Vehicle Tracking Using On-Line Fusion of Color and Shape Features

Kai She[1], George Bebis[1], Haisong Gu[1], and Ronald Miller[2]
[1]Computer Vision Laboratory, University of Nevada, Reno, NV
[2]Vehicle Design R & A Department, Ford Motor Company, Dearborn, MI
(she,bebis,gu)@cs.unr.edu, rmille47@ford.com

Abstract— **A real-time on-road vehicle tracking method is presented in this work. The tracker builds statistical models for the target in color and shape feature spaces and continuously evaluates each of the feature spaces by computing the similarity score between the probabilistic distributions of the target and the model. Based on the similarity scores, the final location of the target is determined by fusing the potential locations found in different feature spaces together. The proposed method has been evaluated on real data, illustrating good performance.**

I. Introduction

With the aim of reducing injury and accident severity, pre-crash sensing is becoming an area of active research among automotive manufacturers, suppliers and universities [1][2][3]. Developing on-board automotive driver assistance systems aiming to alert a driver about driving environments, and possible collision with other vehicles has attracted a lot of attention. In these systems, robust and reliable vehicle detection and tracking are critical steps for driver assistance and collision avoidance. The focus of this paper is on the problem of on-road vehicle tracking using optical sensors. Our past work on vehicle detection can be found in [4]-[9] while a review on on-road vehicle detection methods using optical sensors can be found in [10].

Tracking moving objects using a moving camera has been a challenging topic in computer vision. The difficulties are caused by continuous changing of the camera position, variation of the appearance of the target in motion, and alteration of the illumination conditions and background. In particular, the camera-assisted car is required to react to the on-road situations in real time, which adds one more constraint to the tracker: the tracking algorithm has to be computationally inexpensive.

Various tracking algorithms have been proposed in the literature, including approaches using optical flow, [11][12], templates and local features [13][1], Kalman filters [14][15] and contours [16][17]. The mean-shift algorithm was first adopted as an efficient tracking technique in [18].

As a nonparametric gradient climbing method, mean-shift can be used to analyze arbitrarily structured feature spaces, which are hard to fit any analytical models in. Compared to the tracking methods which exhaustively search the neighborhood of the predicted location of the target, mean-shift does an optimal search for fast target localization. To predict the search region more effectively, mean-shift was combined with Kalman filter in [19]. To improve the ability of mean-shift tracking with respect to scale changes, a scale selection mechanism was introduced in [20] based on Lindeberg's theory. Mean-shift has been used in the past to track moving targets in sequences of FLIR imagery [21] and human bodies (i.e., non-rigid objects) in [22].

The success or failure of any tracking algorithm depends a lot on the degree that the tracked object can be distinguished from its surroundings [23]. In particular, the set of features used by the tracking algorithm to represent the object(s) being tracked plays a major role in tracking performance. All of the mean-shift based trackers mentioned above use only color information to form the visual feature space where the probability models are built. Using color information alone is not sufficient for on-road vehicle tracking. Competing clutters having similar colors as the target may appear in the background quite often. Moreover, the color of the target being tracked varies due to light changes and road conditions. For example, when a car goes through a tunnel, it turns dark grey despite of its original color due to drastic illumination changes. A tracker solely based on color information will be deceived during this situation. On the other hand, tracking could suffer less if using cues other than color such as shape information (i.e., vehicle shape remains rigid in motion). Consequently, robust tracking performance can be gained by switching to shape information when color information becomes less reliable and the opposite.

This paper proposes a robust mean-shift based vehicle tracker, with an on-line feature space evaluation and decision fusion mechanism embedded. The tracker represents the target using color and shape information and adapts the representation to the proper feature space using an on-line decision rule. The statistical representations for the model vehicle and the vehicle candidates are acquired in four different feature spaces: vertical edge, horizontal edge, diagonal edge and color using the HSV (Hue Saturation Value) representation. Mean-shift analysis is then employed in each feature space to derive the potential position of the target. This is done by finding the target having the most similar statistical distribution to a given model in the same feature space. The final target location is determined by fusing the four candidate locations together. Each candidate location is weighted based on the similarity of the target with the model in the corresponding feature space. The proposed tracking algorithm has been evaluated on many video sequences taken on I80 in Reno, Nevada. Our results illustrate robust tracking performance in real time.

