ACIVS, Advanced Concepts for Intelligent Vision Systems, Brussels, Belgium, Aug. 31-Sept. 3, 2004

# RETRIEVAL OF VEHICLE TRAJECTORIES AND ESTIMATION OF LANE GEOMETRY USING NON-STATIONARY TRAFFIC SURVEILLANCE CAMERAS

[1,2]*José Melo*, [1]*Andrew Naftel*, [2]*Alexandre Bernardino and* [2]*José Santos-Victor*

jpqm@isr.ist.utl.pt
[1]University of Manchester Institute of Science & Technology, Manchester M60 1QD, UK
[2]Instituto Superior Técnico, Av. Rovisco Pais 1049 - 001 Lisboa, Portugal

## ABSTRACT

A tracking system is presented for obtaining accurate vehicle trajectories using uncalibrated traffic surveillance cameras. Techniques for indexing and retrieval of vehicle trajectories and estimation of lane geometry are also presented. An algorithm known as Predictive Trajectory Merge-and-Split (PTMS) is used to detect partial or complete occlusions during object motion. This hybrid algorithm is based on the constant acceleration Kalman filter and a set of simple heuristics for temporal analysis. The resulting vehicle trajectories are modeled using variable low-order polynomials.

A comparative evaluation of several distance metrics used in trajectory cluster analysis, indexing and retrieval is also presented. We propose some changes to metrics presented in previous work and make a comparative study with a modified form of the Hausdorff distance.

Some preliminary results are presented on the estimation of lane geometry through K-means clustering of individual vehicle trajectories using the proposed metrics. An advantage of our approach is that estimation of lane geometry can be performed with non-stationary, uncalibrated traffic cameras in real time.

## 1. INTRODUCTION

Rising traffic levels and increasingly busier roads are a common problem across the globe. Consequently, there is an urgent need to develop intelligent traffic surveillance systems that can play an important role in highway monitoring and road management schemes. One purpose is to detect and signal potentially dangerous situations.

This paper addresses the problem of generating accurate vehicle trajectories through object segmentation, motion tracking and screening of partial and complete occlusions, using non-stationary, uncalibrated traffic cameras. Typically, these are operator-controlled pan-tilt-zoom (PTZ) cameras. In [20] is demonstrated that by building a self-consistent aggregation of many individual trajectories and by taking into account vehicle lane changes, lane geometry can be estimated from stable video sequences. This information could be used as input to some higher level system to classify normal and abnormal driving situations such as dangerous lane weaving.

In our work, rather than performing object tracking under partial or total occlusion, we describe an occlusion reasoning approach that detects and counts the number of overlapped objects present in a segmented blob. Trajectory points are then classified according to whether they are generated by a single or overlapped object. Previously [19] we have described the Predictive Trajectory Merge-and-Split (PTMS) algorithm for performing the aforementioned task. It uses a Kalman filter (KF) and a set of simple heuristic rules to enforce temporal consistency on merging and splitting overlapping objects within detected blobs. The method is independent of the camera viewpoint and requires no *a priori* calibration of the image sequences.

All the obtained vehicles trajectories are modeled using variable low-order polynomials.

We propose some changes to the metrics proposed in previous work [20] and propose a modified form of the Hausdorff distance for a comparative study of the inter-trajectories distance.

We evaluate the quality of the metrics by analyzing the results obtained by querying a trajectories database using a trajectory query.

The chosen distance metric is also used in the trajectories clustering to estimate lane centers. For the estimation of lane centers, we use a modified K-Means algorithm in conjunction with a RANSAC approach to exclude outliers and suppress the effects of vehicle lane changes on the estimation of lane geometry.

## 2. REVIEW OF PREVIOUS WORK

The starting point for much work in motion tracking is the segmentation of moving objects based on background subtraction methods [1] [2]. 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.

Shadow detection and elimination strategies have been commonly employed to remove extraneous segmented features [4] [5] [6] [7].

It is also important to account for partial and complete occlusions in the video data stream [7] [8] [9] [10]. Occlusion detection can be performed using an extended Kalman filter that predicts the position and 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].

More specifically, higher level traffic analysis systems have been developed for accident detection at road intersections [9] [11] and estimating traffic speed [12] [13]. Rather general techniques for object path detection, classification and indexing have also been proposed [10] [14], [15] [16] [17].

