2008

Vehicle tracking by non-drifting mean-shift using projective Kalman filter

Philippe L. Bouttefroy
philippe@uow.edu.au

Abdesselam Bouzerdoum
University of Wollongong, bouzer@uow.edu.au

Son Lam Phung
University of Wollongong, phung@uow.edu.au

Azeddine Beghdadi
Universit´e Paris, beghdadi@galilee.univ-paris13.fr

Publication Details

P. L. M. Bouttefroy, A. Bouzerdoum, S. Lam. Phung & A. Beghdadi, "Vehicle tracking by non-drifting mean-shift using projective Kalman filter," in Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems, 2008, pp. 61-66.

Vehicle tracking by non-drifting mean-shift using projective Kalman filter

Abstract
Robust vehicle tracking is essential in traffic monitoring because it is the groundwork to higher level tasks such as traffic control and event detection. This paper describes a new technique for tracking vehicles with mean-shift using a projective Kalman filter. The shortcomings of the mean-shift tracker, namely the selection of the bandwidth and the initialization of the tracker, are addressed with a fine estimation of the vehicle scale and kinematic model. Indeed, the projective Kalman filter integrates the non-linear projection of the vehicle trajectory in its observation function resulting in an accurate localization of the vehicle in the image. The proposed technique is compared to the standard Extended Kalman filter implementation on traffic video sequences. Results show that the performance of the standard technique decreases with the number of frames per second whilst the performance of the projective Kalman filter remains constant.

Disciplines
Physical Sciences and Mathematics

Publication Details
P. L. M. Bouttefroy, A. Bouzerdoum, S. Lam. Phung & A. Beghdadi, "Vehicle tracking by non-drifting mean-shift using projective Kalman filter," in Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems, 2008, pp. 61-66.

**Proceedings of the 11th International IEEE
Conference on Intelligent Transportation Systems
Beijing, China, October 12-15, 2008**

Vehicle Tracking by non-Drifting Mean-shift using Projective Kalman Filter

Philippe Loic Marie Bouttefroy[1], Abdesselam Bouzerdoum[1], Son Lam Phung[1] and Azeddine Beghdadi[2]

Abstract— **Robust vehicle tracking is essential in traffic monitoring because it is the groundwork to higher level tasks such as traffic control and event detection. This paper describes a new technique for tracking vehicles with mean-shift using a projective Kalman filter. The shortcomings of the mean-shift tracker, namely the selection of the bandwidth and the initialization of the tracker, are addressed with a fine estimation of the vehicle scale and kinematic model. Indeed, the projective Kalman filter integrates the non-linear projection of the vehicle trajectory in its observation function resulting in an accurate localization of the vehicle in the image. The proposed technique is compared to the standard Extended Kalman filter implementation on traffic video sequences. Results show that the performance of the standard technique decreases with the number of frames per second whilst the performance of the projective Kalman filter remains constant.**

I. Introduction

Vehicle tracking has been a focus of attention in the past years due to increasing demand in visual surveillance and security on highways. Robust vehicle tracking provides the groundwork to higher level tasks for Intelligent Transportation System (ITS). Accurate trajectory extraction provides essential statistics for traffic control, such as speed, vehicle count and average vehicle flow. It also enables higher level tasks such as event detection (e.g. accident, animal crossing) or traffic regulation (e.g. dynamic speed adaptation, lane allocation).

There have been several techniques proposed for traffic monitoring in the literature based on motion extraction and vehicle tracking. Because monitoring cameras are fixed, background subtraction techniques provide efficient segmentation of motion areas. Background subtraction by mixture of Gaussians is generally used for this purpose [13] [14], although other techniques such as temporal median operation and filtering appear in the literature [7] [8]. The segmentation provides blobs representing the vehicles in motion. In this context, feature-based tracking [4], and in particular histogram-based tracking, is not suitable due to the small size of the vehicles in the image. Indeed, in the far distance, the apparent size of the vehicles only provides a mere observation of the target color distribution which is not sufficient for robust discrimination upon which feature-based tracking relies. Traffic monitoring algorithms are differentiated by the technique used for tracking vehicles. Even though Bayesian filtering, and in particular Kalman filters, is extensively used to predict the position of the vehicle, the implementation differs in many ways. The state vector can be modeled with data directly available from blobs such as kinematic parameters [1] [10] or scale [7]. Other authors proposed to further process the image to track corners [11] or contours [9] [8] that are fed into the Kalman filter. Choi *et al.* [3] used a quad-tree scale invariant segmentation and template matching to achieve tracking of vehicles. Gloyer *et al.* [5] proposed a 3D model scene to track vehicles through the video sequence.

