Detection and Classification of Highway Lanes Using Vehicle Motion Trajectories

José Melo, Andrew Naftel, *Member, IEEE*, Alexandre Bernardino, *Member, IEEE*, and José Santos-Victor, *Member, IEEE*

***Abstract*—Intelligent vision-based traffic surveillance systems are assuming an increasingly important role in highway monitoring and road management schemes. This paper describes a low-level object tracking system that produces accurate vehicle motion trajectories that can be further analyzed to detect lane centers and classify lane types. Accompanying techniques for indexing and retrieval of anomalous trajectories are also derived. The predictive trajectory merge-and-split algorithm is used to detect partial or complete occlusions during object motion and incorporates a Kalman filter that is used to perform vehicle tracking. The resulting motion trajectories are modeled using variable low-degree polynomials. A K-means clustering technique on the coefficient space can be used to obtain approximate lane centers. Estimation bias due to vehicle lane changes can be removed using robust estimation techniques based on Random Sample Consensus (RANSAC). Through the use of nonmetric distance functions and a simple directional indicator, highway lanes can be classified into one of the following categories: entry, exit, primary, or secondary. Experimental results are presented to show the real-time application of this approach to multiple views obtained by an uncalibrated pan–tilt–zoom traffic camera monitoring the junction of two busy intersecting highways.**

***Index Terms*—Lane detection, motion trajectory, scene interpretation, vehicle tracking.**

I. Introduction

RISING traffic levels and increasingly busier roads are a common feature across the globe. Consequently, there is an increasing requirement to develop intelligent traffic surveillance systems that can play an important role in highway monitoring and road management systems. Their purpose, among other things, is to provide real-time statistical data on traffic activity and to signal potentially anomalous situations, e.g., accident detection or dangerous driving.

This paper addresses the problem of lane detection and classification through an analysis of vehicle trajectories using standard installation traffic surveillance cameras. These are operator-controlled nonstationary cameras and usually of the pan–tilt–zoom (PTZ) variety, which allows for multiple views. They are capable of covering larger road scenes compared to static cameras. Lane detection and classification is an important first step in building a semantic scene description of the highway. This can lead to generation of statistical data on traffic activity, such as vehicle density, lane changes, and detection of anomalous situations (e.g., accidents, congestion, and dangerous driving).

We demonstrate that accurate vehicle trajectories can be obtained by object segmentation, tracking, and screening of partial and complete occlusions. An occlusion reasoning approach is used to detect and count the number of overlapped objects present in a segmented region. Trajectory points are then classified according to whether they are generated by a single or overlapped object. Trajectory paths are then represented using low-order polynomials for dimensionality reduction.

Cluster analysis can be performed on the coefficient space to build a self-consistent aggregation of many individual trajectories. By taking into account vehicle lane changes, lane geometry can be estimated from uncalibrated but stable video sequences. The use of nonmetric distance functions permits a classification of lane systems based on simple heuristics into different categories, e.g., *entry*, *exit*, *primary*, or *secondary*.

The method that we present here is independent of camera viewpoint and requires no special *a priori* calibration of the image sequences. Although lane width is not constant across the image because of perspective effects, all that the method requires is a crude estimate of the "average" lane width in pixel coordinates. The ability to overcome the need for camera calibration is especially advantageous, as it permits the use of standard traffic camera installations without the burden of performing tedious calibration procedures.

The remainder of the paper is organized as follows. We review some related work on vehicle tracking and analysis of motion trajectories in Section II. Section III describes our algorithm for generating vehicle trajectories with attention paid to occlusion handling. Techniques for cluster analysis and trajectory representation are described in Section IV. These are applied to a description of lane geometry and lane categorization. An experimental evaluation of the proposed techniques is presented in Section V, and the paper concludes with a discussion and summary in Section VI.

II. Related Work

The starting point for previous works in motion tracking and analysis is the segmentation of moving objects based on

Manuscript received February 2, 2005; revised July 21, 2005. This work was supported in part by the Portuguese Agency for Innovation (ADI) under Project Monitorização Automática de Fluxo de Trânsito Automóvel e Detecção Acidentes e Avarias em Auto-Estradas (ADI-INTELTRAF) and by the U.K. Department of Trade and Industry through a Knowledge Transfer Partnership (KTP) Programme Award under Grant T-4231. The Associate Editor for this paper was A. Eskandarian.

J. Melo and A. Naftel are with the School of Informatics, University of Manchester, Manchester M60 1QD, U.K. (e-mail: jpqm@isr.ist.utl.pt; a.naftel@manchester.ac.uk).

A. Bernardino and J. Santos-Victor are with the Instituto Superior Técnico, Lisbon 1049-001, Portugal (e-mail: alex@isr.ist.utl.pt; jasv@isr.ist.utl.pt).

Digital Object Identifier 10.1109/TITS.2006.874706

background subtraction techniques [1], [2]. A recent survey of different environment modeling approaches applicable to visual surveillance applications can be found in [3].

Typically, each pixel is modeled using a Gaussian distribution built up over a sequence of individual frames, and segmentation is then performed using an image differencing strategy. It is particularly important to employ a background model update strategy in uncontrolled outdoor environments. Shadow detection and elimination strategies have been commonly employed to remove extraneous segmented features [4]–[7].

It is also important to handle partial and complete occlusions in the video data stream [8]–[10]. Occlusion detection can be performed using an extended Kalman filter (KF) that predicts the position and the size of object bounding regions. Any discrepancy between the predicted and measured areas can be used to classify the type and extent of an occlusion [9], [10].

Applications of vision-based surveillance techniques to higher level traffic analysis systems have also been developed specifically for accident detection at road intersections [10], [11], estimation of traffic speed [12], [13], and accident prediction [14]. More general techniques for object path detection, trajectory classification, and indexing have also been proposed [15]–[17].

In [12], an algorithm to estimate mean traffic speed using uncalibrated cameras is presented. It employs geometric constraints in the image, interframe vehicle motion, and distribution of vehicle lengths. Although vehicle speed can be determined using operator-controlled PTZ cameras, the algorithm only works on straight roads and does not handle vehicle lane changes. Traffic flow histograms and calculation of the image vanishing points are used in [13] to measure mean speed but with limitations similar to those in the previous approach.

The importance of analyzing object trajectories for the information that they convey on motion understanding is increasingly realized. Recent contributions to motion data mining and clustering have been made [18]–[21], although this work is not particularly directed at understanding traffic flow, which is typically highly constrained by road geometry.

The contribution of our paper is stated as follows. We attempt to achieve a higher level road description than that presented in [12] and [13] through processing of vehicle trajectories from uncalibrated video sequences. This paper is partly inspired by [9] in which vehicle trajectories are approximated by cubic polynomials using least squares (LS). However, it is shown that the effects of vehicle lane changes and outliers in the tracking process can be minimized using the RANSAC estimator [22]. However, we choose to cluster these trajectories to build a model of the lane geometry using nonmetric distance functions. The only *a priori* information needed is an approximate estimate of the average lane width, in image coordinates, to disambiguate the zoom level used.

Trajectory clustering and classification has been previously applied in [23] to provide a natural language description of vehicle activity in a scene, e.g., vehicle turns left at junction with low speed. We are looking for higher level semantics that can be used to describe overall highway lane geometry and traffic behavior.

III. Object Tracking With the Predictive Trajectory Merge-and-Split (PTMS) Algorithm

The proposed system uses a multistage approach to determine the vehicle motion trajectories and eventually the lane geometry. An overview of the system is shown in Fig. 1. First, we build a background model to segment foreground objects. Then, the PTMS algorithm is used to achieve two goals, namely 1) predicting the position of the detected vehicles, by means of a KF [24] and 2) performing a time-consistent analysis (grouping) of the detected blobs, possibly merging and splitting detected blobs due to partial or complete occlusions. This process permits the identification of blobs composed of multiple vehicles that should not be used as input to the trajectory clustering algorithm, because the position estimation may be unreliable.

A. Vehicle Detection (Segmentation)

The initial vehicle detection stage is based on the adaptive smoothness method [1] to build a background model. This model is updated continuously, at every time instant, based on those pixels that were not detected as moving regions. This approach assumes that the camera is static, whereas in outdoor scenarios, cameras are frequently subject to small motion disturbances (e.g., due to wind). In this case, more sophisticated background modeling techniques can be used, e.g., employing multiple background models instead of only one.