Object trajectories have previously been used for indexing and retrieval of video data [10] [18] [20]. However, the performance of different similarity metrics has not been well explored.

The contribution of this paper is to show that our proposed distance metrics give equal or better results than the Hausdorff distance, and that these metrics can be used to cluster vehicle trajectories from uncalibrated image sequences to estimate lane geometry. This information can be used as input to some higher level system to classify normal and abnormal road situations.

## 3. PREDICTIVE TRAJECTORY MERGE-AND-SPLIT (PTMS) ALGORITHM

The proposed system uses a multi-stage approach to determining the vehicle motion trajectories and thereafter, the estimated lane geometry. Firstly, we build a background model to segment foreground objects. A detected foreground blob comprises a connected region having more than a certain pre-defined minimum number of pixels ($K_{min}$) in its area. A constant acceleration Kalman Filter (KF) is used to track the blobs through image coordinate space. The PTMS algorithm [19] is then used to perform a time-consistent analysis of those detected blobs allowing for merging and splitting due to partial and complete occlusions. An overview of the system is shown in Fig. 1. We now briefly describe the main features of the PTMS algorithm.

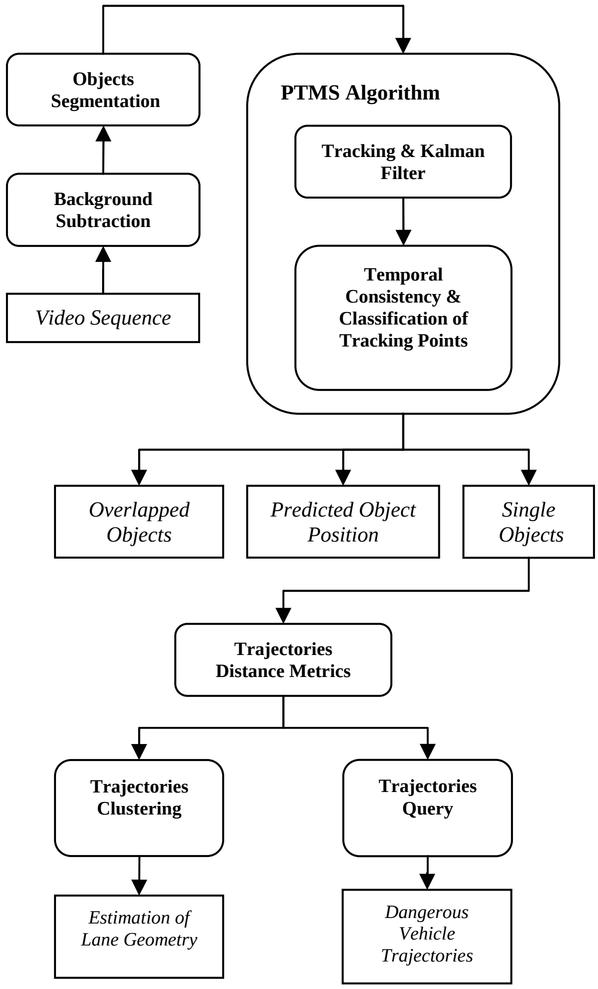


*Fig. 1.* ***Block diagram of the proposed system***

The presence of shadows or ‘near’ occlusions caused by traffic congestion can seriously degrade 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 rules 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 KF to predict for each blob the most likely position in the next frame that the blob will appear. Each blob is considered to have a number of children, i.e. number of different objects a blob is composed of. At the beginning, every blob is initialized as having one child. For each frame and for every blob:

Step 1. Determine whether there is a 1-1 correspondence by checking size and position of blobs in consecutive frames and comparing positions and sizes.

Step 2. For every blob that does not match the previous condition; determine whether the size has decreased by more than Ω (where Ω is expressed as a percentage). If so, decrease the number of its occluded objects by 1.

Step 3. Determine whether any blob has decreased its size by less than Ω. If so store that information.

Step 4. Determine whether any new blob has appeared in the vicinity of a blob that had decreased its size and have a number of child blobs greater than 1. If so, decrease the number of occluded objects in the old blob. This implies the old blob was occluding the new blob.

Step 5. Check if there are any new blobs in the new frame.