Vehicle tracking from traffic monitoring presents particular characteristics due to the nature of the video sequences and the vehicle trajectories; some making the tracking more challenging (indicated by "-" below) and some providing restrictive clues facilitating the tracking (indicated by "+" below):

1) *Low definition and highly compressed video sequence* (-) is the result of the information network infrastructure. Restricted bandwidth only allows low bit flows;

2) *Very low frame rate* (-) makes the information about the position of the vehicle sparse due to the restricted bandwidth;

3) *Uniform Vehicle Speed* (+) during the tracking of the vehicle; and

4) *Vehicle trajectory is constrained* (+) by the shape of the road.

The contribution of this paper is to embed projective information in the Kalman filter in order to provide an accurate estimate of the vehicle position. More precisely, the integration of the camera calibration equations and the restrictive clues (marked as "+" above) in the observation model of the Kalman filter leads to efficient and robust tracking of vehicles by reducing the observation noise and providing robustness to the sparsity of the vehicle position. The rest of the paper is organized as follows. Section II develops the background subtraction technique and the mean-shift tracker. Section III introduces the projective Kalman filter and, in particular, the derivation of the observation function in subsection III-B. Section IV presents some of the performances of the algorithm on traffic monitoring sequences before concluding in Section V.

[1] P. L. M. Bouttefroy, A. Bouzerdoum and S. L. Phung are with the University of Wollongong, ECTE School, Northfields Av., 2500 NSW, Australia.

[2] A. Beghdadi is with Université Paris 13, L2TI, Institut Galilée, Av. J. B. Clément 93430 Villetaneuse, France.

II. Motion detection and Mean-shift Blob Tracking

The technique proposed in this paper is based on motion detection by background subtraction and mean-shift blob tracking. First, the motion image is extracted from the video sequence. Then, the mean-shift algorithm is used to track the blobs through the sequence.

A. Background Modeling Using Gaussian Mixtures

Background modeling using Gaussian mixtures is a pixel-based process. For a given location, the distribution of the pixel value (ξ) is modeled by superposition of a set of independent Gaussian distributions. The probability density function of a Gaussian mixture comprising K components is given by

$$p(\xi) = \sum_{k=1}^{K} w_k \, \mathcal{N}(\xi; \mu_k, \Sigma_k) \,, \tag{1}$$

where w_k are the prior probabilities, also called the *weights*, and $\mathcal{N}(\xi; \mu_k, \Sigma_k)$ is the normal density of mean μ_k and covariance matrix Σ_k. The aim in background modeling is to estimate the parameters w_k, μ_k and Σ_k over time. The update of the parameters and the background modeling follow the technique proposed by Stauffer and Grimson in [13]. For background modeling, the K Gaussians are sorted by decreasing weight-to-standard-deviation ratio, w_k/σ_k. Then, the B first Gaussians for each pixel value (ξ) model the background with

$$B = \text{argmin}_{K_B} \left(\sum_{k=1}^{K_B} w_k \geq \lambda \right) , \tag{2}$$

where λ is a constant threshold determining the (*a priori*) proportion of background in the scene. The motion image (M) is composed of every pixel for which the Mahalanobis distance to each Gaussian component of the background model is greater than a given threshold. For further details on the procedure refer to [13]. Figure 1 displays an image and its corresponding motion mask.

Fig. 1. Background subtraction on a low definition image (128×160). *Left* is the original image; *right* is the motion image.

B. The Mean-shift Tracker

Mean-shift is a non-parametric density estimator that is derived from the Parzen-window. It uses an adaptive gradient ascent method to find modes in a probability density distribution. Mean-shift differs from vector quantization algorithms in that it finds the local modes of a distribution rather than minimizing the error function. In other words, the use of a kernel precludes the outliers such as false motion detection that can prevent convergence to the local maximum. As a result, mean-shift provides a more accurate localization of a target. A comprehensive introduction to mean-shift can be found in [2].