Moving objects are extracted through background differencing. Detected blobs having an area smaller than a certain predefined minimum number of connected pixels $(K_{\min})$ are deemed to be noise and disregarded. Erode and dilate morphological operations are used to eliminate small holes within blobs. Although shadow removal is not incorporated in the detection process, the background update module uses a double thresholding operation to attenuate self-shadowing.

B. Steady-State KF

If we wish to build complete motion histories for each tracked object, it is necessary to filter the position estimates over time and solve tracking instabilities caused by near and partial occlusions, shadows, and image noise. In the case of multiple simultaneous object tracking, if we lose track of one vehicle and another vehicle is suddenly detected nearby, there is an obvious danger of mistaken vehicle identification.

The KF was used to estimate the target trajectory. The KF is based on a dynamical model of the system that seeks to explain how the system state and observations change with time, and how they are affected by noise. Let $x_{(k)}$, $v_{(k)}$, and $a_{(k)}$ be the position, velocity, and acceleration of the blob centroid at time k along one of the image directions. Even when a vehicle moves at constant velocity, its image velocity will vary over time due to camera perspective effects. To account for such velocity variation, we model the vehicle image dynamics as a constant acceleration dynamic system. Both the vehicle maneuvers (i.e., changes in velocity) and the effect of perspective will be modeled as a (local) acceleration term and an external

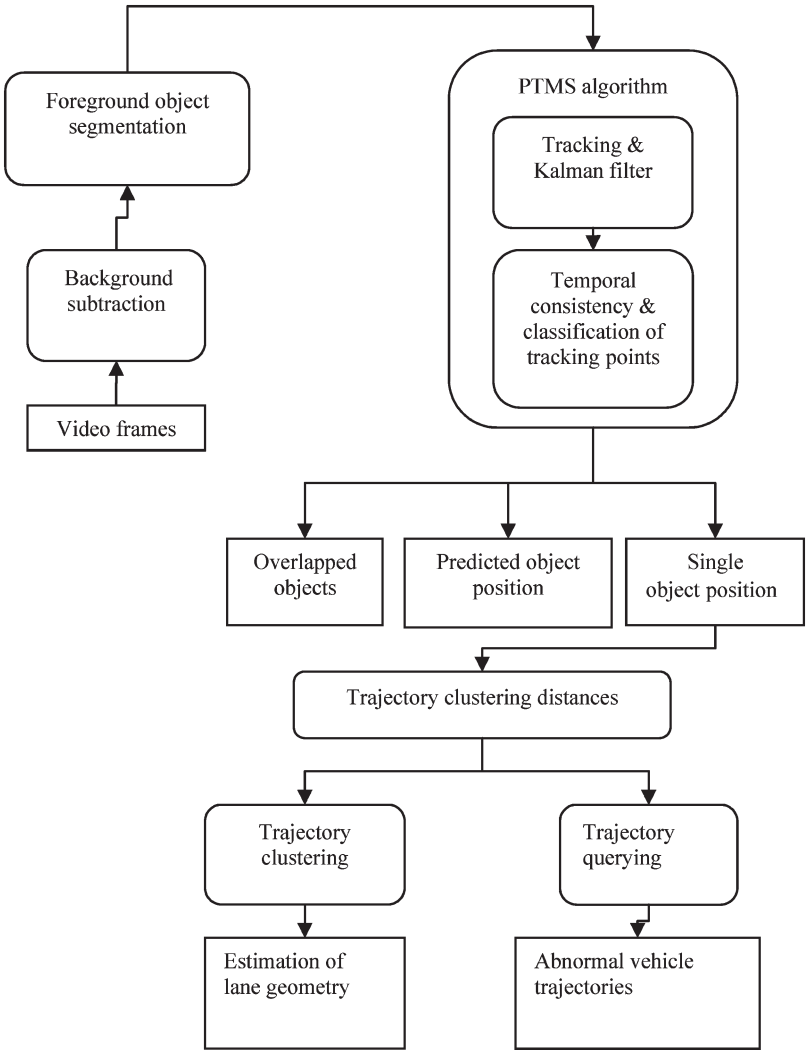


Fig. 1. Overview of the proposed vehicle tracking and lane classification system.

noise source. We further assume that the tracker provides noisy measurements $Y_{(k)}$ of the vehicle image position. Denoting the system state vector by $S_{(k)}$ and combining the position, velocity, and acceleration, the system dynamics and observations can then be described as follows:

$$S_{(k)} = \mathbf{\Phi} S_{(k-1)} + \xi \qquad Y_{(k)} = C^T S_{(k)} + \eta$$

with

$$S_{(k)} = \begin{bmatrix} x_{(k)} & v_{(k)} & a_{(k)} \end{bmatrix}^T$$
$$\mathbf{\Phi} = \begin{bmatrix} 1 & \tau & \tau^2/2 \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix}$$
$$C^T = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{1}$$

where ξ and η are independent zero-mean Gaussian-distributed random vectors with covariances $\mathbf{Q}$ and $\mathbf{R}$, matrix $\mathbf{\Phi}$ is the transition matrix, vector C indicates that the system observations are the position estimates, and τ is the system sampling period.

The KF has a prediction and a filtering step. In the prediction step, the problem consists of determining the estimate of the state vector at time $k+1$, given only the information available at time k, $S_{(k+1|k)}$. Then, once a new measurement is available at time $k+1$, the filtering step plays the role of updating the current estimate to reflect the new information contained in the new measurement $S_{(k+1|k+1)}$ as follows:

Prediction :

$$S_{(k+1|k)} = \mathbf{\Phi} S_{(k|k)}$$

Update/filtering :

$$S_{(k+1|k+1)} = S_{(k+1|k)} + K_k \left(Y_{(k+1)} - C^T S_{(k+1|k)} \right). \tag{2}$$

In the standard formulation of the KF, the so-called Kalman gain K_k is a time-varying gain that depends on the noise covariance matrices Q and R. This would require computing the covariance matrices of the tracking error and solving a discrete Riccatti equation at each frame. Instead, we have adopted to use the steady-state version of this filter, which is obtained by computing the Kalman gain over time and using the values of the gains reached for the steady state. The computation is much simpler, and the system becomes time invariant, thus

simplifying the analysis. Still, the steady-state version of the constant acceleration filter, which is often called the $\alpha - \beta - \gamma$ filter, requires the evaluation of (constant matrices) $\mathbf{R}$ and $\mathbf{Q}$. Instead, as proposed in [24], we computed the Kalman gain as a function of one single parameter—the *maneuverability* index—which quantifies the noise ratio between the system state equation and the observations. Following [24], we have

$$K_k = K = [\,\alpha \quad \beta/\tau \quad \gamma/\tau^2\,]^T$$

with

$$\alpha = 1 - \theta^3 \quad \beta = 1.5(1-\theta^2)(1-\theta) \quad \gamma = (1-\theta)^3 \qquad (3)$$

where the parameter $0 < \theta < 1$ is the maneuverability index, which can be adjusted so that the filter's transient response is critically damped [24]. The analysis in the second image direction y can be done in the same way or both x and y components integrated into a single six-dimensional state vector and a two-dimensional observation vector. When an occlusion is detected, the KF is not updated with the new value of the target position, $x_{(k)}$.

In the following section, we will see how to maintain the temporal consistency (data association) of the tracked blobs, e.g., in detecting the vehicle occlusions.

C. Temporal Consistency and Classification of Tracking Points With Heuristic Merge-and-Split Rules

The presence of shadows or "near" occlusions caused by traffic congestion can seriously degrade the accuracy of blob detection. Typically, several vehicles may be misdetected as one single vehicle with consequent problems for generating an object trajectory. Approaches based on spatial reasoning use more complex object representations such as templates or trained shape models. However, this is dependent on image resolution and only works under partial occlusion. A better approach is to use a temporal smoothness constraint in checking vehicle positions under different types of occlusion. Here, we propose a set of temporal heuristics that can easily complement a spatial approach.

The algorithm works as follows: First, we define a blob as a connected region resulting from the background subtraction process. Then, use (2) to predict the most likely position of the blob in the next frame. In addition to its size and position, a blob is characterized by its *number of components* (i.e., grouped vehicles). At the beginning, the number of components of each blob is initialized to one, and this increases each time a merge situation is detected. The algorithm description is stated as follows:

Temporal Grouping Algorithm

For each frame and for every blob, do the following.