Step 6. If there are any new blobs in the same position of several old blobs, it means that the new blob is composed of the old blobs, and the number of its children is increased by the number of the old blobs minus 1.

The algorithm works well most of the time. The principal drawback is when the initial blob is composed of several objects. In this case, it will be mis-detected as one single object. To tackle this problem, a template or model-based approach 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 5.

## 4. TRAJECTORIES RETRIEVAL AND ESTIMATION OF LANE GEOMETRY

In highly constrained environments, such as highways, it is tempting to use vehicle trajectories rather than image analysis of static scenes to determine the lane geometry. The former approach has a number of advantages:

- It enables use of non-stationary pan-tilt-zoom cameras rather than calibrated static cameras.
- Techniques based on object trajectories are independent of scale and viewpoint.
- Motion data is generally more robust to light variation and sensor noise than static image scene data.

The method assumes that the average lane width in image coordinates is approximately known or can be easily estimated. However, it does not require *a priori* knowledge of the number of lanes or road geometry, i.e. whether it is a straight or curved section of highway.

### 4.1. Overview of Trajectory Modelling

The trajectories are pre-processed to remove obvious inconsistencies in the data caused by tracking errors, sensor noise or camera motion instabilities (high winds can be a problem). Excluded trajectories are deemed to be those having inter-point separation greater than some lower bound or curve length below some threshold value. By analysing all the trajectories generated, we can discover the main direction of traffic flow, and by doing so, the predominant coordinate that is monotonically increasing or decreasing. This can be performed by aggregating the slopes from the start and end points of each trajectory found.

We then fit a least squares polynomial of degree $M$ for each trajectory in the chosen coordinate direction. Starting from $M = 1$, we use the average residual error of fit to ascertain the optimal value of $M$. If the error is greater than some threshold, $M$ is increased by 1 and the trajectory is re-fitted ($M \leq 3$). For all the highway scenes tested, we have found that low degree polynomials up to order 3 are sufficient to describe the lane curvature and trajectories denoting lane changes. Finally, the RANSAC algorithm [18] is used in conjunction with least squares to eliminate outliers or mis-classified tracker points.

In section 4.2, we propose some changes to the metrics presented in previous work [20]. We make a comparative study between the former metrics, the proposed metrics and a modified version of the Hausdorff distance.

As an example of the use of the proposed metrics we show a trajectories data-base query, and the clustering of trajectories to estimate lane centres.

The method used for estimation of lane centres by clustering of vehicles trajectories is described in section 4.4 with a brief description of the algorithm.

### 4.2. Inter-Trajectory Distance Metrics

In some approaches [10], the separation between trajectories is calculated using modified Hausdorff distance. Although this works with arbitrary points sets, it is extremely sensitive to outliers and expensive to compute as it involves $O(MN)$ operations, where $M$, $N$ are the sizes of the trajectory point sets. Computation time can be considerably reduced if we use the polynomial coefficients for each trajectory to evaluate simpler expressions for the metrics. In [20], is proposed the use of Mean

($d_{\mathrm{mean}}$), Maximum ($d_{\max}$) and Minimum ($d_{\min}$) distances to measure inter trajectories distance.
A geometric interpretation of $d_{\max}$ and $d_{\min}$ is shown in Fig. 2.

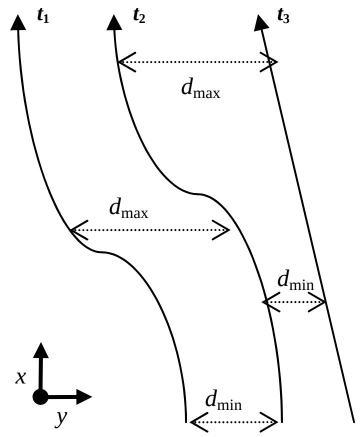


Fig. 2. *Interpretation of distance measures* $d_{\max}$, $d_{\min}$ *in terms of difference polynomial d.*

In this paper we present some alternative metrics to these three types of distances. For the Mean distance we propose the use of Root Mean Square instead of the Mean distance, for the Minimum distance we propose the use of the same formula complemented with a scan line search to check for polynomial intersections. For the Maximum distance metric we do not propose any change.
We assume that $x$ is the independent variable in the model.
Let $T_1$ and $T_2$ denote two sets, where

$$\begin{aligned} T_1 &= \{ t_1(x) : x \in [a,b] \} \\ T_2 &= \{ t_2(x) : x \in [c,d] \} \end{aligned} \quad (1)$$