In this paper, mean-shift is applied to the motion mask in order to determine the position of a blob center. Let us denote the approximate position of the blob center $\hat{\mathbf{c}} = [c_x \; c_y]^T$, the set of motion pixel location $\mathbf{M} = \{\mathbf{m}_1, ..., \mathbf{m}_N\}$ and g a Gaussian isotropic kernel with bandwidth b. The new position of the blob center $\mathbf{c}$ is defined as

$$\mathbf{c} = \frac{\sum_{n=1}^{N} g\left(\left\| (\hat{\mathbf{c}} - \mathbf{m}_n)/b \right\|^2 \right) \mathbf{m}_n}{\sum_{n=1}^{N} g\left(\left\| (\hat{\mathbf{c}} - \mathbf{m}_n)/b \right\|^2 \right)} \,, \tag{3}$$

where N is the number of motion pixels. The mean-shift vector defining the shift in the center estimation is now defined as

$$\overrightarrow{p_{g,b}}(\hat{\mathbf{c}}) = \mathbf{c} - \hat{\mathbf{c}} \,. \tag{4}$$

The mean-shift vector $\overrightarrow{p_{g,b}}(\hat{\mathbf{c}})$ points toward the blob center. Recalling mean-shift is not a global mode finding procedure but a variable step-size gradient-ascent process, several iterations are required to locate a given center. Thus, Eq. (3) is iterated until $||\overrightarrow{p_{g,b}}(\hat{\mathbf{c}})|| < \gamma$ with $\hat{\mathbf{c}} \leftarrow \mathbf{c}$. The constant γ is arbitrarily set to 3 in our experiments. A lower value for γ does not improve the tracking accuracy in our experiments.

The convergence to the true blob center is ensured under two conditions: (1) The estimated center $\hat{\mathbf{c}}$ is initialized in the basin of attraction of the blob; (2) the bandwidth b of the kernel is adequate. The basin of attraction of a blob is defined as the set of locations for which the mean-shift converges to the blob center. Failing to initialize the mean-shift in the basin of attraction is the reason why mean-shift trackers diverge and lose track of the object. The initialization condition will be addressed in Section III. The match between the bandwidth of the kernel and the size of the blob is essential to ensure convergence. Indeed, a too large bandwidth would cause divergence in the presence of neighboring blobs. On the other hand, a too small bandwidth would lead to uncertainty in the blob location. In summary, mean-shift will efficiently and robustly track vehicles provided that the center is initialized in the basin of attraction and the bandwidth of the Kernel matches the size of the blob.

III. Projective Kalman filtering

We propose to estimate the kinematic variables, namely position and speed, and the size of vehicles through Kalman filtering to initialize the mean-shift tracker in the basin of attraction of its respective blob with adequate bandwidth. An accurate estimation of the observed position and scale of vehicle in the image will provide robust tracking by mean-shift. The general framework of Kalman filtering is set by the state-space equations:

$$\mathbf{X}_k = \mathbf{f}(\mathbf{X}_{k-1}, \mathbf{v}_{k-1}) \,, \tag{5}$$

$$\mathbf{Z}_k = \mathbf{h}(\mathbf{X}_k, \mathbf{w}_k)\,, \tag{6}$$

where $\mathbf{X}_k$ and $\mathbf{X}_{k-1}$ are the state vectors at times k and $k-1$, $\mathbf{f}$ and $\mathbf{h}$ are the process and observation functions, and $\mathbf{v}_{k-1}$ and $\mathbf{w}_k$ are the process and observation noises at times $k-1$ and k, respectively. The vector $\mathbf{Z}_k$ is the observation at time k. It is assumed that the probability density of the state $\mathbf{X}_k$, the process noise $\mathbf{v}_{k-1}$ and the measurement noise $\mathbf{w}_k$ are zero-mean Gaussians with respective covariance matrices $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{R}$. Here, the terminology is interpreted in the stricter sense: the process equation (5) models the physical process applying to the vehicle (Newton's laws), and the observation equation (6) models the observed trajectories after projection on the image plane. We propose to estimate the position and speed of the vehicle along the direction of the road (tangential direction) because the projection severely distorts the observations; as a consequence, the projection is highly non-linear. The projection on the normal direction, although non-linear, is much less severe and normal tracking is less critical because blobs are well delineated. The state vector is defined as $\mathbf{X} = (x; \dot{x}; s)^T$ where x and $\dot{x}$ are the position and speed of the vehicle following the tangential direction and s is the size of the vehicle.