1) Determine whether there is a one-to-one correspondence between the blobs in consecutive frames by comparing their sizes and positions. This condition is met when the tracking system succeeds in keeping individual tracks for each vehicle. The positions and sizes are updated over time.
2) If the blob size has increased above a percentage threshold Ω (typically of the order of 25%), this may correspond to a *merge* situation. If the previous image contains multiple blobs in the area under analysis, the number of components of the new blob is increased by the number of blobs found in that area.
3) If the blob size has decreased below a certain percentage threshold Ω between two frames, this may be due to a possible split. Determine whether any new blobs appeared in the vicinity of the predicted blob position, provided that the original blob consisted of a number of components greater than 1. If so, decrease the number of components in the new blobs, originating from the split of the blob found in the previous image.
4) Check if there are any new blobs in the next image.

The algorithm works reasonably well for most of the time. The principal drawback is when the initial blob is composed of several objects. In this case, it will be misdetected as a single object. To tackle this problem, a spatial grouping algorithm could be applied to the initial blobs to determine whether they are composed of one or more objects. The results of applying the PTMS algorithm are presented in Section V.

IV. Lane Detection Using Clustering of Vehicle Trajectories

In a previous work, lane detection has been undertaken using feature extraction on static images, e.g., by use of Hough transform [13]. Inasmuch as this might be feasible using stationary cameras in conditions of good visibility and low traffic density, it is difficult to achieve in practice when using uncalibrated traffic cameras and image quality is low. In highly constrained environments such as highways, it is tempting to use vehicle trajectories rather than image analysis of static scenes to detect lanes. The trajectory-based approach has a number of advantages.

1) It enables use of operator-controlled PTZ cameras rather than calibrated static cameras.
2) Techniques based on object trajectories are more robust to scale and viewpoint transformations.
3) Motion data are more robust than static scene analysis with respect to light variation and sensor noise.
4) Although tracking in cluttered scenes is quite a challenging problem, one can still gather sufficient statistical evidence over a large number of frames to be able to compute reliable information about motion trajectories. This is because we can afford to discard ambiguous data and keep only very reliable (trajectory) estimates for the clustering step.

Using motion trajectories does not require *a priori* knowledge of the number of lanes or road geometry, e.g., whether a particular section of highway is straight or curved. Due to image perspective, the lane width will change with image position. The method assumes that the "average" lane width t_w in image coordinates is known in advance to disambiguate the overall image scale or camera zoom level. This information will allow

us to discard ambiguous trajectory estimates (e.g., those close to the vanishing point). The parameter t_w can be easily estimated by an operator and does not need to be extremely precise.

A. Trajectory Modeling

We choose to model the vehicle motion path with a low-degree polynomial. The output of the PTMS algorithm is a set of tracked points (x_{ij}, y_{ij}), $1 \leq i \leq n_j$ for each vehicle trajectory S_j, $1 \leq j \leq N$. The trajectory can be approximated by a polynomial $P_m(x)$ of degree $m < n_j$ as follows:

$$y \approx P_m(x) = a_0 + a_1 x + \cdots + a_m x^m. \tag{4}$$

The unknown $m+1$ coefficients $\{a_i\}$ can be determined using LS by minimizing the function E with respect to $a_0, a_1, \ldots$ as follows:

$$E(a_0, a_1, \ldots, a_m) = \sum_{i=1}^{n_j} \left[y_i - (a_0 + a_1 x + \ldots + a_m x_i^m) \right]^2. \tag{5}$$

By preprocessing the trajectory point set $\{S_j\}$, we can discover the main direction of traffic flow and determine which independent coordinate increases/decreases monotonically. This will lead to a choice of either $P_m(x)$ or $P_m(y)$ as trajectory models. This process can be performed by aggregating the slopes from the start and end points of each trajectory. A similar result can be obtained using principal component analysis on the tracked points. This is also used to determine the direction of the traffic nearest to the camera. In more general situations, one can resort to higher level curve parameterizations, such as piecewise-defined polynomials or B-splines [28]. The choice of parameterization does not play a critical role (apart from singularities) in the remaining processing modules. The simple parameterization we have chosen has proved sufficient for the scenarios where we conducted tests.

Preprocessing is also used to remove obvious data inconsistencies caused by tracking errors, sensor noise or unstable camera motion, e.g., high winds. We discard short or partial trajectories and those whose point separations are greater than some predetermined threshold.

We then fit a LS polynomial of degree m for each trajectory S_j in the chosen coordinate direction. Starting from $m = 1$, we use the average fitting error to ascertain the optimal value of m. If the error is greater than some predetermined threshold, m is increased by 1, and the trajectory is refitted. For all the highway scenes tested, we have found that low-degree polynomials up to order 3 were sufficient to describe the lane curvature. Assuming that for short sections of highway, vehicle lane changes can constitute any percentage of the overall trajectories, a RANSAC estimator [22] was used in conjunction with LS to eliminate outliers such as lane changes or tracking errors. These would otherwise bias the results for the detection of lane centers. RANSAC is robust to outlier trajectories produced by frequent vehicle lane changes, undetected overlapped vehicles and noise in the video sequence. The use of robust estimators is critical in achieving the improvements in overall system performance.

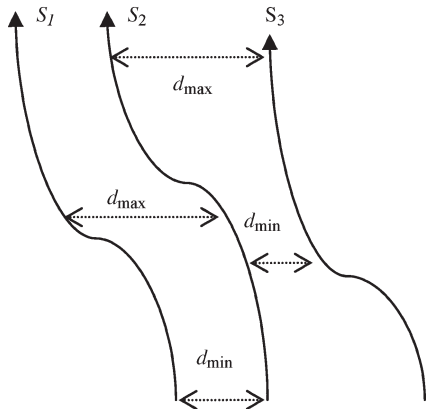


Fig. 2. Interpretation of the distance functions $d_{\max}$ and $d_{\min}$ between trajectory approximations of S_1, S_2, and S_3.

B. Modified K-Means Algorithm for Trajectory Clustering

A robust K-means clustering algorithm that works in the polynomial coefficient space for improved speed and efficiency is now described. An upper bound is set on the maximum number of trajectories that can belong to each cluster. The cluster centers correspond approximately to detected lane centers.

We propose a set of nonmetric distance functions to be used for cluster analysis of trajectories. Nonmetric distance functions $d(P, Q)$ satisfy the usual conditions of nonnegativity $d(P, Q) \geq 0$, identity $d(P, Q) = 0$ iff $P = Q$, and symmetry $d(P, Q) = d(Q, P)$ but fail to satisfy the triangle inequality $d(P, Q) \leq d(P, R) + d(R, Q)$. However, visual observations of similarity judgements can often be shown to violate metric axioms [25], and there is a good case for relaxing the triangle inequality constraint in certain application domains. Given two trajectories Q and S with LS polynomial approximations P_Q, P_S, we can define the following maximum, minimum, and root mean square (rms) distance functions:

$$\begin{aligned} d_{\max}(P_Q, P_S) &= \arg\max_{x \in X} |P_Q(x) - P_S(x)| \\ d_{\min}(P_Q, P_S) &= \arg\min_{x \in X} |P_Q(x) - P_S(x)| \\ d_{\mathrm{rms}}(P_Q, P_S) &= \sqrt{\frac{1}{x_2 - x_1} \int_{x_1}^{x_2} \left(P_Q(x) - P_S(x) \right)^2 dx} \end{aligned} \tag{6}$$

where X is the interval over which distance is calculated. Here, it is assumed that the x-coordinate is monotonic. A geometric interpretation of $d_{\max}$ and $d_{\min}$ is given in Fig. 2. The distance d_{rms} in (6) represents a quantity related to the area between P_Q and P_S, normalized with respect to the length of the interval of integration, and can be evaluated as a closed-form expression, e.g., in [26], for $m = 3$. The distance functions $d_{\max}$ and $d_{\min}$ can be evaluated through a line search technique. Note that we always have $d_{\min}(P_Q, P_S) \leq d_{\mathrm{rms}}(P_Q, P_S) \leq d_{\max}(P_Q, P_S)$. In a previous work, the cluster distances were calculated using the Hausdorff distance [23]. Although this works with arbitrary trajectory point sets, it is expensive to compute as it requires $0(n_1 n_2)$ operations, where n_1 and n_2 denote the number of points in sets Q and S. It is also sensitive to outlier points

that can dominate the distance calculation. However, it is a useful baseline calculation for comparison purposes, as we shall later see.