$t_1$(x), $t_2$(x) are polynomial trajectory models. The difference polynomial $d(x)$ can be defined as the set $D$ such that

$$D = \{ d(x) = t_1(x) - t_2(x) : x \in [a,b] \cup [c,d] \} \quad (2)$$

We define a set of distance measures $d(T_1,T_2)$ between $T_1$ and $T_2$ using D as follows:

$$d_{\max}(T_1,T_2) = \sup_{x\in I} \{ d(x) \} \quad (3)$$

$$d_{\min}(T_1,T_2) = \inf_{x\in I} \{ d(x) \} \quad (4)$$

$$d_{mean}(T_1,T_2) = \frac{1}{l_2 - l_1} \int_{l_1}^{l_2} (d(x))dx \quad (5)$$

The proposed metrics are:

$$dc_{\min}(T_1,T_2) = \inf_{x\in I} \{ d(x) \} \quad (6)$$

$$dc_{mean}(T_1,T_2) = \sqrt{\frac{1}{l_2 - l_1} \int_{l_1}^{l_2} \{ d(x) \}^2 \, dx} \quad (7)$$

Where $l_1$, $l_2$ are the lower and upper bounds on $I$, and $I = [a, b] \cup [c, d]$.
In Eq. (4) a closed form expression is obtained by differentiation and finding the stationary points of $d(x)$. If the trajectories intersect, none of these stationery points will correspond to the point with minimum distance. Therefore, the Minimum Distance in Eq. (6) is an extension to Eq. (4), a line search technique is employed to check for polynomial intersections, if the polynomials do not intersect Eq. (4) is used.
For comparison purposes, we propose to calculate the Mean, Maximum and Minimum distances, by using a set of modified Hausdorff-type distance measures $d_H(T_1,T_2)$, defined as:

$$d_{h\max}(T_1,T_2) = \max_{t_i \in T_1} \max_{t_j \in T_2} \| t_i - t_j \| \quad (8)$$

$$d_{h\min}(T_1,T_2) = \min_{t_i \in T_1} \min_{t_j \in T_2} \| t_i - t_j \| \quad (9)$$

$$d_{h\mathrm{mean}}(T_1,T_2) = \max_{t_i \in T_1} \min_{t_j \in T_2} \| t_i - t_j \| \quad (10)$$

where $t_i$ and $t_j$ represent the point sets of $T_1$ and $T_2$ respectively. In each case, we actually use the modified symmetric form of the Hausdorff distance metric, defined as

$$d_H(T_1,T_2) = \max(d_h(T_1,T_2), d_h(T_2,T_1)) \quad (11)$$

### 4.3. Trajectory Retrieval by Reference Query

We can recover vehicle trajectories of interest by specifying a reference query $T_Q$ and producing a rank-ordered list of similar trajectories according to one of the above specified distance metrics Eqs. (3)-(10). For example, we might wish to retrieve all vehicle trajectories that stray into the emergency stopping lane.
The Maximum distance metric can be used to query all the trajectories that lay with a certain maximum distance from the reference trajectory. Therefore, if the trajectories have similar lengths and a small Maximum distance they will have similar directions.
The Average distance metric can be used to search for trajectories which have a particular shape.
The Minimum distance metric can be used to query for the trajectories that lay in a minimum distance or that intersect the reference trajectory.

We present some results of a comparative evaluation of the distance measures in section 5. In section 5, we also present a practical example of the detection of vehicles which cross the straight line by using the corrected Minimum distance metric, $dc_{min.}$

After evaluating the results obtained in the trajectory query using the different distance metrics, presented in section 5, the metrics which show better results, $dc_{min}$, $dc_{mean}$ and $d_{max}$ are used to calculate the inter-cluster distances in the K-means algorithm that follows.

### 4.4. Modified K-means Cluster Analysis

Next, we briefly describe a robust K-means clustering algorithm that works in the coefficient space of the polynomials, by using the corrected form of the Maximum, Minimum and Average distance metric.