The idea underlying the integration of the homographic transformation (projection on the CCD plane) in the Kalman filter is that it provides a better estimate of the state than a homographic transformation followed by Kalman filtering. Indeed, because of the non-linear nature of the transform, a slight change in the observation is the result of a large change in the state for distant objects. The projective Kalman filter is able to maintain an accurate estimate of the state whereas homographic transformation followed by Kalman filtering fails to capture such a change because the error due to the physical trajectory and the error due to the projection on the plane are not differentiated. In the proposed method, these two errors are modeled by two separate Gaussian processes $\mathbf{v}_{k-1}$ and $\mathbf{w}_k$, respectively.

The Extended Kalman Filter (EKF) is used to estimate the state because the projection is highly non-linear. The functions $\mathbf{f}$ and $\mathbf{h}$ are locally linearized by computing the respective Jacobian matrices $\hat{\mathbf{F}}$ and $\hat{\mathbf{H}}$. The EKF recursively estimates the state vector in two steps: prediction and update.

1) Prediction: The state vector $\hat{\mathbf{X}}_k^-$ and its covariance matrix $\hat{\mathbf{P}}_k^-$ are estimated with the value available at time $k-1$.

$$\hat{\mathbf{X}}_k^- = \mathbf{f}(\mathbf{X}_{k-1}, 0)\,, \tag{7}$$

$$\hat{\mathbf{P}}_k^- = \mathbf{Q_{k-1}} + \hat{\mathbf{F}}_{k-1}\mathbf{P}_{k-1}\hat{\mathbf{F}}_{k-1}^T\,. \tag{8}$$

2) Update: When a new measurement $\mathbf{Z}_k$ becomes available, i.e. when the mean-shift tracker has converged to the center of the blob, the state vector is updated as follows:

$$\mathbf{X}_k = \hat{\mathbf{X}}_k^- + \mathbf{K}_k\left[\mathbf{Z}_k - \mathbf{h}_k(\hat{\mathbf{X}}_k^-, 0)\right], \tag{9}$$

$$\mathbf{P}_k = \hat{\mathbf{P}}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T\,. \tag{10}$$

where

$$\mathbf{S}_k = \hat{\mathbf{H}}_k\hat{\mathbf{P}}_k^-\hat{\mathbf{H}}_k^T + \mathbf{R}_k \quad \text{and} \quad \mathbf{K}_k = \hat{\mathbf{P}}_k^-\hat{\mathbf{H}}_k^T\mathbf{S}_k^{-1}\,.$$

The Jacobians $\hat{\mathbf{F}}_{k-1}$ and $\hat{\mathbf{H}}_k$ are evaluated at $\mathbf{X}_{k-1}$ and $\hat{\mathbf{X}}_k^-$, respectively, with process and observation noise equal to 0.

A. EKF Process Equation

The process equation models the physical trajectory of the vehicle. Therefore, assuming that the vehicle speed varies slowly, the system equation $\mathbf{f}$ is written as

$$\mathbf{f}(\mathbf{X}_{k-1}, \mathbf{v}_{k-1}) = \begin{bmatrix} x_k \\ \dot{x}_k \\ s_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \dot{x}_{k-1}\Delta_t \\ \dot{x}_{k-1} \\ s_{k-1} \end{bmatrix} + \mathbf{v}_{k-1}. \tag{11}$$

B. EKF Observation Equation

The observation function projects the physical trajectory onto the CCD plane as shown in Fig. 2.

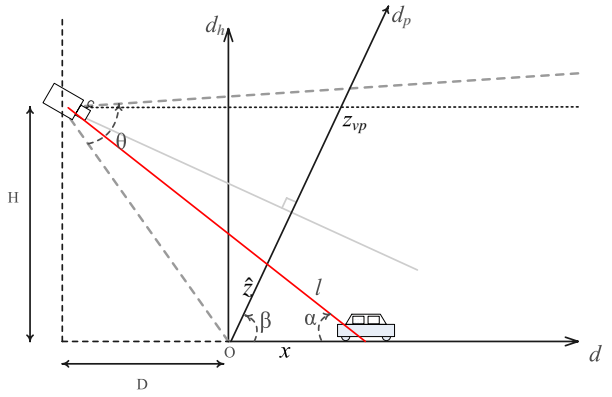

Fig. 2. Projection of the vehicle on a plane parallel to the CCD plane of the camera. The graph shows a cross section of the scene along the direction d (tangential to the road).

We propose to project the trajectories along the tangential direction d onto the d_p axis, knowing the following "easy-to-measure" parameters:

- Angle of view (θ);
- Height of the camera (H); and
- Ground distance (D) between the camera and the first location captured by the camera.