II. Tracking Algorithm

Our tracking algorithm takes three major steps to perform the on-road vehicle tracking. Four different features are extracted for each input frame first. In each feature space, a mean-shift estimator finds a potential target position based on the corre-

sponding statistical model. Finally the target location is determined by dynamically fusing the tracking outputs from the four feature spaces.

A. Feature Spaces

In this work, we assume that the most promising features for tracking form the best feature space. This is the space where the target and model statistical distributions are most similar. The best feature space for tracking needs to adapt over time as the appearances of the target and the background keep changing due to the fact that both the target and the camera are moving.

Color and shape provide important information to describe a vehicle. In this work, a separate probabilistic model of the target is built in each of the following four feature spaces: HSV color space, vertical edge space, horizontal edge space, and diagonal edge space. The proposed framework is very general and can support additional feature spaces in the future, such as motion and texture features. The ability of these feature spaces to characterize the target reliably varies from time to time. Characterizing vehicle appearance using shape features is quite robust to illumination changes, however, it could fail when the target is partially occluded. On the other hand, color information helps the tracker to distinguish the object most of the time but deceives the tracker when drastic illumination changes appear. The key issue addressed in this work is developing an automatic mechanism which performs on-line evaluation on how similar the candidate target is to the model in each feature space and combines the tracking results together based on their similarity scores.

A.1 Color Features

First, the input image is transformed from the RGB (Red Green Blue) color space to the HSV color space [24]. HSV space separates out hue (color) from saturation (how concentrated the color is) and value (brightness). The hue image is created using the hue channel in HSV color space. We have observed that when the brightness becomes very low or very high, then the hue becomes noisy and unstable. Therefore, we threshold the brightness to ignore hue pixels with extreme brightness values (i.e., only the hue values with corresponding brightness values greater than 10 and less than 240 are considered).

A.2 Shape Features

To achieve real-time performance, a fast feature extraction method was adopted in this work to extract edge information in different directions. Our motivation is that rear views of a vehicle contain mostly vertical,, horizontal, and diagonal edges. To achieve this, the Haar-like feature extraction method, first introduced in [25] for detecting human faces, was adopted in this study. By computing an integral image first, which is an intermediate representation for the original image, the Haar-like features can be computed very rapidly by just using a lookup table (see [25] for more details). Three simple rectangle masks were used to extract horizontal, vertical and diagonal edge information, as shown in Figure 1.

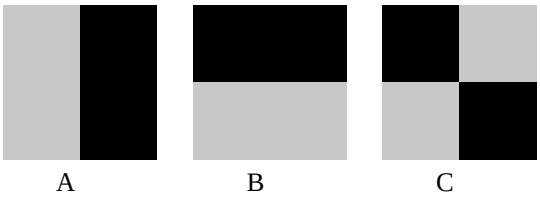


Fig. 1. Basic masks used to extract the Haar-like features. (a) horizontal edge information, (b) vertical edge information, and (c) diagonal edge information.

B. Mean Shift Estimator

In each feature space, a different mean-shift estimator is applied to find the target's potential location. The core of the mean shift tracking algorithm is computing the mean shift vector Δy recursively. The current target centroid location vector y_j and the new target centroid location y_{j+1} are related through a translation:

$$\Delta y \quad = \quad y_{j+1} - y_j \tag{1}$$

While a normal kernel was adopted in [22], the new target centroid location is derived as:

$$y_{j+1} \quad = \quad \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \tag{2}$$

where the set $\{x_i\}_{i=1\ldots n}$ represents n pixel locations in the search window and w_i the corresponding weight assigned to each pixel.

B.1 Probabilistic Model Representations

Using a two dimensional Epanechnikov kernel, written as

$$K_E(x) \quad = \quad \frac{2}{\pi h^2}\left(h^2 - \| x \|^2\right) \tag{3}$$

for a given feature u, the kernel density estimates for both of the target model, denoted as q_u, and the target candidates around the current target centroid location y, denoted as $p_u(y)$, can be expressed by

$$p_u(y) \quad = \quad \frac{\sum_{i=1}^{n} K_E(y - x_i)\,\delta\left(b(x_i) - u\right)}{\sum_{i=1}^{n} K_E(y - x_i)} \tag{4}$$

where $b(x_i)$ is a function of the features, computed using color or shape features in our application, and δ is the Kronecker delta function. The denominator in (4) normalizes the probability histogram by imposing the condition $\sum_{u=1}^{m} p_u = 1$, while m represents the total number of features.