K-means [27] is a very simple and effective clustering algorithm, which requires specifying the number of sought clusters in advance (parameter K). First, K points are chosen at random as initial cluster centers. Data measurements are assigned to their closest cluster center according to the distance measured by a suitable metric. In the following step, the mean of all data points in each cluster is calculated. These means are taken as the new centroids for the various clusters. Finally, the whole process is repeated with the new cluster centers. This iterative process continues until the same points are assigned to each cluster in consecutive rounds. At this point, the clusters centers have stabilized and will remain the same thereafter. This process converges to a local minimum only, as different final cluster centers can arise by choosing a different initialization.

The modified K-means algorithm proposed here will determine the number of clusters automatically and has the following overall structure. The initial clusters are built from a set of trajectories $\{S_j\}$ by searching for those S_k that maximize $d_{\min}(S_i, S_j)$. The clustering loop then iterates by assigning trajectories S_j to those clusters that minimize the $d_{\max}(S_j, C_k)$, where C_k denotes the cluster centroids. The distance $d_{\max}$ in (6) is used as an auxiliary distance to sort S_js within each cluster. The cluster points are represented by the $(m+1)$-dimensional vector of polynomial coefficients, where m is the degree of the polynomial. The following steps are used to build an initial set of trajectory clusters:

Cluster Initialization Algorithm

1) Create a reference trajectory S_R at the edge of the image parallel to the main orientation of traffic flow.
2) Select a second trajectory, S_j, which maximizes $d_{\mathrm{rms}}(S_j, S_R)$, provided that it exceeds some predetermined threshold and subject to the constraint $\#\{S_j\} > t_{\min A}$, where $\#\{S_j\}$ denotes the number of points in the trajectory. A sensible choice of threshold on d_{rms} is the "average" lane width t_w.
3) Select the next S_k that maximizes distance $d_{\min}(S_j, S_k)$ to all other trajectories present in the set, provided that $d_{\min}(S_j, S_k) > t_w$ and $\#\{S_k\} > t_{\min A}$.
4) Repeat step 3) until no further trajectories can be added.

Trajectory Clustering Algorithm

1) Compute the distances between each trajectory and cluster centers $d_{\max}(S_j, C_k)$.
2) Associate trajectories S_j to the nearest cluster using metric $d_{\max}(S_j, C_k)$, provided that $d_{\max}(S_j, C_k) < t_w$ and $\#\{S_j\} > t_{\min B}$. These constraints ensure that short and partial trajectories are excluded.
3) The cluster means (i.e., polynomial estimate representative of the cluster) are updated after five trajectories have been associated to each cluster, using RANSAC to eliminate outliers.
4) The process stops when 20 trajectories have been added to each cluster.

For the cluster initialization, we try to choose the longest trajectories, and for the trajectory clustering, we try to exclude very short trajectories, such that $t_{\min \mathrm{A}} > t_{\min \mathrm{B}}$.

It is found that use of $d_{\max}(S_j, C_k)$ discards most trajectories representing lane changes. RANSAC then removes the few remaining outliers erroneously added to the cluster. Once the cluster means C_k have become stable, we regard these as the lane centers. The distance functions (6) can now be reused for the purpose of categorizing lanes, as we shall see in the next section.

C. Lane Classification

Quite often, highway traffic monitoring is done with PTZ cameras, which raises the issue of determining which direction the camera is looking at. Here, we achieve this objective by using additional contextual information such as the number of lanes in a certain highway or the number of roads crossing at the observed region. The camera orientation is established by calculating a linear approximation to the main direction of traffic flow. This gives the approximate angle between the main highway and the direction in which the camera is pointing. The main orientation of traffic flow is classified as *right* or *left*. Each trajectory has a direction that is classified as *positive* or *negative* by inspecting the dominant $(\Delta x, \Delta y)$ increments. If positive values of Δx predominate, the direction is classified as positive, otherwise, it is negative.

The cluster centers representing lanes are classified into one of the following categories: primary, entry, exit, or secondary. For the sake of simplicity, we assume that every highway has at least four primary lanes, two in each direction of flow. In abnormal situations such as temporary roadworks and contraflow, this assumption may be violated. The four primary lanes in the center of the highway are found by searching for neighboring lanes having opposite directional flow. The distance measures are initially applied to the first two lanes having the same direction.

We use l_i to index a particular lane and L to represent the set of all lanes found in the scene. The lane classification scheme can be expressed by the following set of rules:

$$\begin{aligned} &\text{For all } l \in L, \\ &\quad \text{"primary,"} && \text{if } d_{\max}(l_i, l) \le \delta \\ &\quad \text{"secondary,"} && \text{else if } d_{\min}(l_i, l) > \delta \\ &\quad \text{"entry} \vee \text{exit,"} && \text{otherwise.} \end{aligned}$$

The choice of entry or exit is determined by the angle between the analyzed lane and the previously classified lane, taking into consideration whether the direction of traffic flow is positive or negative.

D. Traffic Flow Analysis, Trajectory Indexing, and Retrieval

Having parameterized the trajectories and defined the clusters and distance metrics between trajectories, we can index the trajectories according to their polynomial coefficients. In this way, we can pose queries about trajectories of interest by specifying a reference query Q and performing a similarity

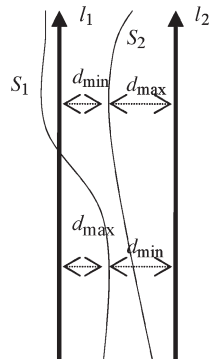


Fig. 3. Interpreting the categorization of vehicle lane changes. S_2 will be classified as a lane change and S_1 represents a vehicle remaining in l_1.

search over $\{S_j\}$ using one of the distance functions $d(Q, S_j)$. The distances are then sorted, and the k-nearest neighbors can be easily determined.

We will describe an example of querying the number of trajectories associated with a given lane. Obviously, this query should exclude trajectories of overlapping vehicles, which are discarded in the tracking phase. Using a previously classified lane as a query trajectory, we build a set Ω containing all trajectories that are closer to the query lane (using $d_{\max}$) than to all the neighboring lanes. Note that all lane changes are discarded, because the maximum distance to the query lane is greater than the minimum distance to other lanes. This situation is shown in Fig. 3. Here, S_1 will be categorized as a vehicle associated to lane l_1, whereas S_2 represents a lane change trajectory. The set Ω_l contains only those vehicle trajectories travelling in lane l only. In the following, L is the set of lanes and S the set of trajectories. Then, for a given query trajectory l, the set Ω_l is formed as follows:

$$\Omega_l = \{S_j \in \mathrm{S} : d_{\max}(l, S_j) < d_{\min}(l_k, S_j)\} \quad \text{for all } l_k \in L \text{ and } l_k \neq l.$$

In the presence of strong perspective and very irregular trajectories, this condition may be too conservative. Consequently, some trajectories that do not correspond to lane changes may be discarded. Although such cases are normally quite rare, a less conservative test can be designed by replacing the use of $d_{\min}(l_k, S_j)$ with the distance between l_k and the point in S_j that is most distant to lane l. Under normal circumstances, the set of vehicle trajectories representing lane changes can be inferred from the aforementioned scheme by the principle of mutual exclusivity. In other words, the set Ω_l contains trajectories within the lane l, whereas its complementary set Ω_l^c can be used to determine those trajectories corresponding to lane crossings. As another example of the type of query that we can perform with this methodology, we might wish to retrieve all vehicle trajectories that stray into the emergency stopping lane.

V. Experiments

The algorithm was tested on two different highway surveillance sequences. The first was recorded with a stationary camera having a resolution of 176 × 144 pixels and a video capture rate of 10 frames/s. The second sequence was taken using a higher resolution camera having 352 × 288 pixels at 25 frames/s. The camera was positioned 15 m above the highway. Unfortunately, the recording of the second sequence was made on a very windy day, and unstable camera motion often resulted in poor tracking results and fragmented trajectories. Nevertheless, this presented us with a challenging data set. In the first sequence, the tracking module generates quite satisfactory results. The tracking can be improved in the future by considering a more sophisticated background model that may include several images, thus accounting for variations due to camera motion disturbances, foliage, etc.