The following steps are used to build an initial set of cluster trajectories:

1. Create a reference linear trajectory $t_R$ at the edge of the image parallel to the main orientation of traffic flow.
2. Build a set with a trajectory $t_J$ that is most distant from $t_R$ using the $dc_{min}$ metric. A sensible choice of threshold is the estimated minimum lane width.
3. Find the next trajectory $t_{J+1}$ ($_J$ =1,2,...) that is most distant using $dc_{min}$ from all trajectories present in the set and has length greater than a certain threshold.
4. If the trajectory found has a distance $dc_{min}$ to the nearest neighbouring trajectory less than the estimated lane width, discard that trajectory.
5. Repeat from step 3 until no further trajectories can be added.

We now describe how to perform the clustering step. The trajectory-to-cluster distance is measured using $d_{max}$. When a trajectory is merged with a cluster, it is performed by merging the original cluster points and trajectory. In order to avoid a cluster becoming biased by merger with an outlier trajectory, each cluster is only remodelled using RANSAC for every 5 trajectories considered.

For all the trajectories, we measure the trajectory-to-cluster distance and merge each trajectory $t$ to the nearest cluster if $d_{max} < \tau$ (for some threshold $\tau$) and curve length $|t| > L_{min}$. For all $t_J$, we calculate min($d_{max}$'s) and add the trajectory indexed by this minimum to the current cluster. The cluster means are updated after a fixed number of repetitions. The number of trajectories added to each cluster is limited to 20 and for every 5 trajectories added, RANSAC is applied to modelling the cluster in order to eliminate outliers. It is found that use of $d_{max}$ for measuring trajectory-to-cluster separation discards most trajectories representing lane changes. RANSAC then removes most remaining outliers erroneously added to the cluster.

In the next section we present some results of applying these techniques to some traffic images captured on a Portuguese highway.

## 5. RESULTS

### TRACKING AND PTMS ALGORITHM

In Fig. 3, we show the result of background subtraction using grey scale sequences. Segmented objects whose areas are lower than $K_{min}$ are denoted in red whereas detected vehicles are coloured purple. We use a different colour to represent the bounding box of a tracked vehicle. When tracking of one vehicle is lost, we place a cross to highlight the position predicted by KF. Due to the camera viewpoint, most of the lost vehicles occur very near to the camera where inter-frame dislocation is large.

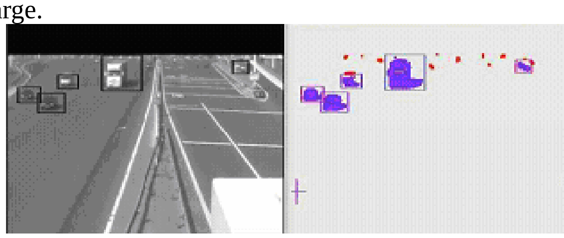

*Fig 3. Tracked vehicles on highway*

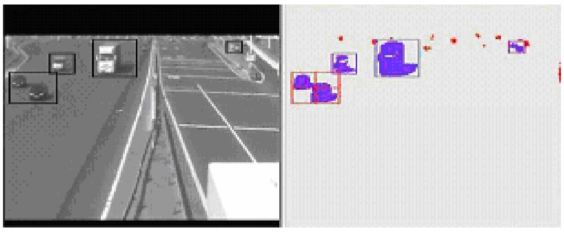

*Fig 4. Tracking and occlusion handling on highway*

Fig. 4 shows the result of occlusion handling applied to the previous figure. It can be observed that the two cars in the left of the image are detected as a single blob, and through the use of PTMS algorithm, we can determine that it corresponds to two cars in the previous frame. The detected blob is displayed with its bounding box in red with a cross drawn in the middle.

In Fig. 5 we display the point trajectories generated by use of KF and PTMS algorithm applied to the same sequence from Figs. 3 and 4. Trajectories in green correspond to single vehicles successfully tracked, whereas those in purple correspond to vehicles previously detected but whose tracking was subsequently lost. Here the points are predicted by output of KF. The red points correspond to trajectories of averaged position of two or more overlapped vehicles detected through use of PTMS.

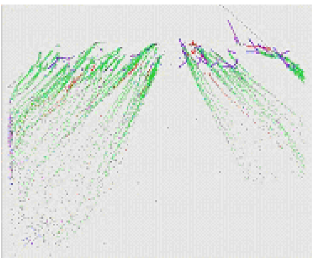

*Fig. 5. Vehicle trajectories generated through hybrid tracking and PTMS algorithm*

## DISTANCE METRICS

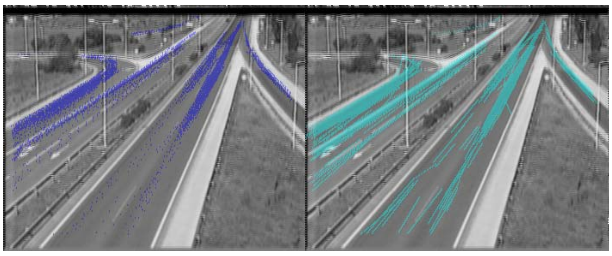

*Fig.6. a) Original points  b) Trajectories Interpolation*

On Fig. 6.a. we can see on dark blue the points that correspond to the blobs classified as single vehicles by the PTMS algorithm. On Fig. 6.b we can see the interpolation of the trajectories points using low order polynomials. This data consists of 237 trajectories calculated from 5 minutes of video.

In the following figures we show the result of trajectory retrieval using the Average, Maximum and Minimum Distances using the Hausdorff and the proposed metric.

A user query consists of a low order polynomial which is created by the interpolation of a set of points defined by the user. In Fig. 7 and all the following figures the query is represented by the yellow line.

The 10 best trajectories, which have the smaller distance to the user query, are retrieved in the following figures. The trajectories in red were calculated using the Hausdorff distance and the trajectories in light blue were calculated using the proposed metric.

**Average Distance Metric**

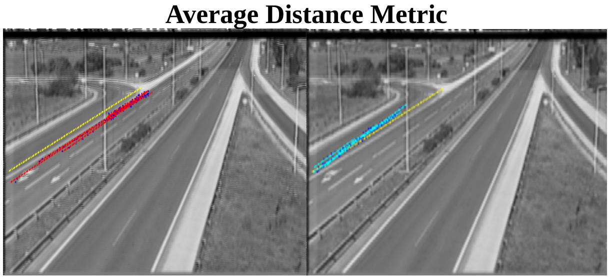

*Fig. 7. a) Hausdorff metric          b) Proposed metric*

**Minimum Distance Metric**

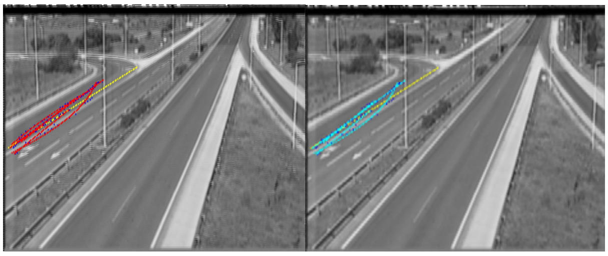

*Fig. 8.    a) Hausdorff metric          b) Proposed metric*

The colors have the same meaning as in the previous figures.

**Maximum Distance Metric**

*Fig. 9.    a) Hausdorff metric   b) Proposed metric*

We can see that the Hausdorff Maximum and Average distance are more sensitive to the trajectories length difference than the proposed distance. The average and the maximum Hausdorff distance return the trajectories that have equal and smaller lengths than the trajectory query, Fig. 7.a e Fig. 9.a.

For the average distance, the RMS and the distance integral return very similar results and are not shown.

The corrected Minimum Distance gives the same result as the Hausdorff distance but better than the original Minimum Distance because tests if the trajectories intersect Fig. 8.

For all the situations, our proposed metrics $dc_{\text{mean}}$ and $dc_{\text{min}}$ gives equal or better results than the Hausdorff distance or metric $d$. The Maximum Distance and the Average Distance usually return similar results. The only difference is that the Maximum Distance discards the trajectories which have any point more distant to the query than the threshold.

**Application: Real-time detection of Dangerous Lane Changes**

*Fig. 10. Trajectory retrieval*

The goal is to retrieve the trajectories that correspond to vehicles that crossed the straight line present on the road. In Fig. 10, we query the trajectories database using the yellow polynomial as a trajectory query. The 4 trajectories in light blue correspond to the trajectories which have a Minimum Distance, $dc_{\min}$, less than 0.5 pixels to the trajectory query.