From Fig. 2, by Al-Kashi theorem, we have

$$\hat{z}^2 = x^2 + l^2 - 2cos(\alpha)xl \tag{12}$$

and

$$l^2 = \hat{z}^2 + x^2 - 2cos(\beta)x\hat{z} \tag{13}$$

where $\cos(\alpha) = \frac{D+x}{\sqrt{(H^2+(D+x)^2)}}$ and $\beta = \arctan(\frac{D}{H}) + \frac{\theta}{2}$. After squaring and substituting l^2 in (12):

$$\left(\cos\alpha x\sqrt{\hat{z}^2 + x^2 - 2\cos\beta x\hat{z}}\right)^2 = \left(x^2 - \cos\beta x\hat{z}\right)^2. \tag{14}$$

Grouping the terms in $\hat{z}$ to get a quadratic form leads to

$$\hat{z}^2x^2(\cos^2\alpha - \cos^2\beta) + 2\hat{z}x^3\cos\beta(1-\cos^2\alpha) + x^4(\cos^2\alpha - 1) = 0\,. \quad (15)$$

After discarding the non-physically acceptable solution, one gets

$$\hat{z}(x) = x\frac{(D+x)H\sin\beta - cos\beta H^2}{(D+x)^2\sin^2\beta - H^2\cos^2\beta}\,. \quad (16)$$

However, because $D \gg H$ and θ is small in practice, the angle β is approximately equal to $\pi/2$ and, consequently, Eq. (16) simplifies to $\hat{z} = \frac{xH}{D+x}$. Note that this result can be verified using Thales's theorem. Finally, we scale $\hat{z}$ with the position of the vanishing point Z_{vp} in the image to find the position of the vehicle in terms of pixel location[1], and denote the function h_x as:

$$z = h_x(x) = \hat{z}(x) \times \frac{Z_{vp}}{\lim\limits_{x\to\infty}\hat{z}(x)} = \hat{z}(x) \times \frac{Z_{vp}}{H}\,. \quad (17)$$

The observed speed of the vehicle $\dot{z}$ is defined as

$$\dot{z} = z_k - z_{k-1} = \frac{D\dot{x}_k}{(x_k+D)(x_k - \dot{x}_k + D)}\,, \quad (18)$$

and the observed size of the vehicle b (bandwidth of the kernel) and its associated function h_s is

$$b = h_s(s,x) = h_x(x_k + \frac{s}{2}) - h_x(x_k - \frac{s}{2}) = \frac{sD}{(x+D)^2 - (\frac{s}{2})^2}\,. \quad (19)$$

The observation function $\mathbf{h}$ is then decomposed as

$$\mathbf{h}(\mathbf{X}_k, \mathbf{w}_k) = \begin{bmatrix} z_k \\ \dot{z}_k \\ b_k \end{bmatrix} = \begin{bmatrix} \frac{x_k Z_{vp}}{x_k+D} \\ \frac{D\dot{x}_k}{(x_k+D)(x_k-\dot{x}_k+D)} \\ \frac{s_k D}{(x_k+D)^2-(s_k/2)^2} \end{bmatrix} + \mathbf{w}_k\,. \quad (20)$$

C. Vehicle Detection and EKF Initialization

The initialization of the variables is essential since the extended Kalman filter estimates the value of the state recursively. In vehicle tracking, the zone where the vehicle appears in the scene is known. Vehicles are detected using the motion mask, and the corresponding blobs are labeled using the connected component technique [6]. We assume here that the vehicle blobs in the detection zone are well delineated. This condition is met in almost every case since the apparent size of the vehicle is large in the detection zone. In the experiments, the rare cases where two vehicles were merged in the same blob is when the traffic is very dense and there is a continuous flow of vehicles. Most of the time, the dense flow of vehicles is correctly segmented. The center $\mathbf{c}$ of each blob is computed through mean-shift after labeling. The initial state vector value $\mathbf{X}_0$ is set as:

[1]The position of the vanishing point can be approximated either manually or automatically [12]. For the experiment purpose, we manually estimated the vanishing point.

$$\mathbf{X}_0 = \begin{pmatrix} h_x^{-1}(c_x) \\ \dot{x}_0 \\ s_0 \end{pmatrix}, \quad (21)$$

where c_x is the position of the object on d_p axis and h_x^{-1} is the inverse function of h_x. The values $\dot{x}_0$ and s_0 are set to the speed and the size of vehicles, respectively. We found that $\dot{x}_0 = 25$m/s and $s_0 = 5$m provide good results for the tested sequences. The state covariance matrix ($\mathbf{P}$) is set to 0 because the state is assumed known with certainty. The process noise and measurement covariance matrices, ($\mathbf{P}$) and ($\mathbf{Q}$) respectively, are initialized as follows:

$$\mathbf{Q} = \begin{pmatrix} 0.2 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.1 \end{pmatrix} \quad \mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (22)$$