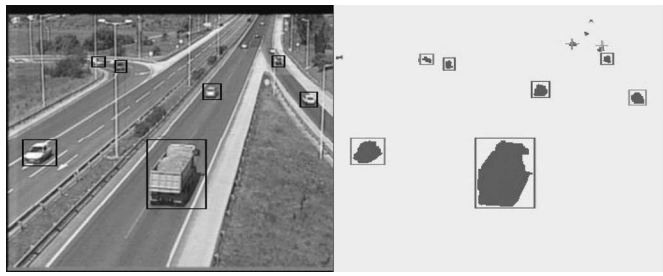


Fig. 4. Left: Sample of video sequence with detected moving objects. Right: Vehicles tracked using the Kalman filter and segmented objects. A cross is placed in the predicted vehicle position when tracking is lost.

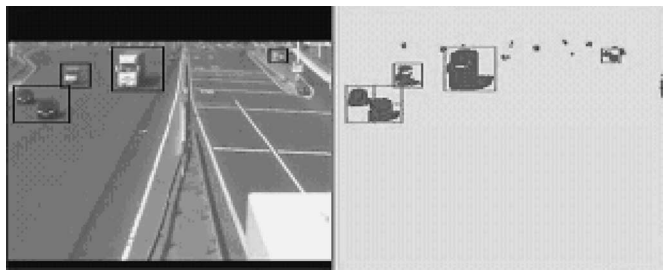


Fig. 5. Tracking and occlusion handling. The two cars on the left are detected as a single overlapping blob. A cross is placed in the segmented region's bounding box to indicate that it corresponded to a single region in the previous frame.

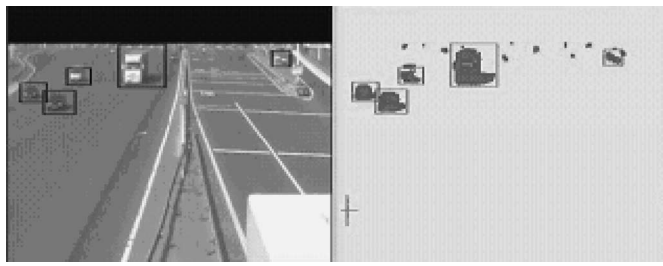


Fig. 6. Tracking result from the previous frame in sequence. Note that the two cars on the left are detected as separate objects in this frame.

A. Output of Tracking and PTMS Algorithm

In Figs. 4–6, we show the results of applying background subtraction and foreground object segmentation using grayscale video sequences. Fig. 4 shows a sample of the foreground detected moving objects.

Color-coded bounding boxes can be used to distinguish segmented objects from successfully tracked vehicles. When

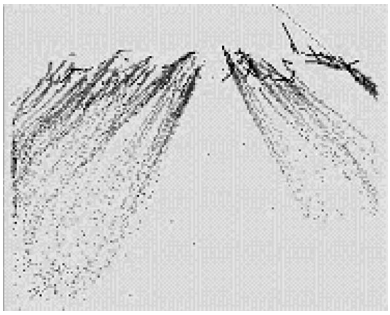
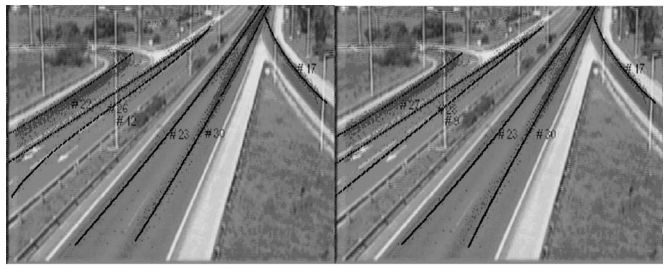


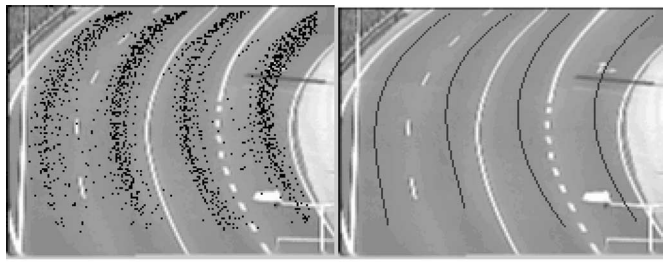
Fig. 7. Vehicle trajectories generated through hybrid tracking PTMS algorithm. These are drawn from the same sequence shown in Figs. 5 and 6.

Fig. 8. Initialization of cluster means. Solid lines denote cluster centers and dots represent individual vehicle trajectories.



(a) (b)

Fig. 9. (a) Final clustering result of lane center detection without using RANSAC estimation. (b) Final result using RANSAC estimation. Note that there is insufficient trajectory data to accurately represent the entry lane farthest from the camera.



(a) (b)

Fig. 10. (a) Trajectory points of single tracked vehicles containing outliers. (b) Lane centers estimated by the output of the modified K-means clustering algorithm.

tracking of one vehicle is lost, we place a cross to highlight the position predicted by KF. In this sequence, the results did not vary significantly whether or not KF prediction was employed. Due to the camera viewpoint, most of the lost vehicles occur very near to the camera, where interframe dislocation is large. On the other hand, vehicles that are farthest from the camera are almost always successfully tracked due, in large part, to the effectiveness of background modeling. In about half the situations, where vehicles are moving away from the camera (i.e., large interframe motion), the KF has insufficient time to initiate the tracking process and in the initial frames tracking is lost. Success of KF is sometimes dependent on the camera viewpoint, it is particularly effective when parallel to the traffic flow. This situation requires further investigation and could benefit from the use of higher sampling frequency (reducing interframe displacement) as well as a more careful design of the filter parameters.

Fig. 5 shows the result of occlusion reasoning applied to a different sequence. The two cars in the left side of the image are detected as a single blob. With the PTMS algorithm, we can determine that it corresponds to two cars in the previous frame seen in Fig. 6. The detected blob is displayed with a cross in the middle of its bounding box.

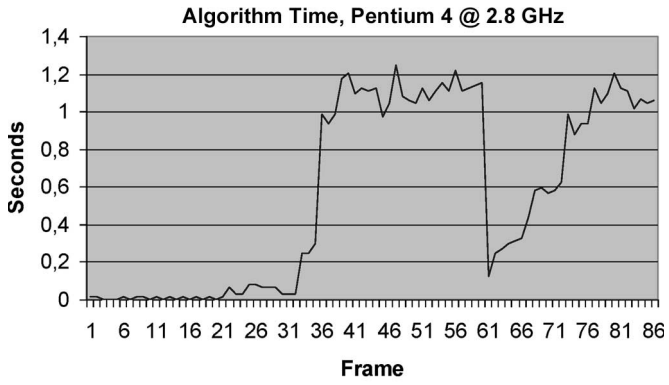
In Fig. 7, we display the point trajectories generated by the use of KF and PTMS algorithm applied to the same sequence from which Figs. 5 and 6 were drawn. For vehicles previously detected and whose tracking is subsequently lost, the trajectory points are predicted using the KF output.

B. Cluster Analysis

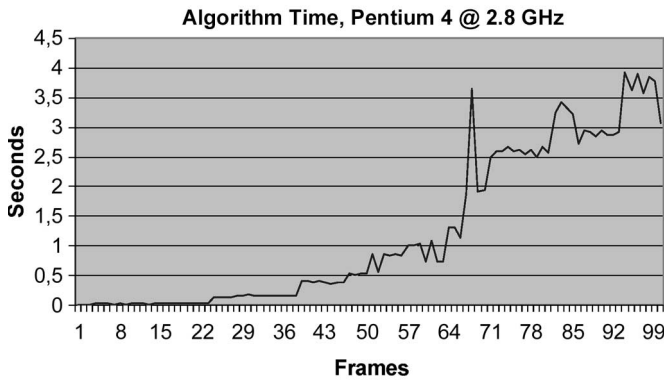
The point trajectories of single vehicles are used to create an initial set of clusters shown in Fig. 8 to estimate the lane centers for a straight segment of the highway. We have used an initial set with a total of 237 partial trajectories and clustered 30 trajectories per lane to estimate the centers shown in Fig. 9. We make no compensation for the height of the vehicles, and thus, the estimation of lane centers is biased by a factor in proportion to the average vehicle height. However, this does not affect the counting operations for calculation of flow density. The results of applying the initialization of the clustering algorithm described in Section IV-A are shown in Fig. 8. Initial cluster means are represented by solid lines and original trajectory points by dots.

