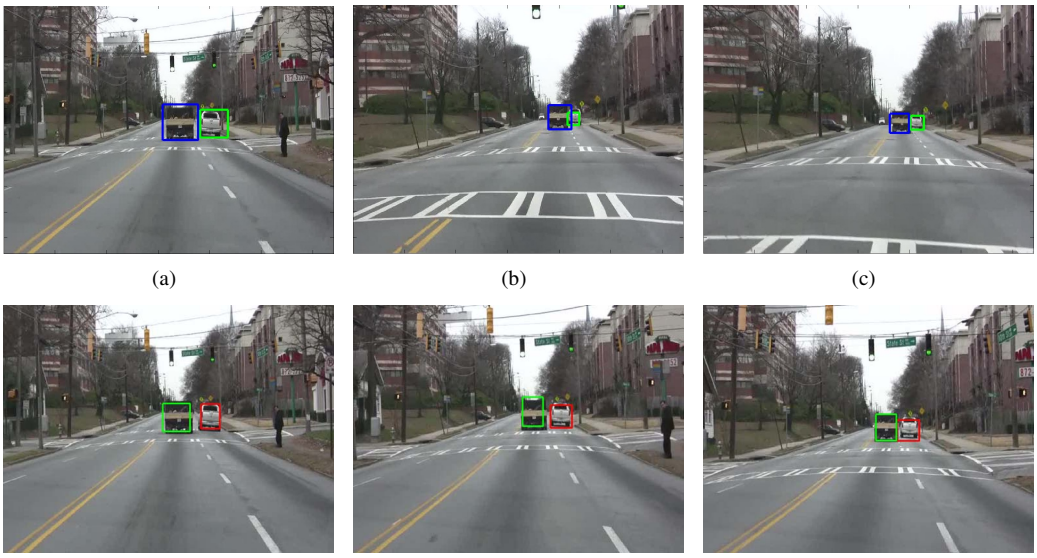
The results of experiment 3 are shown in Fig. 6. The scenario is challenging since vehicles are very cluttered. The vehicle at the right side of the road is a parked car and the middle car is partially occluded by both cars at its right side and left side. The top row of Fig. 6 shows that our system is fairly robust to deal with these tough cases. The bottom row shows the tracking result. As one can see in Fig. 6(e)(f), only two cars are tracked since there are only two detected cars in Fig. 6(d). Once the incoming car is detected, a new tracker would be initiated.

Fig. 7 shows another example of robust detection. Experiment 4 is performed on a challenging video stream captured in a car at speed over 60 mph on the highway with a lower resolution than the previous experiments As one can see, the camera was not facing directly forward but face a little toward the left side of the car. Our system can still estimate the road region and accurately detect the vehicles on

(a) (b) (c)

(d) (e) (f)

Figure 4. Experiment 1 Detection: (a) frame 83, (b) frame 104, (c) frame 226; Tracking: (d) frame 272, (e) frame 278, (f) frame 305.



(a) (b) (c)

(d) (e) (f)

Figure 5. Experiment 2 Detection: (a) frame 16, (b) frame 94, (c) frame 115; Tracking: (d) frame 9, (e) frame 31, (f) frame 51.

the road.

7. Conclusions and Discussion

In this paper, we proposed a novel method to effectively detect vehicles on the road from videos captured by a camera on a moving platform. The vehicles can be detected without using any camera intrinsic and motion parameters. Experiment results show the proposed method has significant detecting and tracking performance. There is no need to impose initial assumptions or to apply future frame information in the detecting algorithm. And, the online analysis method can adapt to various environments. Thus, the proposed method could be generally applied to detect and track vehicles with irregular camera movement and in complex environment. Future research is aimed at extending our algorithm for auto video segmentation.

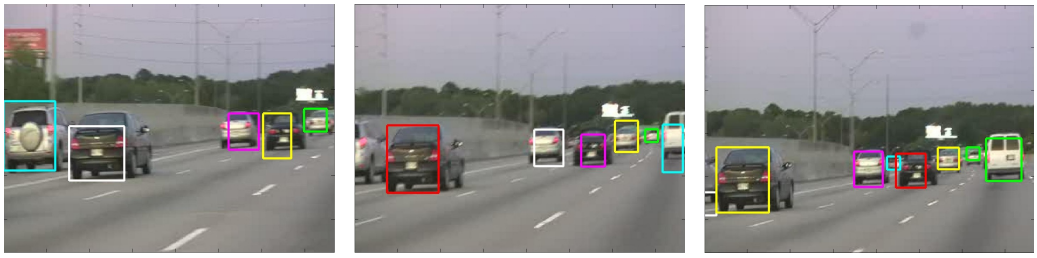
References

[1] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Robust Multi-Person Tracking from a Mobile Platform. *Pat-*

(a) (b) (c)

(d) (e) (f)

Figure 6. Experiment 3 Detection: (a) frame 123, (b) frame 177, (c) frame 213; Tracking: (d) frame 126, (e) frame 153, (f) frame 171.



(a) (b) (c)

Figure 7. Experiment 4 Detection: (a) frame 129, (b) frame 133, (c) frame 137.

tern Analysis and Machine Intelligence, 31(10):1831–1846, 2009.

[2] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[3] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1–8. IEEE, 2009.

[4] J. Kang, I. Cohen, G. Medioni, and C. Yuan. Detection and tracking of moving objects from a moving platform in presence of strong parallax. In *IEEE International Conference on Computer Vision*, 2005.

[5] Z. Khan, T. Balch, and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(11):1805–1819, 2005.

[6] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3d scene analysis from a moving vehicle. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[7] C. Lin and W. Wolf. MCMC-based Feature-guided Particle Filtering for Tracking Moving Objects from a Moving Platform. In *IEEE International Conference on Computer Vision Workshop*, 2009.

[8] F. Meyer. Topographic distance and watershed lines. *Signal Processing*, 38(1):113–125, 1994.

[9] S. Sivaraman and M. M. Trivedi. Active learning for on-road vehicle detection: a comparative study. *Machine Vision and Applications*, pages 1–13, 2011.

[10] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *European Conference of Computer Vision*, pages 352–365. Springer, 2010.

[11] R. von Gioi, J. Jakubowicz, J. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(4):722–732, 2010.

[12] K. Yamaguchi, T. Kato, and Y. Ninomiya. Vehicle ego-motion estimation and moving object detection using a monocular camera. In *IEEE International Conference on Pattern Recognition*, volume 4, 2006.