D. State Backward Projection to Mean-shift Tracker

The estimation of the state in the space provides the ground distance x of the vehicle from the camera, its speed and its estimated size (which should be constant). However, the mean-shift tracker operates on the observed plane, i.e. the motion image. The state is thus back projected to the observation plane. As discussed in section II-B, the two essential parameters for mean-shift are the estimated position of the object and the bandwidth of the kernel. Mean-shift is run for each incoming frame. It thus needs to be initialized with the estimated position and bandwidth for the incoming frame at time $k+1$:

$$c_{x,k+1} = h_x(x_{k+1}) \approx h_x(x_k + \dot{x}_k\Delta_t)\,, \quad (23)$$

and

$$b_{k+1} = h_s(s_{k+1}, x_{k+1}) \approx h_s(s_k, x_k + \dot{x}_k\Delta_t)\,. \quad (24)$$

The component $c_{y,k+1}$ is initialized to its previous value (at time k) since it is not estimated by the Projective Kalman filter, i.e. $c_{y,k+1} = c_{y,k}$.

IV. Vehicle Tracking and Performance analysis

The proposed technique is tested on 15 traffic monitoring video sequences, each of which is approximately 6 minutes long and contains about 195 vehicles each. The video sequences are low-definition (128×160) to comply with the characteristics of traffic monitoring sequences. First, we compare the performances of the projective Kalman filter with the standard Extended Kalman filter for a rate of 30 frames per second (fps). Second, we compare the two Kalman filters for different frame rate, from 30fps down to 3fps. The second scenario provides a realistic evaluation of the algorithm performances for traffic monitoring where the frame rate is usually low.

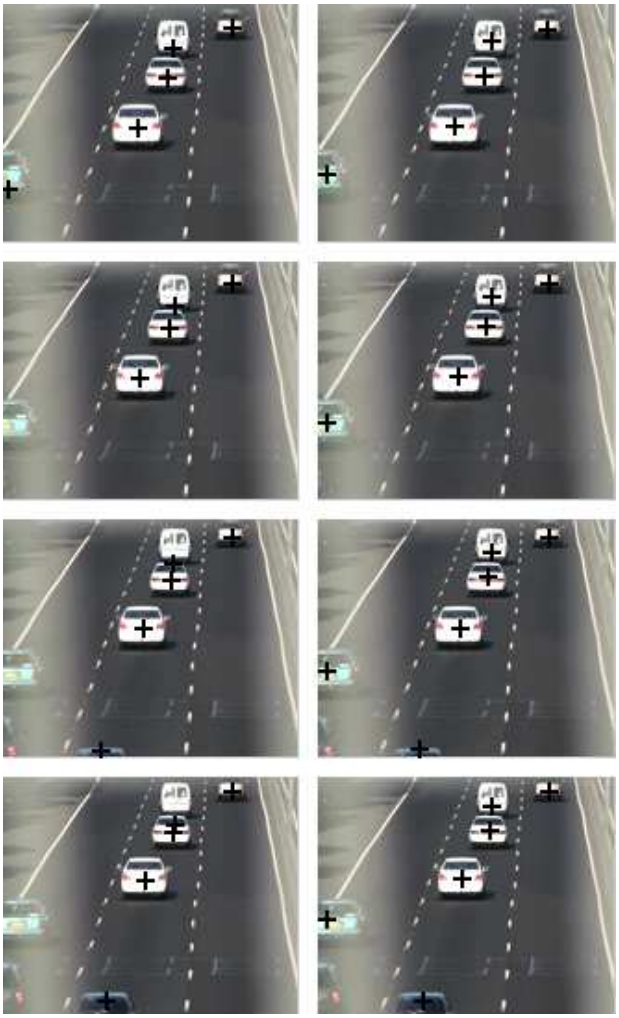

Fig. 3. Sequence of images illustrating the drift of a tracker on the top vehicle of the middle lane: the positions of each tracked object is indicated by a dark cross. The rows correspond to frames 4340, 4345, 4350, 4355 from top to bottom. *left* column is the standard Extended Kalman filter tracker; *right* column is the proposed tracker.