B.2 Distance Measure and Minimization

In order to find the target candidate whose density distribution is most similar to the model, we need an appropriate distance metric to measure the similarity between two histogram distributions. The Bhattacharyya coefficient is a near optimal choice due to its close relation to the Bayes error and its properties are illustrated in [26]. The distance between two m-bin histogram is defined as

$$d(y) \quad = \quad \sqrt{1 - \rho(y)} \tag{5}$$

The Bhattacharyya coefficient $\rho(y)$ is given by

$$\rho(y) \quad \equiv \quad \rho\left[p(y), q\right] \; = \; \sum_{i=1}^{m} \sqrt{p_u(y) q_u} \tag{6}$$

where p and q indicate the target and model distributions respectively. To minimize the distance metric (5), the Bhattacharyya coefficient has to be maximized. After plugging in (4) for $p_u(y)$ and q_u, the Bhattacharyya coefficient can be approximated using Taylor expansion as

$$\rho\left[p\left(y\right),q\right] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{p_u\left(y_0\right)q_u} + \frac{C}{2}\sum_{i=1}^{n} w_i K_E\left(y-x_i\right) \quad (7)$$

where

$$w_i = \sum_{u=1}^{m}\delta\left[b\left(x_i\right)-u\right]\sqrt{\frac{q_u}{p_u\left(y_0\right)}} \quad (8)$$

Assigning the weights given in (8) to each of the pixels in the search window, the new location of the target center is obtained by (2). The maximization of the Bhattacharyya coefficient is achieved by the mean-shift iterator. The target locations found using different feature spaces are then fused together to determine the final target location. We discuss our fusion strategy in the next subsection.

C. Feature Set Evaluation and Decision Fusion

The feature set evaluation and fusion procedure are illustrated in Figures 2 and 3 respectively. To combine the four possible target locations found by the four different mean-shift estimators, we compute the similarity of each statistical distribution at the candidate locations with the model distribution using the Bhattacharyya coefficient. The higher the Bhattacharyya coefficient is, the more likely is that the model appears at the target's candidate location. The feature space providing the highest similarity will have the most contribution in determining the final location of the target candidate. By normalizing the Bhattacharyya coefficients, we can interpret their values as the probability that the model is present at the candidate location.

In Figure 2, we initialize the model by detecting the target to be tracked (top row). The pictures in the second row illustrate the results of feature extraction in the four different feature spaces. From left to right, they are: vertical edge map, horizontal edge map, diagonal edge map, and thresholded hue map. Then, we generate four statistical distributions for the model in each of the corresponding feature space as shown in the third row. These are simply the normalized histograms of edge magnitude and hue information.

Figure 3 is an example of the feature evaluation and fusion procedure for a given input image. For every input frame, the weighted images shown in the third row are computed using the statistical models shown in the second row. Then, the location of the target is estimated using mean-shift in each of the four feature spaces. After the four possible locations of the target are found by performing mean-shift mode seeking on the weighted images, they are fused together to determine the final location of the target.

Let the four possible locations found in the vertical edge, horizontal edge, diagonal edge and color feature spaces be:

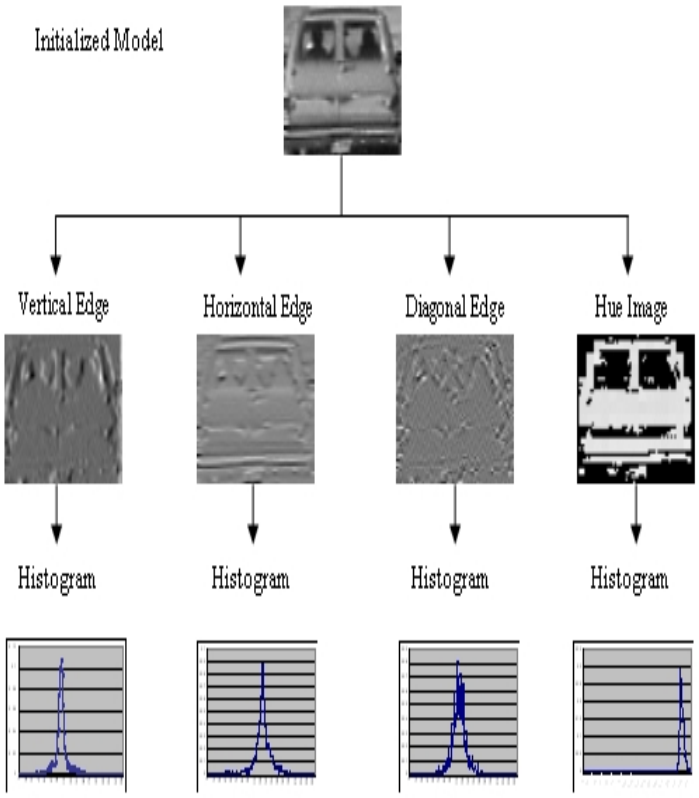


Fig. 2. Probability distributions generated for the model at initialization

$x_v, y_v, x_h, y_h, x_d, y_d$ and x_c, y_c respectively. Also, let the corresponding Bhattacharyya coefficients be: BC_v, BC_h, BC_d and BC_c. Then, the final center location (x, y) is determined as follows:

$$x = \frac{BC_v}{BC_v+BC_h+BC_d+BC_c} * x_v + \frac{BC_h}{BC_v+BC_h+BC_d+BC_c} * x_h + \frac{BC_d}{BC_v+BC_h+BC_d+BC_c} * x_d + \frac{BC_c}{BC_v+BC_h+BC_d+BC_c} * x_c \quad (9)$$

$$y = \frac{BC_v}{BC_v+BC_h+BC_d+BC_c} * y_v + \frac{BC_h}{BC_v+BC_h+BC_d+BC_c} * y_h + \frac{BC_d}{BC_v+BC_h+BC_d+BC_c} * y_d + \frac{BC_c}{BC_v+BC_h+BC_d+BC_c} * y_c \quad (10)$$

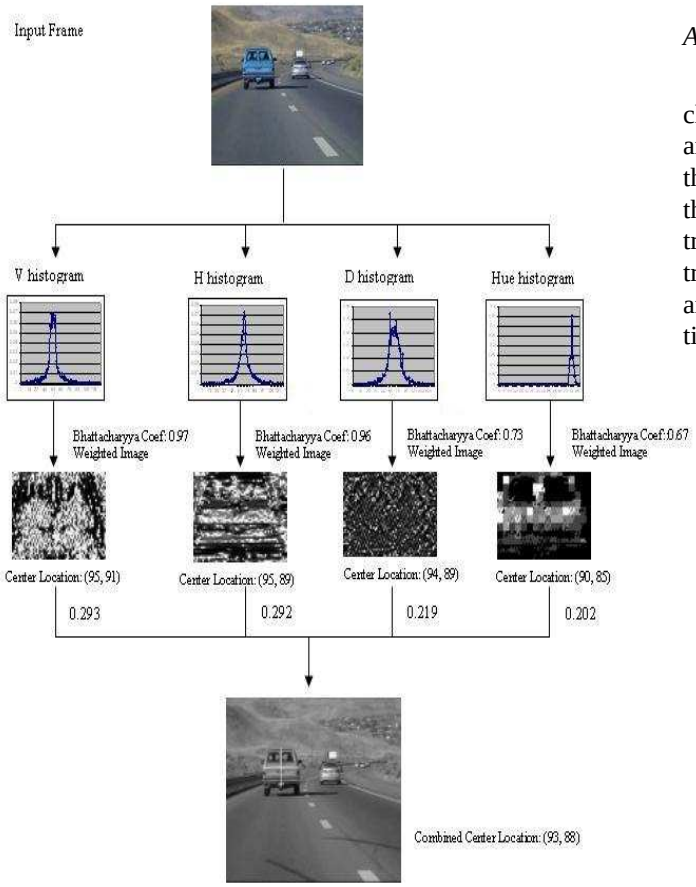


Fig. 3. Fusing the four possible target locations together for each of the input new frame

D. Scale and Model Adaptation

The tracking algorithm iterates through each image frame from the video sequence, extracting different feature sets as described in the previous subsection. The search window in the mean-shift algorithm is updated every 20 frames. After selecting the feature cue which best describes the target object, the scaling is done by simply modifying the radius of the search window by a certain fraction, such as $\pm 10\%$. The window size which provides the largest Bhattacharyya coefficient is then chosen to contain the target.