The stable clusters produced after termination of the algorithm are illustrated in Fig. 9. The final number of trajectories added to each initial cluster are also labeled. Fig. 9(a) and (b) shows a comparison of the result with and without the RANSAC approach. It shows that RANSAC leads to an improved estimate of the lane center on the inner lane farthest from the camera. Lane changes are caused by vehicles moving to the outer lane to avoid new vehicles entering the highway from the entry lane farthest from the camera.

The clustering approach has also been tested on curved sections of highway. From a total of 175 partial trajectories in the video sequence, the modified K-means algorithm uses 30 trajectories per lane to estimate the centers. Again, RANSAC can be used to make the clustering stage robust to lane changes. Fig. 10 demonstrates the algorithm can work with any type of road geometry.



(a)



(b)

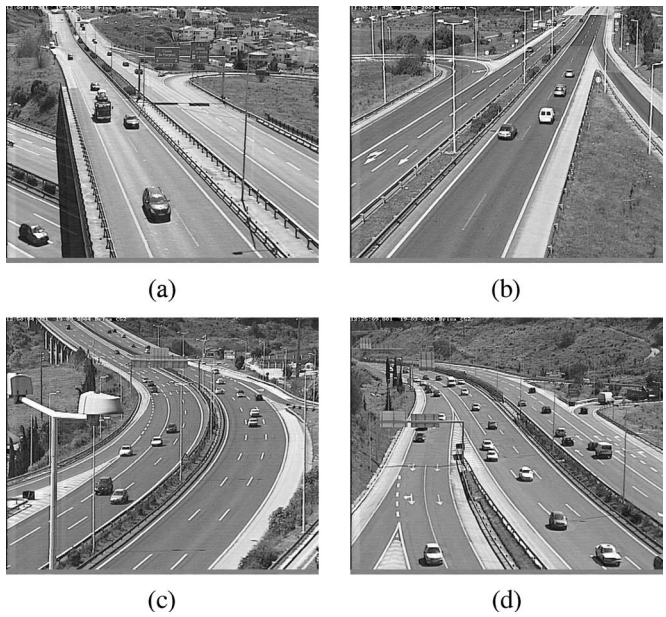
Fig. 11. (a) Execution time taken by clustering loop for straight section of highway shown in Fig. 9. (b) Execution time taken for the curved section of highway shown in Fig. 10.

The processing times for each frame in the sequence are shown in Fig. 11. For curved sections of the highway, the time taken to perform clustering loop is longer, as the order of the polynomial is successively incremented up to $m = 3$. In each case, the algorithm starts with zero clusters and adds two new trajectories per frame. This experiment demonstrates the potential for real-time performance of lane detection.

C. Lane Classification and Traffic Analysis

We applied the lane classification scheme to a relatively complex highway scene near Lisbon, Portugal. A single PTZ camera observes two intersecting highways—the junction of A5 and Circular Regional Interior de Lisboa (CRIL). CRIL has four primary lanes, two entry lanes, and one exit lane in the field of view closest to the camera, whereas A5 has seven primary lanes and four exit lanes. The traffic is usually heaviest on A5. In this case, the sequences were recorded on a very windy day, resulting in unwanted camera motions. This motion can reach a maximum of 5 pixels and represents a challenging test scenario. Typical camera views and a schematic of the road layout are shown in Figs. 12 and 13. The main directions and numbers of lanes are summarized in Table I.

Using the information displayed in the schematic representation of Fig. 13 and the data in Table I, the goal is to determine the direction in which the camera is viewing and categorize the lanes and lane directions. Each lane is classified into one of the following categories: *highway lane*, *entry*, or *exit*. For each lane, the vehicle trajectories are counted and this information is displayed on the images.



(a) (b) (c) (d)

Fig. 12. PTZ camera views for each of the four main directions of traffic flow. (a) and (b) are opposite views of highway CRIL. (c) and (d) are opposite views of highway A5.

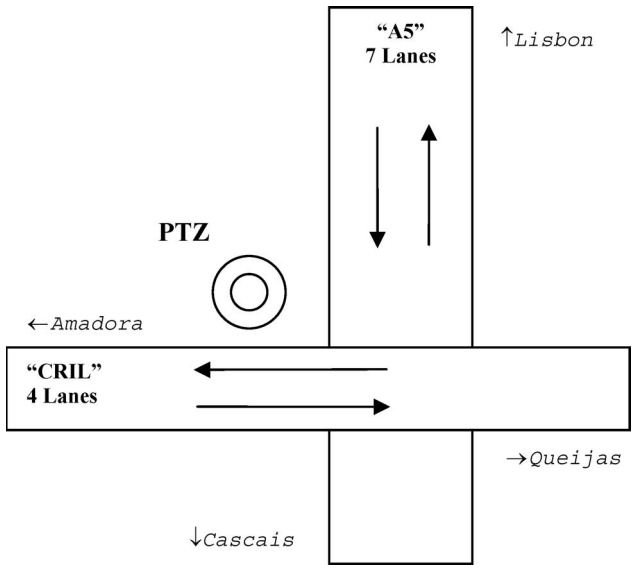


Fig. 13. Schematic layout of the highway intersection for A5/CRIL in the vicinity of the PTZ camera.

TABLE I
SUMMARY OF TOTAL NUMBERS OF LANES AND DESTINATIONS FOR MAIN DIRECTIONS OF TRAFFIC FLOW

Highway name	A5	CRIL
Minimum Number of Lanes	11	6
Direction Looking Right	Cascais	Amadora
Direction Looking Left	Lisbon	Queijas

The camera orientation is easily found by estimating the angle between the main direction of traffic flow and the horizontal axis. To determine which highway is currently being viewed, we use the number of detected lanes together with the

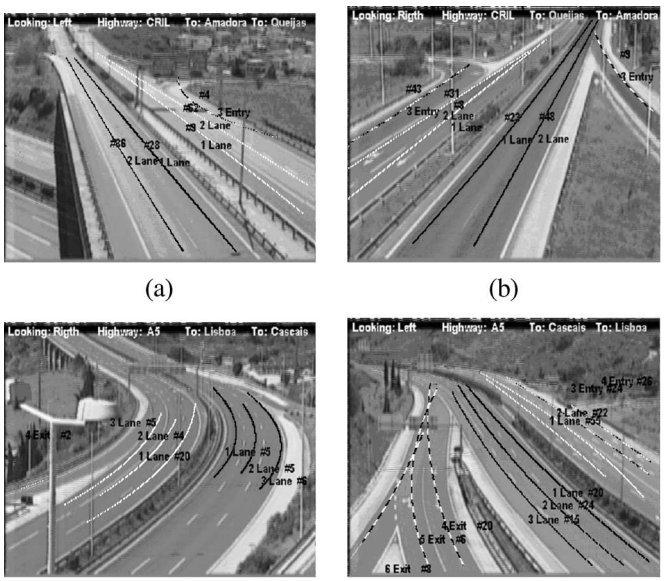


Fig. 14. Lane detection and classification result for (a) and (b) highway CRIL and (c) and (d) highway A5. The numbers of trajectories are also shown. The camera viewing and traffic flow direction are displayed at the top of the image. The traffic flow directions are distinguished by the color of the highlighted lane centers. (a and b) black—toward Amadora; white—toward Queijas. (c and d) black—toward Cascais; white—toward Lisbon. Exit or entry lanes are marked as black and white striped in (a), (b), and (d).

information presented in Table I. After the camera orientation is found, the destination IDs can also be established from the table. This is the only calibration information required to perform the lane classification once the lane detection is completed. The classification of entry or exit lanes can be retrieved from the trajectory data. The algorithm performs well when the lanes are nearly straight, e.g., CRIL highway. Here, the number of vehicles in each lane/direction and number of lane changes can be successfully calculated. This information is very useful for automated traffic flow analysis. The results are shown in Fig. 14.

The results obtained for the A5 highway, which are shown in Fig. 14(c)–(d), are not as good, especially when the camera is looking right. There are several reasons for this. A greater number of lanes means that vehicle resolution is smaller, and hence, vehicle detection is more sensitive to camera motion. Furthermore, when the camera is looking right, the zoom level is greater than when looking left, and this produces a degraded result. The success of the classification scheme is highly dependent on the quality of the lane modeling, which, in turn, is dependent on the output of vehicle tracking module. Hence, the results tend to be better for straight roads. Further results of detection and classification of lanes with color images can be found at: http://vislab.isr.ist.utl.pt/inteltraf/.