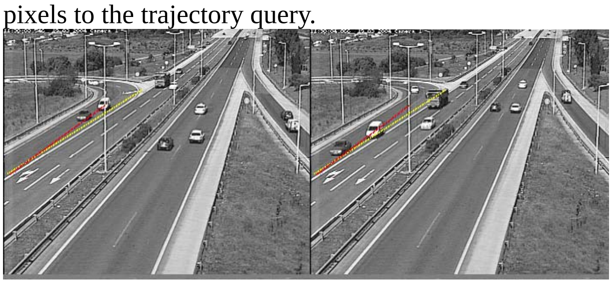

*Fig. 11. Retrieved trajectory a) Beginning b) Middle*

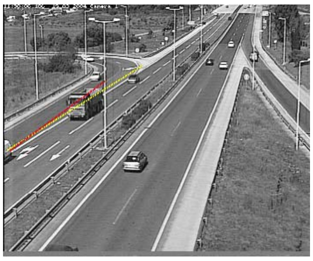

*Fig. 11. Retrieved trajectory c) End*

By assigning a video index frame to all the trajectories, it is possible to check the ground-truth of the retrieved trajectories.

In fig. 11, we can see one of detected dangerous trajectories represented in red, with the vehicle at the beginning, middle and end of the trajectory.

The trajectories querying resulted in 2 corrected trajectories match, 0 false negatives and 2 false positives.

The false positives are mainly due to two factors: the height of the detected vehicles is not corrected and causes that truck or vans to be misdetected. The other factor is that the PTMS algorithm sometimes misclassifies platoons of vehicles as single vehicle which causes that the centroid of the platoon will be misdetected as well.

The absence of false negatives is very important because it means that there are no undetected vehicle trajectories.

### TRAJECTORIES CLUSTERING

We show some preliminary results of applying the clustering approach to the computed trajectories described in section 4. The computed point trajectories of single vehicles (Fig. 6a) are used to estimate the lane centres (Fig. 12a) on the highway. From a total of 237 partial trajectories in the image sequence, the K-means clustering algorithm uses a maximum of 30 trajectories per lane to estimate the centres. In Fig. 12.a can be observed the number of trajectories added to each initial cluster.

It should be noted that although the original trajectory data contains vehicle lane changes, the RANSAC fitting method can be made insensitive to these by careful parameter tuning. In Fig. 12.b and Fig. 13, the cluster algorithm is applied to different sequences

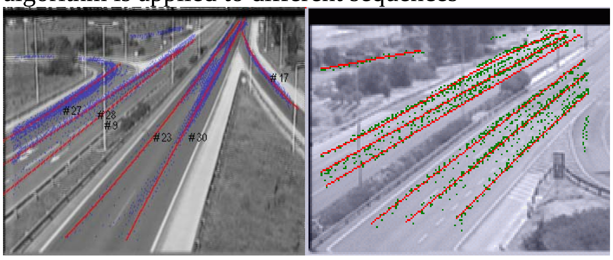

*Fig.12. a) Lane Estimation b) Lane Estimation*

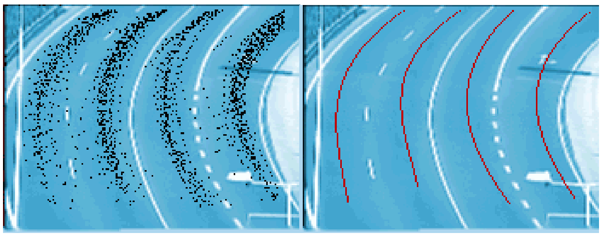

*Fig.13. a) Original Points b) Lane Estimation*

## 6. DISCUSSION AND CONCLUSIONS

This paper presents an algorithm for vehicle tracking with the following characteristics; temporal integration with a Kalman Filter, time-consistent merging-and-splitting of overlapped detected blobs, aggregation of trajectory data to estimate lane centres and trajectories query.

Our proposed distance metrics show equal or better results than the Hausdorff distance in the trajectory retrieval, with the advantage of being much cheaper to compute. In the context of highways, the proposed metrics able the use of real-time trajectory query and real-time trajectory clustering.

The preliminary results demonstrate the feasibility of using ordinary uncalibrated stationary or PTZ cameras to analyse traffic behaviour 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 behaviour.

More results can be found at the author's webpage[2].


## ACKNOWLEDGEMENTS

José Melo is partially funded by the Portuguese project ADI-INTELTRAF. The authors would also like to thank the Instituto Superior Técnico (IST), Portugal for providing the highway video sequences used in this report.


## 7. REFERENCES


[1] Gutchess D., Trajkovics M., Cohen-Solal E., Lyons D., Jain A.K, A Background Model Initialization Algorithm for Video Surveillance, in *proc. IEEE ICCV* 2001, Pt.1 (2001) 744-740.

[2] O. Javed, K. Shafique, M. Shah, A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information, in *Proc. Workshop on Motion and Video Computing*, (2002) 22-27.

[3] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Real-Time Surveillance of People and Their Activities. *IEEE Trans. Patt. Anal. Mach. Intell.* 22 (2000) 809-830.

[4] Prati, I. Mikié, C. Grana, M. Trivedi , Shadow Detection Algorithms for Traffic Flow Analysis: a Comparative Study, *IEEE Trans. Intell. Transport. Syst.* (2001) 340-345.

[5] P. KaewTraKulPong, R. Bowden, An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection, in *proc. 2nd European Workshop on Advanced Video Based Surveillance Systems,* AVBS01 (2001).

[6] A. Elgammal, R. Duraiswami, Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance, *Proc. IEEE*, 90 (2002).

[7] R. Cuchiara, C. Grana, M. Piccardi, A. Prati, Detecting Objects, Shadows and Ghosts in Video Streams By Exploiting Colour and Motion Information *Proceedings of 11th International Conference on Image Analysis and Processing ICIAP* (2001)

[8] Koller,D.; Weber,J.; Malik.J.: Robust Multiple Car Tracking with Occlusion Reasoning, *Proc.Third European Conference on Computer Vision*, LNCS 800, Springer-Verlag (1994).

[9] H. Veeraraghavan, O. Masoud, N. Papanikolopoulos: Computer Vision Algorithms for Intersection Monitoring, *IEEE Trans. Intell. Transport. Syst*. 4 (2003) 78-89.

[10] Y.Jung, K. Lee, Y. Ho, Content-Based event retrieval using semantic Scene interpretation for automated traffic surveillance, *IEEE Trans. Intell. Transport. Syst*. 2 (2001) 151-163.

[11] S. Kamijo, Y. Matsushita, K. Ikeuchi, M. Sakauchi Traffic Monitoring and Accident Detection at Intersections, *IEEE Trans. Intell. Transport. Syst.* 1 (2000).

[12] Daniel J. Dailey, F. W. Cathey, S. Pumrin: An Algorithm to Estimate Mean Traffic Speed Using Uncalibrated Cameras, *IEEE Trans. Intell. Transport. Syst.* 1 (2000)

[13] Todd N. Schoepflin, Daniel J. Dailey, Dynamic Camera Calibration of Roadside Traffic Management Cameras for Vehicle Speed Estimation, *IEEE Trans. Intell. Transport. Syst.* 4 (2003) 90-98.

[14] D. Makris, T. Ellis, Path Detection in Video Surveillance, *Image and Vision Computing*, 20 (2002) 895-903.

[15] C. Stauffer, W. Grimson, Learning Patterns of activity using real-time tracking, *IEEE Trans. Patt. Anal. Mach. Intell*. 22 (2000) 747-757.

[16] N. Johnson, D. Hogg, Learning the distribution of Object Trajectories for Event Recognition, *Image and Vision Computing*, 14 (1996) 609-15.

[17] A. Elgammal, R. Duraiswami, Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance, *Proc. IEEE*, 90, (2002).

[18] M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. of the ACM*, 24 (1981) 381-395.

[19] J. Melo, A. Naftel, A. Bernardino, J. Santos-Victor. Viewpoint Independent Detection of Vehicle Trajectories and Lane Geometry from Uncalibrated Traffic Surveillance Cameras, *accepted* ICIAR, 2004

[20] A. Naftel, J. Melo, A. Bernardino, J. Santos-Victor. Highway Lane Detection and Classification Using Vehicle Motion Trajectories, *submitted to* IASTED, 2004


[2] http://omni.isr.ist.utl.pt/~jpqm/inteltraf.htm