A. Projective Kalman Filter versus Standard Extended Kalman Filter

The standard Extended Kalman filter has been implemented in several traffic monitoring and analysis systems, see, e.g., [1], [11]. The standard Extended Kalman filter implements the same process function as in (11). However, the observation function is modeled with the identity matrix whereas the proposed projective Kalman filter uses the observation function described in (20). We propose here to estimate the robustness of the tracking by introducing a drift measure. Indeed, it is virtually impossible to evaluate the robustness of a tracker objectively; even comparing the ground truth with the output of a tracking algorithm is not satisfactory because it would not provide a framework to discriminate between errors generated by uncertainty and actual drifting of the tracker. We propose here to estimate the percentage of vehicles tracked without severe drift, i.e. for which the track is not lost. Since the vehicles are converging to the vanishing point, the trajectory of the vehicle along the tangential axis is monotonically decreasing. As a consequence, we propose to measure the number of steps where the vehicle position decreases (p_d) and the number of steps where the vehicle position increases or is constant (p_i), which is characteristic of a drift of the tracker. The percentage of vehicles tracked without severe drift is then calculated as

$$\text{Correct Tracking Rate} = \frac{p_d}{p_d + p_i}\,. \tag{25}$$

The average over 28,000 steps shows a percentage of correct tracking of 79.6% for the standard Extended Kalman filter and 96.4% for the projective Kalman filter. The proposed tracker shows more robust tracking, especially when vehicles are in the long distance. Fig. 3 is an example of a tracker that drifts. With the standard method (*left* column), the tracker on the top vehicle in the middle lane slowly drifts away from the vehicle tracked to the following one because the tracker is initialized on the edge of the two basins of attraction. After 15 frames, the tracker has changed basin of attraction and tracks the following vehicle. The proposed algorithm successfully tracks the vehicle throughout the sequence. In particular, the results for the proposed algorithm show a better ability to track long distance objects, prone to more projective noise than the standard algorithm.

B. Influence of Frame Rate on Tracking

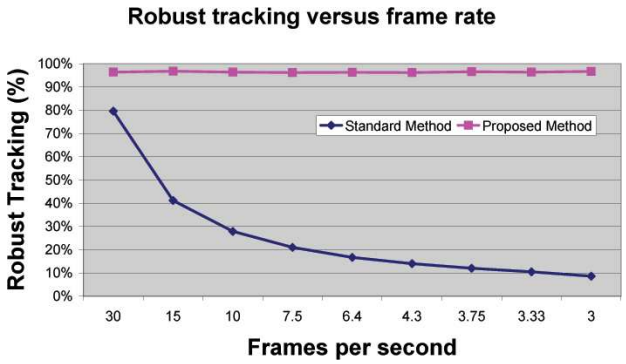

Fig. 4. Plot of Correct Tracking rate versus frame rate.

In this section, the two algorithms are evaluated for different frame rates. Aside from their low-definition, traffic monitoring video sequences present a very low frame rate due to the difficulty to transmit the video stream to the traffic agency. We propose here to evaluate the performances of mean-shift trackers initialized with the standard and the projective Kalman filter. We processed the videos sequences with decreasing frame rates, from 30fps to 3fps. The tracking robustness is evaluated according to the correct tracking rate measure in (25). The results are summarized in Table I and displayed in Fig 4. Whilst the rate of correct tracking decreases with the frame rate for the standard technique, it

TABLE I

CORRECT TRACKING RATE OF THE STANDARD AND PROJECTIVE KALMAN FILTER FOR DIFFERENT FRAME RATES.

Frame Rate	30fps	15fps	10fps	7.5fps	6fps	4.3fps	3.75fps	3.33fps	3fps
Standard EKF	79.6%	41.2%	27.9%	21.0%	16.7%	14.0%	12.0%	10.5%	8.6%
Projective Kalman Filter	96.4%	96.8%	96.4%	96.2%	96.3%	96.2%	96.6%	96.4%	96.7%

remains constant with the proposed method. Indeed, when the number of frames per second decreases, the displacement of the vehicle increases. As a consequence, the standard method is unable to robustly track the vehicles because the algorithm fails to initialize the mean-shift in the basin of attraction. Some examples are presented in Fig. 5. Tracking with the standard Extended Kalman filter fails for distant objects because the basin of attraction is small and the standard Extended Kalman filter does not provide a fine estimation of the position for the initialization of the tracker. The projective Kalman filter, on the other hand, provides accurate estimation of the vehicle position in the image integrating the decrease in the frame rate through adjustment of the vehicle speed; therefore, the proposed approach is insensitive to the frame rate.

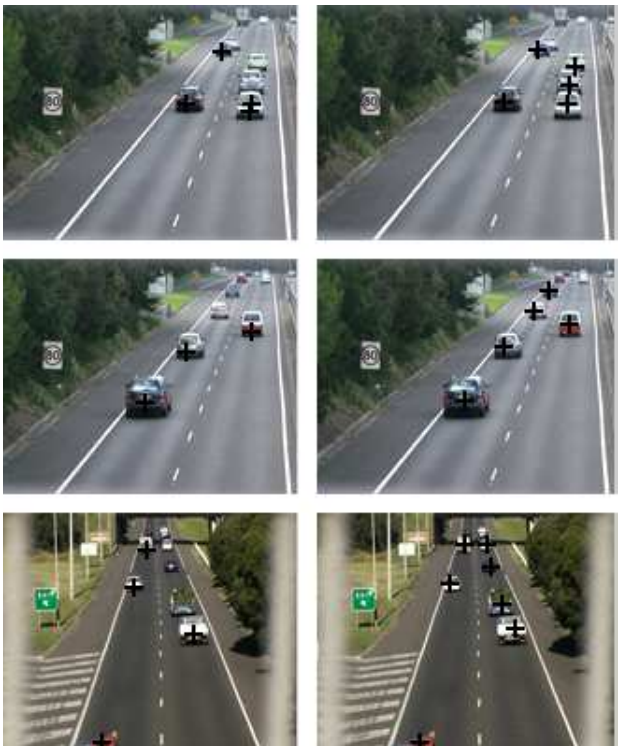

Fig. 5. Tracking robustness in case of low frame rate (3fps) for the standard (*left*) and the proposed method (*right*). With the standard method, the tracker drifts quickly and is unable to track the vehicle.

V. CONCLUSION

This paper proposed a tracking algorithm based on mean-shift and a projective Kalman filter. The algorithm achieves robust tracking due to the integration of the projection equation of the vehicle onto the image plane of the CCD camera. In particular, the observation function of the projective Kalman filter models the trajectory of vehicles with respect to their ground distance to the camera. The results showed that both the standard and the projective Kalman filter algorithms achieve robust tracking at a rate of 30fps, even though the projective Kalman filter performs better on long distance vehicles. However, the robustness of the standard EKF drops quickly with the frame rate whilst the robustness of the projective Kalman filter remains constant.

REFERENCES

[1] E. Bas, M. Tekalp, and F. S. Salman. Automatic vehicle counting from video for traffic flow analysis. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 392–397, 2007.

[2] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[3] J.-Y. Choi, K.-S. Sung, and Y.-K. Yang. Multiple vehicles detection and tracking based on scale-invariant feature transform. In *Proceedings of IEEE Intelligent Transportation Systems Conference*, pages 528–533, 2007.

[4] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[5] B. Gloyer, H. K. Aghajan, K-Y. Siu, and T. Kailath. Vehicle detection and tracking for freeway traffic monitoring. In *Proceedings of Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 970–974, 1994.

[6] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision*. Addison-Wesley Longman Publishing Co., Inc.

[7] Y.-K. Jung and Y.-S. Ho. Traffic parameter extraction using video-based vehicle tracking. In *Proceedings of IEEE Intelligent Transportation Systems Conference*, pages 764–769, 1999.

[8] D. Koller, J. Weber, and J. Malik. Towards realtime visual based tracking in cluttered traffic scenes. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 201–206, 1994.

[9] C.-P. Lin, J.-C. Tai, and K.-T. Song. Traffic monitoring based on real-time image tracking. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 2091–2096, 2003.

[10] J. Melo, A. Naftel, A. Bernardino, and J. Santos-Victor. Detection and classification of highway lanes using vehicle motion trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 7(2):188–200, 2006.

[11] Z. Qiu, D. An, D. Yao, D. Zhou, and B. Ran. An adaptive kalman predictor applied to tracking vehicles in the traffic monitoring system. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 230–235, 2005.

[12] D. Schreiber, B. Alefs, and M. Clabian. Single camera lane detection and tracking. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 302–307, 2005.

[13] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, 1999.

[14] Qi Zang and R. Klette. Robust background subtraction and maintenance. In *Proceedings of IEEE International Conference on Pattern Recognition*, volume 2, pages 90–93, 2004.