D. Trajectory Indexing and Retrieval

We present some retrieval results for a user-defined query trajectory. The aim is to detect the vehicles that change lane in a dangerous situation or enter a prohibited zone. The user first specifies a query Q simply by sketching a straight line on a representative background scene. In this case, the user wishes to detect those vehicles crossing the boundary between the entry lane and first lane in Fig. 12(b), which is shown in the upper left image quadrant. The trajectories database is then queried for a specific time duration to restrict the number searched. We specify a search for all S_j that satisfy $d_{\min}(Q, S_j) < 1$. The result of performing this query is shown in Fig. 15. For verification purposes, we calculate the ground truth using each tracking point as a key index to the original video sequence.

Fig. 15. Performing a similarity search under the $d_{\min}$ distance function with a user-specified query. Retrieved vehicle trajectories are displayed.

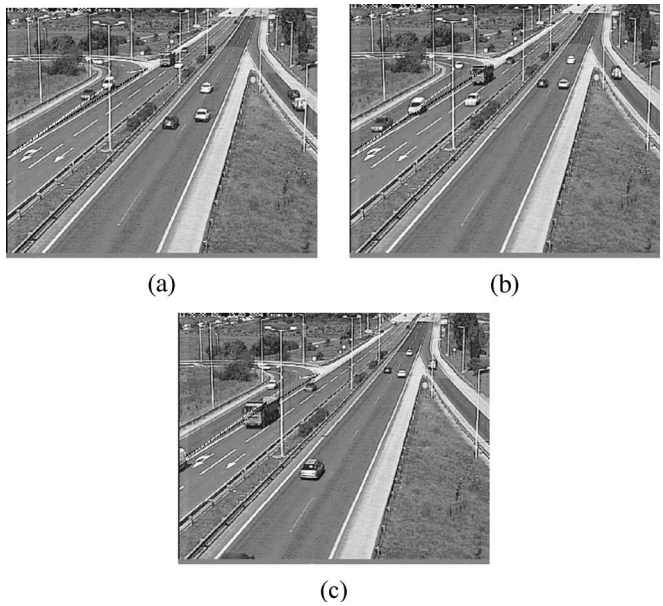


Fig. 16. Retrieved sequence for vehicle trajectory that satisfies similarity query search at the (a) beginning, (b) middle, and (c) end of the video clip. Note that the white vehicle in the top left quadrant of image (a) does cross the boundary marking of the entry and highway lanes. It can be seen just leaving the camera view in image (c).

The position of a tracked vehicle during the beginning, middle, and end of a retrieved trajectory is shown in Fig. 16. It can be verified that the vehicle trajectory intersects the user-specified query and, indeed, crosses the lane boundary between entry and outer highway lanes. However, one of the retrieved trajectories for this example query posed is a misdetection. This can arise due to two factors. The PTMS algorithm does not correct for the object height (parallax), and therefore, some high-sided vehicles such as trucks will generate a heavily biased trajectory. Furthermore, the algorithm sometimes classifies

TABLE II
SUMMARY OF RETRIEVAL RESULTS FOR EXAMPLE QUERY

Total number of trajectories	237
Correct matches retrieved	2
False positives retrieved	2
False negatives retrieved	0

convoys of vehicles traveling together as a connected blob and labels it as a single vehicle. This produces an incorrect blob centroid localization error. However, an advantage of this approach is that search and retrieval of trajectories of interest can be performed in real time due to the low computational cost incurred by indexing polynomial coefficients. This would not be possible using the Hausdorff distance metric.

The overall results for this particular query are summarized in Table II.

VI. DISCUSSION

As stated previously, the algorithm performs best when the lanes are nearly straight, e.g., CRIL highway. Here, the number of vehicles in each lane/direction and number of lane changes can be successfully calculated. In Fig. 14(b), it is seen that the lane detection is accurate apart from lane 3 on the left side of the image heading towards Queijas, which is estimated as a straight road. This is due to a number of factors, including unstable camera motion, presence of a lighting pole, and overall distance from the camera. This degrades the performance of the tracking algorithm. Similarly, the vehicles joining the highway from the entry lane just appear in the camera view for only a few frames making initialization of the KF state vector difficult for prediction stage. In general, KF tracking performs worse for curved road segments, possibly due to a conservative adjustment of the maneuverability index (see Section III-A). A similar problem occurs in Fig. 14(a) with the road on the right side of the image heading towards Queijas. The result is not very accurate due to incomplete vehicle tracking. Furthermore, the unstable camera motion due to high winds causes a strong blurring effect in the background model. This blur is especially apparent in the upper right quadrant of Fig. 14(a). As the background model is built from 150 frames in the sequence, any motion present during this period will blur the captured tracking sequence. Alternatively, one could use multiple background models to cope with some motion variability, as in [29].

The lane geometry estimated in Fig. 14(d) is relatively good, although the accuracy is much higher in the image region closer to the camera than those regions farther away. The detection on the right half of the image is not good due to small appearance of vehicle size. This results in misclassification of lane types carrying traffic towards Lisbon. The algorithm misclassifies these lanes as two entry lanes rather than a correct labeling as exit lanes. Furthermore, the detection of lane centers is also poor due to uncontrolled camera motions and high local curvature of the lanes in the upper half of the image. This causes a degradation of tracking performance with consequent effects on trajectory clustering.

We can summarize the traffic scene interpretation data contained in all four PTZ camera views, i.e., approximate numbers of detected vehicle trajectories per lane, where each lane direction is classified and labeled, as shown in Fig. 17. It can be concluded that the present study provides a useful basis for traffic flow analysis from multiple views obtained using standard traffic monitoring cameras.

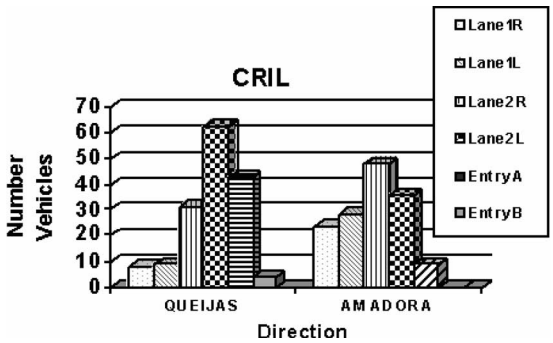


Fig. 17. Summary of the traffic flow in each lane derived from an analysis of vehicle trajectories and lane classification using multiple views from a stationary PTZ camera.

VII. CONCLUSION

First, the paper proposes an algorithm for vehicle tracking with the following characteristics: temporal integration with a KF, time-consistent merging-and-splitting of overlapped detected blobs, aggregation of trajectory data to estimate lane centers, and removal of the need for calibrated cameras. The results demonstrate the feasibility of using uncalibrated stationary or PTZ cameras to analyze traffic behavior in real time. The algorithm is viewpoint independent and does not make any *a priori* assumption regarding lane geometry. The results can be used as input to higher level traffic monitoring systems for estimating traffic speed, frequency of lane changes, accident detection, and classification of anomalous driver behavior. We use some limited assumptions regarding camera zoom and image scale. One drawback of the clustering approach is that due to occlusions, vehicle trajectories are sometimes misdetected, and hence, partitioned into erroneous cluster sets. It is often difficult to distinguish these from genuine lane changes at the postprocessing stage.

Second, an algorithm that performs lane detection and classification for highway scenes using only vehicle trajectory data and a basic knowledge of the road layout is described. This can be used to generate a higher level scene description, which is useful for automated traffic flow analysis. The method works with both uncalibrated PTZ and static road traffic cameras and can cope with variable lane geometry. The lane detection stage is robust to vehicle lane changes and missing or incomplete trajectory data through use of robust estimators based on RANSAC. The classification scheme works well in the case of straight (or almost straight) roads but performs poorly when road curvature is high. This could be due to instability in vehicle tracking leading to incomplete trajectories, or unsuitable distance metrics.

This paper can be extended in multiple directions. First, we can improve the vehicle detection and tracking by relying on more flexible models for the background. Second, data association and detection of merge and split situations can be improved

by better tuning of the KF or using the time-varying version of the filter. We also aim to explore other clustering algorithms such as self-organizing maps [30], which have been used to improve vehicle tracking. Finally, to deal with strong perspective effects, we could envisage an iterative process whereby the estimated lane centers are used to determine a projective planar transformation that would make these lanes parallel in the image. The corrected image would be used for reestimating the highway lanes, and hence, improve classification.