The statistical representations of the models being tracked in different feature spaces need to be updated since the target and the camera are moving over time. A proper threshold was chosen empirically to decide when the model needs to be adapted in a certain feature space. In particular, when computing the Bhattacharyya coefficients every 20 frames, their values are compared to the threshold. Based on the result of this comparison, the statistical distributions of the model in each of the feature spaces is updated.

III. Experiments

The proposed tracker has been tested under various highway tracking scenarios. To evaluate and compare different tracking approaches, we compared tracking results with ground truth information using a set of video sequences.

A. Tracking Performance Evaluation

For objects having rectangular shape like the rear of vehicles, their position and size can be represented by the upper left and the bottom right corners of a rectangular window enclosing the object. To evaluate the accuracy of tracking, we compute the overlapping ratio between the rectangular area found by the tracker and the rectangular area picked by a human (i.e., ground truth). As shown in Figure 4, let us assume that rectangles A and B represent the ground truth and tracking results respectively. Then, the overlapping ratio r is computed as:

$$r = \frac{2 * C}{A + B} \tag{11}$$

where $C = A \cap B$.

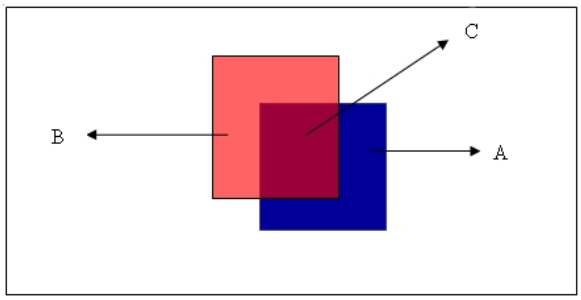


Fig. 4. Overlapping area between the tracking result and the ground truth

B. Results

In this section, we present several tracking results to demonstrate the performance of the proposed algorithm. The video sequences were taken on I80 in Reno, Nevada. The video sequences were chosen to test the tracker under challenging conditions including scale changes, illumination changes, and partial occlusion. For each sequence, we compare tracking results with ground truth and we plot the overlapping ratio r over time.

A blue truck, shown in Figure 5, is being tracked in video sequence 1. The truck is driving under normal conditions and there is no occlusion or drastic illumination changes. The size of the target vehicle changes gradually through the sequence as it moves further away from the host vehicle. Figure 6 shows the tracking results in each feature space separately as well as using on-line fusion. While the five tracking methods have comparable performances, the tracker using on-line fusion (shown in orange) has the best performance.

The second video sequence contains a white van, shown in Figure7(a). The corresponding overlapping ratios are shown in Figure 8. The tracking results show that using color cue alone is not enough to produce good results, and that the tracker fails to track the vehicle when it goes under a bridge and its color changes drastically as shown in Figure 7(b). Figure 7(c) shows the tracking results using on-line fusion. While the accuracy of tracking using color alone drops for more than 50% due to significant illumination changes, shape features still preserve well the structure of the target. As a result, the tracker that employs

(a)
Manually initialized model.

(b)
Tracking using on-line feature fusion tracks the target correctly when a truck in brighter color merges in front.

(c)
Successful tracking with scale changes.

Fig. 5. Video sequence contains scale changes.

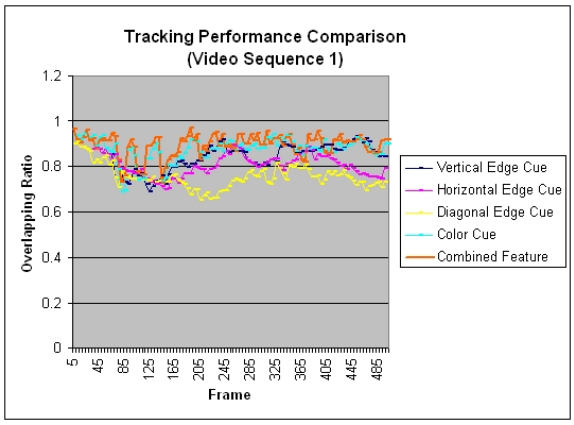


Fig. 6. Tracking accuracy for video sequence 1 using individual features and on-line feature fusion.

both color and shape features maintains high performance as shown in Figure 8 (orange line).

The last video sequence contains a blue SUV, shown in Figure 9(a), and presents challenges due to partial occlusion. As shown in Figure 10, using shape information alone is not sufficient and tracking drifts away from the target when the vehicle is partially visible. Figure 9(b) shows the results of tracking using vertical edge information only. However, when combining color with shape features, tracking is successful as shown in Figure 9(c).

Overall, on-line fusion of color and shape features provides a more robust approach for on-road vehicle tracking. Shape information compensate for color variations due to illumination changes, while color information compensate for shape changes due to partial occlusion.

IV. Conclusion

We presented an effective on-road tracking method which utilizes both shape and color information. Using color information and shape features based on edges in three directions, we build statistical models of vehicle appearance in each feature space. Then, the potential target location is found in each feature space using the mean-shift algorithm. The results in each feature space are ranked by computing similarity scores between the model and the target. Fusion takes place on-line and uses the similarity

(a)
Manually initialized model.

(b)
Tracking using color information alone fails due to illumination changes.

(c)
Tracking using on-line feature fusion.

Fig. 7. Video sequence contains drastic illumination changes.

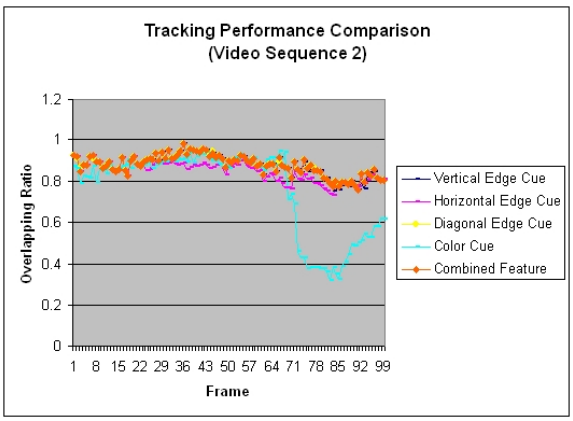
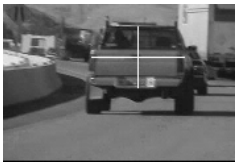
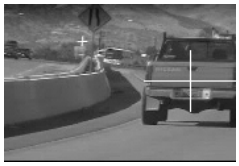


Fig. 8. Tracking accuracy for video sequence 2 using individual features and on-line feature fusion.



(a)
Manually initialized model.

(b)
Tracking using vertical edge information alone fails due to partial occlusion.



(c)
Tracking on-line feature fusion.

Fig. 9. Video sequence contain partial occlusions.

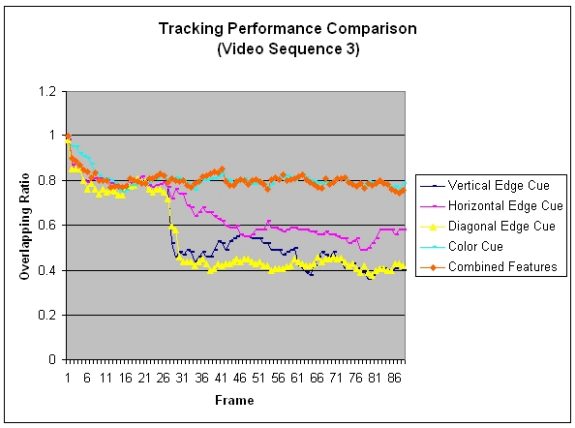


Fig. 10. Tracking accuracy for video sequence 3 using individual features and on-line feature fusion.

scores to compute the final location of the target. Our experimental results demonstrate robust tracking performance under various tracking scenarios, including scale changes, drastic illumination changes, and partial occlusion.

For future research, we plan to consider fusing more cues such as texture and optical flow. Since the mean shift algorithm is applied separately in each feature space, the proposed approach is amenable to a parallel implementation which implies real-time performance.

Acknowledgements: This research was supported by the Ford Motor Company under grant No.2001332R, the University of Nevada, Reno under Applied Research Initiative (ARI) grant, and in part by NSF under CRCD grant No.0088086.

REFERENCES

[1] M. Betke, E. Haritaoglu and L. S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle," *Machine Vision and Applications*, vol. 12, no. 2, pp. 69–83, September, 2000.

[2] Alan J. Lipton, Hironobu Fujiyoshi and Raju S. Patil, "Moving target classification and tracking from real-time video," *DARPA Image Understanding Workshop*, 1998.

[3] Gildas Lefaix, Eric Marchand and Patrick Bouthemy, "Motion-based obstacle detection and tracking for car driving assistance," *Int. Conf. on Pattern Recogntion*, vol. 4, pp. 74–77, Canada, August 2002.

[4] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using gabor filters and support vector machines," *International Conference on Digital Signal Processing*, July, 2002,Greece.

[5] Z. Sun, G. Bebis, and R. Miller, "Quantized wavelet features and support vector machines for on-road vehicle detection," *Seventh International Conference on Control, Automation, Robotics and Vision*, December, 2002, Singapore.

[6] Z. Sun, G. Bebis, and R. Miller, "Improving the performance of on-road vehicle detection by combining gabor and wavelet features," *IEEE International Conference on Intelligent Transportation Systems* , September, 2002, Singapore.

[7] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, "A real-time precrash vehicle detection system," *IEEE International Workshop on Application of Computer Vision*, Dec., 2002.

[8] Z. Sun, G. Bebis, and R. Miller, "Boosting object detection using feature selection," *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2003.

[9] Z. Sun, G. Bebis, and R. Miller, "Evolutionary gabor filter optimization with application to vehicle detection," *IEEE International Conference on Data Mining*, 2003.

[10] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using optical sensors: A review," *IEEE International Conference on Intelligent Transportation Systems*, 2004.

[11] Kung-Hao Liang and Tardi Tjahjadi, "Multiresolution segmentation of optical flow fields for object tracking," *Applied Signal Processing*, vol. 4, no. 3, pp. 179–187, Springer-Verlag, London, 1997.

[12] Alireza Behard, Ali Shahrokni, Seyed Ahmad Motamedi, "A robust vision-based moving target detection and tracking system," *Proceeding of Image and Vision Computing conference*, Dunedin, New Zeland 26th-28th, November 2001.

[13] Benjamin Coifman and David Beymer, Jitendra Mailik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Trasportation Research: Part C*, vol. 6, no. 4, pp. 271–288, 1998.

[14] Janguang Lou, Hao Yang, Weiming Hu, Tieniu Tan, "Vesual vehicle tracking using an improved ekf," *Asian Conference on Computer Vision*, 2002.

[15] Dieter Koller, Joseph Weber, and Jitendra Malik, "Robust multiple car tracking with occlusion reasoning," *Third European Conference on Computer Vision*, pp. 186–196, Springer-Verlag, 1994.

[16] Esther B. Meier and Frank Ade, "Using the condensation algorithm to implement tracking for mobile robots," *Third European Workshop on Advanced Mobile Robots*, pp. 73–80, Zurich, Awitzerland, 6th-8th September 1999.

[17] Andrea Giachetti, "Applications of contour tracking techniques," *citeseer.ist.psu.edu/7949.html*.

[18] Dorin Comaniciu, Visvanathan Ramesh and Peter Meer, "Real-time tracking of non-rigid objects using mean shift," *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 142–149, Hilton Head Island, South Carolina, 2000.

[19] Dorin Comaniciu and Visvanathan Ramesh, "Mean shift and optimal prediction for efficient object tracking," *IEEE Int. Conf. Image Processing*, vol. 3, pp. 70–73, Vancouver,Canada, 2000.

[20] Robert T. Collins, "Mean-shift blob tracking through scale space," *IEEE Computer Vision and Pattern Recognition*, vol. Madison, WI, pp. 234–240, June 16-22, 2003.

[21] Alper Yilmaz, Khurram Shafique, Niels Lovo, Xin Li, Teresa Olson, Mubarak A. Shah, "Target tracking in flir imagery using mean-shift and global motion compensation," *proceedings of IEEE Workship on Computer Vision Beyond Visible Spectrum*, Hawaii, 2001.

[22] Fatih Porikli and Oncel Tuzel, "Human body tracking by adaptive background models and mean-shift analysis," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, March 2003.

[23] Robert T. Collins and Yanxi Liu, "On-line selection of discriminative tracking features," *IEEE International Conference on Computer Vision*, vol. Nice, France, pp. 346–352, October 2003.

[24] Gary R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *IEEE Workshop on Appliic. Comp. Vis.*, vol. 2, pp. 214–219, Princeton, 1998.

[25] Paul Viola, Michael Jones, "Robust real-time object detection," *Second International Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, Computing, and Sampling*, Vancouver, Canada, July 13, 2001.

[26] A. Djouadi, O. Snorrason, F.D. Garber, "The quality of training-sample estimates of the bhattacharyya coefficient," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, January 1990.