To conclude, we modestly hope that this paper has demonstrated the power of motion analysis for high-level semantic description of highway structure and traffic understanding and that the use of robust estimation over extended periods of time (as opposed to static image analysis) allows designers and engineers to achieve the desired levels of accuracy and robustness in future traffic surveillance systems.

REFERENCES

[1] D. Gutchess, M. Trajkovics, E. Cohen-Solal, D. Lyons, and A. K. Jain, "A background model initialization algorithm for video surveillance," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, Vancouver, BC, Canada, Jul. 2001, pp. 734–740.

[2] O. Javed, K. Shafique, and M. Shah, "A hierarchical approach to robust background subtraction using color and gradient information," in *Proc. Workshop Motion Video Comput.*, Orlando, FL, Dec. 2002, pp. 22–27.

[3] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 3, pp. 334–352, Aug. 2004.

[4] A. Prati, I. Mikic, C. Grana, and M. M. Trivedi, "Shadow detection algorithms for traffic flow analysis: A comparative study," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Oakland, CA, Aug. 2001, pp. 340–345.

[5] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Proc. 2nd Eur. Workshop Adv. Video Based Surveillance Syst.*, Kingston, U.K., Sep. 2001.

[6] R. Elgammal and R. Duraiswami, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, Jul. 2002.

[7] R. Cuchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting objects, shadows and ghosts in video streams by exploiting colour and motion information," in *Proc. Int. Conf. Image Anal. Process.*, Palermo, Italy, Sep. 2001, pp. 360–365.

[8] D. Koller, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning," in *Proc. 3rd Eur. Conf. Comput. Vis.*, Stockholm, Sweden, May 1994, pp. 189–196.

[9] Y. Jung, K. Lee, and Y. Ho, "Content-based event retrieval using semantic scene interpretation for automated traffic surveillance," *IEEE Trans. Intell. Transp. Syst.*, vol. 2, no. 3, pp. 151–163, Sep. 2001.

[10] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 108–118, Jun. 2000.

[11] H. Veeraraghavan, O. Masoud, and N. Papanikolopoulos, "Computer vision algorithms for intersection monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 2, pp. 78–89, Jun. 2003.

[12] D. J. Dailey, F. W. Cathey, and S. Pumrin, "An algorithm to estimate mean traffic speed using uncalibrated cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 98–107, Jun. 2000.

[13] N. Todd, N. Schoepflin, and D. J. Dailey, "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 2, pp. 90–98, Jun. 2003.

[14] W. Hu, X. Xiao, D. Xie, T. Tan, and S. Maybank, "Traffic accident prediction using 3-D model-based vehicle tracking," *IEEE Trans. Veh. Technol.*, vol. 53, no. 3, pp. 677–694, May 2004.

[15] D. Makris and T. Ellis, "Path detection in video surveillance," *Image Vis. Comput.*, vol. 20, no. 12, pp. 895–903, Oct. 2002.

[16] C. Stauffer and W. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[17] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image Vis. Comput.*, vol. 14, no. 8, pp. 609–615, Aug. 1996.

[18] G. Kollios, S. Sclaroff, and M. Betke, "Motion mining: Discovering spatio-temporal patterns in databases of human motion," in *Proc. ACM SIGMOD Workshop Data Mining Knowl. Discov.*, Santa Barbara, CA, May 2001.

[19] F. Bashir, A. Khokhar, and D. Schonfeld, "Segmented trajectory-based indexing and retrieval of video data," in *Proc. IEEE Int. Conf. Image Process.*, Barcelona, Spain, 2003, pp. 623–626.

[20] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic, "Discovering clusters in motion time-series data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2003, pp. 375–381.

[21] D. Buzan, S. Sclaroff, and G. Kollios, "Extraction and clustering of motion trajectories in video," in *Proc. Int. Conf. Pattern Recog.*, Cambridge, U.K., Aug. 2004, pp. 521–524.

[22] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[23] J. Lou, Q. Liu, T. Tan, and W. Hu, "Semantic interpretation of object activities in a surveillance system," in *Proc. 16th Int. Conf. Pattern Recog.*, 2002, pp. 777–780.

[24] E. Brookner, *Tracking and Kalman Filtering Made Easy*. Hoboken, NJ: Wiley, 1998.

[25] D. Jacobs, D. Weinshall, and Y. Gdalyahu, "Condensing image databases when retrieval is based on non-metric distances," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jan. 1998, pp. 596–601.

[26] S. Khalid and A. Naftel, "Evaluation of matching metrics for trajectory-based indexing and retrieval of video clips," in *Proc. 7th IEEE Workshop Appl. Comput. Vis.*, Breckenridge, CO, Jan. 2005, pp. 242–249.

[27] H. Spath, *Cluster Analysis Algorithms*. Chichester, U.K.: Ellis Horwood Ltd., 1980.

[28] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using B-Snake," *Image Vis. Comput.*, vol. 22, no. 4, pp. 269–280, Apr. 2004.

[29] T. E. Boult, R. J. Micheals, X. Gao, and M. Eckmann, "Into the woods: Visual surveillance of noncooperative and camouflaged targets in complex outdoor settings," *Proc. IEEE*, vol. 89, no. 10, pp. 1382–1402, Oct. 2001.

[30] A. Bevilacqua, L. Di Stefano, and S. Vaccari, "Occlusion robust vehicle tracking based on SOM," in *Proc. WACV/MOTION*, Breckenridge, CO, Jan. 2005, pp. 84–89.

José Melo received the Licenciatura degree in information systems and computer engineering from the Instituto Superior Técnico, Lisbon, Portugal, in 2002 and the M.Phil. degree in computer vision from the University of Manchester, Manchester, U.K., in 2005.

He was with the Computer Vision Group at the Institute for Systems and Robotics, Lisbon, from 2002 to 2005, collaborating in the national project Inteltraf and working on the development of automatic surveillance algorithms for highway traffic. He was a Knowledge Transfer Partnership (KTP) associate with the University of Manchester from 2004 to 2005, working on a research project in the area of closed-circuit television systems. He is currently developing computer vision solutions for automatic inspection in the semiconductor industry.

Andrew Naftel (M'05) received the B.Sc. degree in mathematics and the Ph.D. degree in photogrammetric engineering from the University of Bradford, Bradford, U.K., in 1985 and 1989, respectively.

During 1990–1999, he was a Lecturer, and then a Senior Lecturer, in the Department of Mathematics and Statistics, University of Central Lancashire, Preston, U.K. In 2000, he joined the Department of Computing, University of Bradford. Since 2002, he has been a Lecturer in the Department of Computation, University of Manchester Institute of Science and Technology, Manchester, U.K., which became the School of Informatics, University of Manchester, in October 2004. His research interests include computer vision, image and video processing, and content-based image and video information retrieval. He is a grant holder and a principal investigator for two nationally funded knowledge transfer partnership programs and works closely with a major manufacturer of closed-circuit television systems and solutions.

Dr. Naftel is a member of the Institute for Learning and Teaching and the British Computer Society.

Alexandre Bernardino (M'05) was born in Lisbon, Portugal, in 1971. He received the Ph.D. degree in electrical and computer engineering in 2004 from the Instituto Superior Técnico (IST), Lisbon.

He is an Assistant Professor at IST and a Researcher in the Computer Vision Laboratory, Institute for Systems and Robotics-Lisbon. He participates in several national and international research projects in the fields of robotics, cognitive systems, computer vision, and intelligent transportation. He has published several articles in international journals and conferences, and his main research interests focus on the application of computer vision, cognitive science, and control theory to advanced robotic and surveillance systems.

José Santos-Victor (M'85) received the Ph.D. degree in electrical and computer engineering from the Instituto Superior Técnico (IST), Lisbon, Portugal, in 1995, in the area of computer vision and robotics.

He is an Associate Professor at the Department of Electrical and Computer Engineering of IST and a Researcher in the Computer and Robot Vision Laboratory, Institute of Systems and Robotics. He is the scientist responsible for the participation of IST in various European and national research projects in the areas of computer vision and robotics. His research interests are in the areas of computer and robot vision, particularly in the relationship between visual perception and the control of action, biologically inspired vision and robotics, cognitive vision, and visual controlled (land, air, and underwater) mobile robots.

Prof. Santos-Victor is an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS.