Applying graph theory to automatic vehicle tracking by remote sensing

*Carlos **Lima Azevedo**
National Laboratory for Civil Engineering
Department of Transportation
Av. Do Brasil, Lisbon, 1700-066
Portugal
Phone: +351 218443541
Fax: +351 218443029
Email: cmazevedo@lnec.pt

João Lourenço **Cardoso**
National Laboratory for Civil Engineering
Department of Transportation
Av. Do Brasil, Lisbon, 1700-066
Portugal
Phone: +351 218443661
Fax: +351 218443029
email: jpcardoso@lnec.pt

Moshe **Ben-Akiva**
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
United States
Phone: +16172535324
Fax: +16172531130
email: mba@mit.edu

* Corresponding Author

Word count: 5062 + 9 Figures + 1 Table = 7562 words
Submitted the 1st August 2013, revised 9th November 2013

Abstract

The estimation of driving behavior models relies on the access to detailed traffic information such as vehicle trajectories. Recent developments in vision-based technologies have allowed an increased collection of vehicle trajectories around the world, with an emphasis on aerial or high observation point imagery methods. Several computer algorithms have been proposed using images of different traffic scenarios with the specific aim of detecting and tracking road users.

Very recently, multiple-object tracking based on constrained flow optimization has been shown to produce very satisfactory results. Generally, this method uses individual image features collected for each candidate vehicle position as main criteria in the optimization process. Although these methods are very effective in controlled scenarios, adverse conditions such as dynamic view points and wider observation areas with low ground sampling distances are known to encumber significantly the vehicle trajectory extraction task.

In this paper we present the application of a k-shortest disjoint paths algorithm for multiple-object tracking using a motion-based optimization based on dual graphs. A graph of possible connections between successive candidate positions was built using speed and lane connectivity. Dual graphs were constructed to allow for acceleration and lane-change-based optimization criteria. The k-shortest disjoint paths algorithm was then used to determine the optimal set of trajectories (paths).

The proposed algorithm was successfully applied to the vehicle tracking in the A44 suburban motorway, in Portugal. Vehicle positions were detected by image processing and 99.4% of the trajectories were successfully and efficiently extracted using the proposed method.

INTRODUCTION

Vehicle trajectory data has been one of the most important sources of information in driving behavior modeling and calibration research (*1*,*2*,*3*). In recent years, the progress in sensing technologies and image processing algorithms allowed for easier collection of such detailed traffic datasets (*4*, *5*). Aerial imagery is the most common process for collecting the base data (*6*), and several vehicle detection and tracking algorithms have been deployed for different traffic scenarios (*7*). A key step in this process is the vehicle tracking step, where the time-independent vehicles detected in successive images, are linked together to reconstruct observed trajectories (see FIGURE 1).

One of the most common approaches relies on the segmented regions or contours properties identified in each frame and uses Kalman filters (*8*, *9*) to reconstruct motion tracks. Region-based tracking is computationally efficient and works well with short image view fields and free-flowing traffic (*10*). However, under congested traffic conditions, vehicles may partially occlude one another, making individual blob identification much more difficult. Feature-based tracking is another approach based on tracking of points which have a particular texture in their respective image positions. These interest points have been long used in the context of motion, stereo, and tracking problems. A desirable quality of an interest point is its invariance to changes in illumination and camera viewpoint. These points (features) are then grouped considering spatial proximity or similar motion patterns along the relevant multiple image frames. These algorithms have distinct advantages over other methods: they are robust to partial occlusions, they don't require initialization, and can adapt successfully and rapidly to variable lighting conditions, allowing real-time processing and tracking of multiple objects (*11*). However, special requirements have to be met as regards to camera calibration and objects with similar motions (*12*). Knowledge-based methods (*13*), which employ a prior knowledge to decide whether the identified object is a road user of interest, and optical flow based methods (a dense field of displacement vectors which defines the translation of each pixel in a region) have also been used (*14*). Each of these methods has, however, presented some weaknesses, such as frequent identity switches or non-simple tuning of its model parameters (*15*).

Graph theory has been recently applied to the vehicle tracking problem with success (*16*). Typically, every region in a frame is represented by a node in the graph. A link between each region in two consecutive frames is generated and labeled with a discrete variable representing the number of objects moving from linked nodes. Trajectories are then extracted using global optimization using a min-cost flow algorithm. Linear Programming can be used to link multiple detections over time, and therefore solve the graph problem (*17*). However, the computational complexity of the dynamic programming approach can be prohibitive when the frame or/and vehicle number is higher.

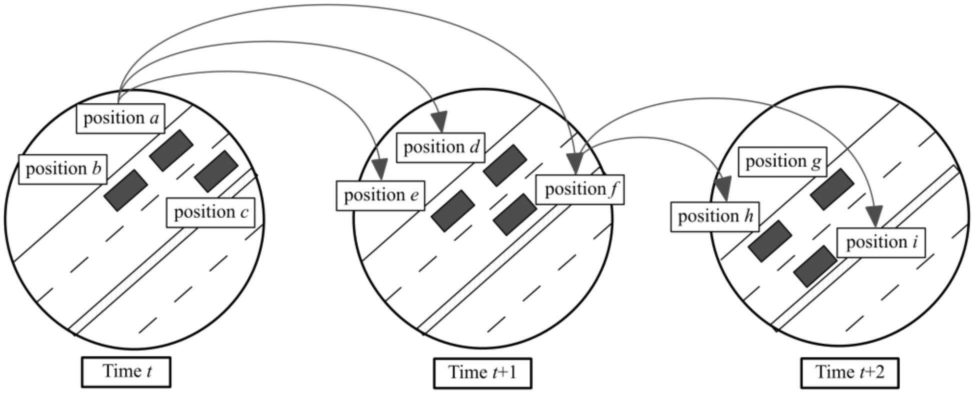


FIGURE 1 The tracking problem

In FIGURE 1 an example of how candidate positions are linked when constructing the graph is presented for a small image set. In each image, three vehicles are captured by the camera. As an example, the possible links from position a at time t and position f at time $t+1$ are shown with arrows.

Recently Berclaz et al. (*16*) reformulated the Linear Programming (LP) problem as a k-shortest disjoint paths problem on a directed acyclic graph. In their study, the area of interest in the image sequence and the time interval of the recording were discretized (as nodes) and linked considering possible object motions, resulting in a directed acyclic graph. Two additional nodes ($source$ and $sink$) were added to account for a consistent number of trajectories (flow) in the data set. These two nodes are linked to all the nodes representing positions through which objects can respectively enter or exit the observed area, such as occlusions or the camera field of view, and to all nodes in first and last image. Any path between the $source$ and the $sink$ nodes represent the flow of a single object in the original problem along the edges of the path, hence a trajectory. The node-disjointness constraint is needed to assure that no location can be shared between two paths (see FIGURE 2).

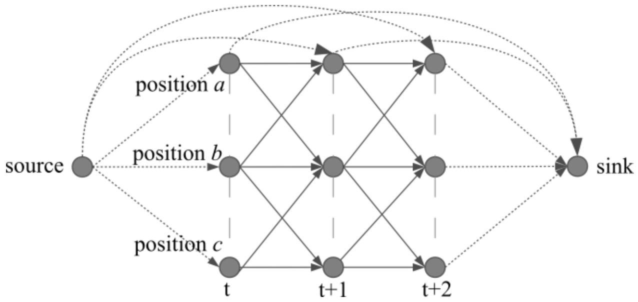


FIGURE 2 Generic multiple object tracking (adapted from (*16*))

In (*16*) the optimization function depended on the marginal posterior probability of the presence of an object in each image, which was obtained previously during the object detection task. When the extraction of object features from the detection algorithm is defective, either due to poor image quality or to low ground sampling distances, feature-based optimization may produce unreal trajectories, which is a known limitation that has to be addressed while extracting vehicle trajectories' data. To overcome these limitations an alternative approach using vehicle motion parameters as optimizing function is proposed.

THE PROPOSED FRAMEWORK

To account for different motion related criteria, the algorithm proposed by Berclaz et al. (*16*) was extended by integrating the use of dual graphs, based on the assumption that any driver has a motion-based optimizing function, i.e., that any trajectory is subject to on a set of motion-based objectives of the driver. Ideally, complex microscopic driving behavior models and Kalman-filter dynamics model may be used in this optimization process using large number of variables and parameters to reconstruct trajectories along with the k-shortest disjoint paths algorithm. Due to the specific nature of the current application, a simpler approach was considered. In free-flow conditions, it was assumed that a driver tends to reach and maintain its target speed; when relaxing the free-flow constrain, the driver tends to minimize changes in acceleration. These changes are even smaller if observations are more frequent, due to vehicle dynamics limitations. Regarding lateral movement, a similar approach can be formulated with the inclusion of lane change tags: when the lateral acceleration (regarding the lane axis) is constant and different from zero for a longer period of time, a lane change might be tagged.

In the next paragraphs the graph construction task is described and the k-shortest disjoint path algorithm is presented.

Graphs Construction

Similarly to the approach proposed by (*16*), our optimization problem was expressed as a graph problem. Instead of dividing the area of interest into possible locations that may or may not be occupied by a vehicle, a primal graph A was built linking acceptable candidate positions already detected in the vehicle detection step. However, dual graphs (*19*) were constructed representing different motion relationships (speeds and accelerations) between the linked positions. The dual graph of G is constructed such that each of its nodes correspond to an edge of G, and each edge to two neighboring edges of G.

Primal Graph

The primal graph represents all connections between acceptable candidate positions in successive images. This graph is constructed based on bounding limits for speed and lane connectivity.

Each detected vehicle position candidate $i^t \in K^t$, where $t \in T$ represents the image from the full image set T; and i^t, a node (candidate position) in the primal graph A. For any location i^t, let $N(i^t) \subset K^{t+1}$ denote the possible positions of i^t at the next observation time $t+1$. To model vehicle positions over time, let us consider a labeled directed graph with $\sum_t^{\mathrm{T}} |K^t|$ nodes, representing all candidate positions in the full image set. Its edges correspond to admissible vehicle motions between successive image shots. For i^t and j^{t+1} (denoted as i and j for simplicity) to be connected with an edge e_{ij}, its computed speed should satisfy eq. (1):

$$0 \leq V_{ij}^l = \frac{X_j^l - X_i^l}{\Delta t_{ij}} \leq V_{\max}^l \qquad (1)$$

where X_i^l and X_j^l, $l \in \{long, lat\}$ are the longitudinal and lateral vehicle positions relatively to the lane center line for the position candidates i and j, respectively. Eq. (1) is also used to assign the longitudinal edge costs $c_{ij}^l = V_{ij}^l$, where V_{ij}^l represents the longitudinal speed from two consecutive positions i and j. A lane change tag $c_{ij}^{lc} = \{0,1\}$, equal to 1 if $lane_i = lane_j$ and 0 otherwise, can also be computed for each edge e_{ij}. Lane connectivity was only used to reduce the size of the primal graph. Its information was extracted from existing geo-referenced lane axis data. Every detected position was projected to the nearest lane axis and linked positions associated with unfeasible lane changing maneuvers (e.g.: different directions) were filtered out.

Dual Graphs

After constructing the primal graph, accelerations might be computed from adjacent edge cost (speed) combination. These combinations produce a new cost for each pair of adjacent edges in the graph, similar to turn costs in route planning graphs. These new costs cannot be stored easily with the edges nor nodes of the primal graph, but they can be attached to a linear dual graph. This can be achieved using linear dual graphs, where edges in the original graph are replaced by nodes, and pairs of consecutive edges by edges (*19*). Given a primal directed weighted graph $A(N,E)$, the graph $B(\dot{N},\dot{E})$ with the following properties is called its complete linear dual graph:

- For each edge e_{ij} in A there is a node $\dot{n}_{ij} = d(e_{ij})$ in B. d is an objective function so that $d^{-1}(\dot{n}_{ij}) = e_{ij}$.
- For each pair of consecutive edges (e_{ij}, e_{jk}) in A, there is an edge $\dot{e}$ in B between the corresponding nodes $\dot{n}_{ij} = d(e_{ij})$ and $\dot{n}_{jk} = d(e_{jk})$.
- A cost function $f_{\dot{c}}: \dot{E} \longrightarrow R$.

The number of nodes in B equals the number of edges in A and the number of edges in B equals the number of connected edge pairs in A.

A first dual graph representing the accelerations, called $\dot{B}$, may be obtained by performing the above procedure once. A second dual graph representing variation of accelerations, called $\ddot{B}$, is obtained by performing a second iteration, using the following cost transformation functions:

1. Acceleration Dual Graph $\dot{B}$:

$$\dot{c}_{ijk}^{l} = a_{ijk}^{l} = \frac{V_{jk}^{l} - V_{ij}^{l}}{\Delta t_{jk} + \Delta t_{ij}} \qquad (2)$$

$$\dot{c}_{ijk}^{lc} = c_{ij}^{lc} \times c_{jk}^{lc} \qquad (3)$$

2. Acceleration Variation Dual Graph $\ddot{B}$:

$$\ddot{c}_{ijkm}^{l} = \Delta a_{ijkm}^{l} = a_{jkm}^{l} - a_{ijk}^{l} \qquad (4)$$

$$\ddot{c}_{ijkm}^{lc} = c_{ijk}^{lc} \times c_{jkm}^{lc} \qquad (5)$$

where, i, j, k and m are the node indexes in the primal graph A. These transformations are represented in FIGURE 3, where the primal graph A is represented in continuous grey lines, dual graph $\dot{B}$ by dashed grey lines and the final dual graph $\ddot{B}$ by bold dark nodes and edges.

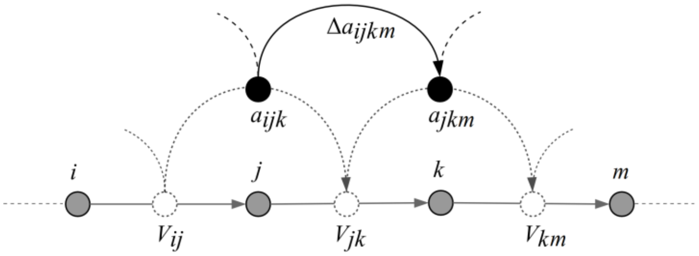


FIGURE 3 Dual graph construction

Additional acceleration-based criteria were used to filter out edges in the Acceleration Dual Graph. Using minimum and maximum longitudinal and lateral accelerations, all edges not satisfying eq. (6) were eliminated from $\dot{B}$:

$$a_{\min}^{l} \leq a_{ijk}^{l} \leq a_{\max}^{l} \qquad (6)$$

The majority of shortest path algorithms take as input a single edge cost value. To avoid the use of multi criteria optimal path problem, a cost function to integrate longitudinal and lateral vehicle movements must be specified. In our application, a simple linear optimizing function was considered. For any edge $\ddot{e}_{ijkm}$ (noted as b for simplicity) in the final dual graph $\ddot{B}$, its cost $\bar{c}_b$ was computed as:

$$\bar{c}_b = \omega^{l} \bar{c}_b^{l} + \omega^{lc} \bar{c}_b^{lc} \qquad (7)$$

where $\bar{c}_b^{l}$ is the value of $\ddot{c}_b^{l}$ normalized to $[0,1]$ and $\bar{c}_b^{lc}$ is equal to $(1 - \ddot{c}_b^{lc})$. ω^{l} and ω^{lc} represent therefore the weight of the longitudinal acceleration variation and a lane change factor. It is worth mention that this simplified approach is acceptable for non-saturated motorway traffic, but does not however, represent a true drivers' trajectory optimizing function valid for all traffic flow conditions. The lane change factor, for example, considers that a driver tends to stay in the same lane, underestimating the effect of strategical lane change in drivers' trajectory optimizing function.

The k-Shortest Disjoint Paths Algorithm

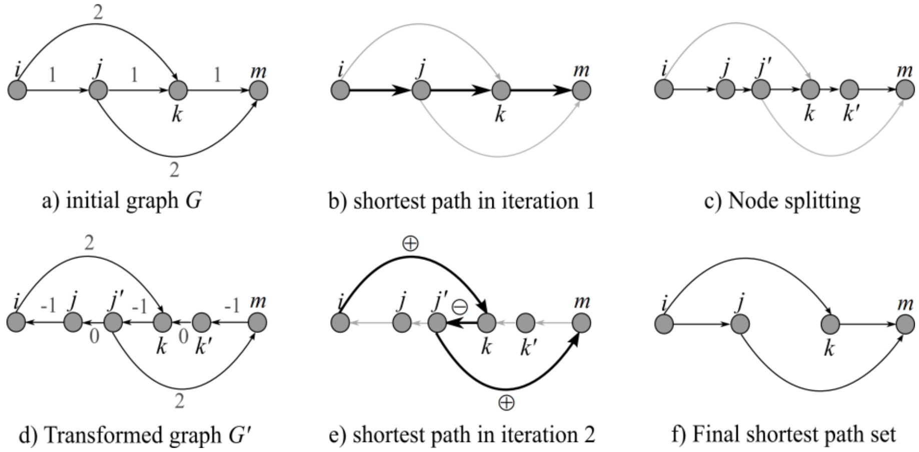
After the graph construction, an extension of the k-shortest disjoint paths algorithm proposed by Suurballe (*18*) was used to compute the best set of trajectories. Suurballe's algorithm relies on the iterative augmentation of signed paths and on a generic shortest path algorithm using modified costs. In this section a short description of the extension of the Suurballe's algorithm proposed in (*16*) is presented, and one should refer to both articles for further details.

Interlacing path and Augmentation

A signed path is a sequence of sign-labeled edges connecting them in order to form a path in a directed graph G, where each edge is assigned with a positive label ⊕ or a negative label ⊖. An interlacing path s, is a special type of signed path linked to a path set P_l, which satisfies the following two conditions:

- An edge is common to both s and P_l if and only if it has a negative label;
- A node is common to both s and P_l if and only if it is on an edge with negative label.

Both conditions are essential to achieve both edge and node-disjointness. The augmentation of P_l and s may be viewed as the addition and subtraction of labelled paths, where adding positive labeled edges of s to P_l and removing negative labeled edges of s from P_l. The augmentation process is illustrated in FIGURE 4 b) e) and f) for a simple graph.



a) initial graph G b) shortest path in iteration 1 c) Node splitting

d) Transformed graph G' e) shortest path in iteration 2 f) Final shortest path set

FIGURE 4 Suurballe's algorithm general framework

The path set obtained in b) composed by a single path $\{i, j, k, m\}$ is augmented by the path $\{i, k, j', m\}$ showed in e), resulting in the disjoint paths set $[\{i, k, m\}, \{i, j, m\}]$.

Graph Transformation

To account for signed paths and augmentation in the original graph G, Suurballe (*18*) proposed two transformations to a path p on G to obtain an interlacing path equivalent s:

- Node splitting: to account for the above-described two conditions, the node-disjointness criteria is relaxed to an arc-disjointness by node splitting: for each node i, introduce an auxiliary node i', reassign all outputs on i as outputs on i', leave all arc lengths unchanged and connect i and i' by an auxiliary link $e_{ii'}$ with cost $c_{ii'} = 0$ (see FIGURE 4).
- Path inversion: To account for signed labeling, the direction and algebraic sign of cost for each arc (and auxiliary arc) of p is inverted. This transformation represents a transformation from signed paths to directed unsigned paths.

The two step transformation is illustrated in FIGURE 4 for a simple path. In c), nodes j and k in path $\{i, j, k, m\}$ are split into j, j' and k, k' respectively. Finally, in d), $source$ and $sink$ nodes were not split to allow multiple flows (paths) from these two nodes. All edges direction in path $\{i, j, k, m\}$ were reverted and its cost signs inverted.

In short, the Suurballe's algorithm (allowing negative costs) performs the following steps:

1. Find the shortest path p_1 from $source$ to $sink$ in G using a generic shortest path algorithm.
2. Split every node i in p_1 and reverse the direction and algebraic sign of all edges in P_l, according to the previous section.
3. Find the shortest path p_2 in the transformed graph G_E using a generic shortest path algorithm.
4. Discard the reversed edges of p_2 from both p_1 and p_2. The remaining edges of p_1 and p_2 form a sub-graph with two edge-disjoint paths from $source$ to $sink$.

Cost Transformation

As the number of vehicles passing in the observed area is unknown, one also needs to optimize the number of paths k. Berclaz et al. (*16*) formulated the general optimizing problem by establishing an equivalence to the LP formulation. As discussed in their paper, the equivalence of the LP and the k-shortest paths formulation by Suurballe results from assuming a convex function of the path set total cost with respect to k. In fact, when assuming that path costs are monotonically increasing at each iteration n, $cost(p_n) \leq cost(p_{n+1})$, being p_n the shortest path computed at the n^{th} iteration of the algorithm, the total cost function for the full path set P_n at iteration n, $(costset(P_n) = \sum_1^n cost(p_n))$ is convex with respect to n. Therefore, the global minimum is reached when $cost(p_n)$ changes sign and becomes non-negative.

In our case study the following transformation of the already combined cost $\bar{c}_a$ (see eq. 7) for the acceleration variations and lane change tags was used:

$$cost_a = \log\left(\frac{\bar{c}_a}{1-\bar{c}_a}\right) \tag{8}$$

Doing so, the $cost(P_n)$ is concave with respect to n, and sets the stopping criterion of the algorithm to obtain the best P_k^* trajectories as:

$$cost\ (P_k^* - 1) \geq\ cost\ (P_k^*) \leq\ cost\ (P_k^* + 1) \tag{9}$$

General Framework

The general algorithm for vehicle tracking and trajectory extraction from the processed images is summarized in FIGURE 5.

THE CASE STUDY

The proposed approach was used in the collection of trajectories aiming at the fine calibration of an advanced micro-simulation tool for safety assessment. Vehicle positions were extracted from a sequence of aerial images using advanced computer vision techniques. The pilot area layout, systems' configuration and image processing algorithms are shortly presented in this section, but the reader should refer to (*23*) for further details.

System Configuration and Pilot Site Layout

Network of interest was the A44 road in the region of greater Porto, Portugal. It is a two-lane urban motorway with less than 5km and 5 main interchanges. It represents one of the main south entrances for the commuters living in the south-western region of greater Porto and to heavy vehicles heading to the main national port.

1. Construct the primal graph A.
2. Compute the dual acceleration and lane change graphs $(\dot{B}_a, \dot{B}_{lc})$.
3. Compute the dual acceleration variation and lane change graphs $(\ddot{B}_a, \ddot{B}_{lc})$.
4. Compute the transformed combined cost graph (B^{T}).
5. Iteration 1: Compute the shortest path p_1 on B^T using the Bellman-Ford algorithm (*21*, *22*).
6. Iteration n:
    a. compute the transformed graph B'^{T}_n using Suurballe's transformation steps (see FIGURE 4).
    b. compute the shortest path p_n on B'^{T}_n using the Bellman-Ford algorithm.
    c. compute the interlacing path s_n from p_n.
    d. compute the full path set P_n by augmentation of s_n on P_{n-1}.
    e. if $costset(P_{n-1}) \geq costset(P_n)$, then return P_{n-1}.

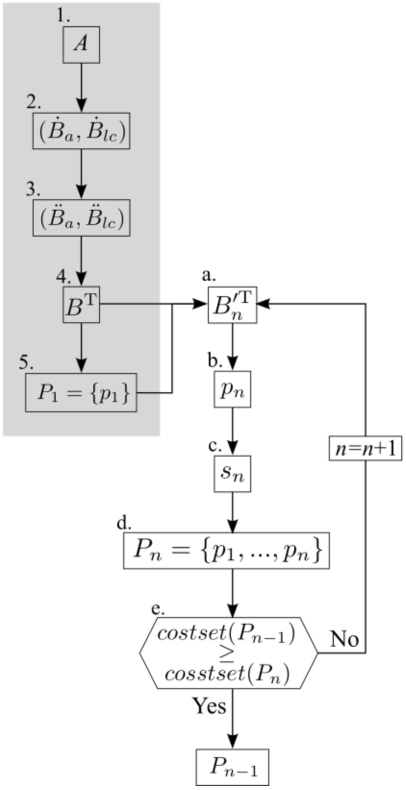

1.
A
2.
$(\dot{B}_a, \dot{B}_{lc})$
3.
$(\ddot{B}_a, \ddot{B}_{lc})$
4.
B^{T}
5.
$P_1 = \{p_1\}$
a.
B'^{T}_n
b.
p_n
c.
s_n
$n{=}n{+}1$
d.
$P_n = \{p_1, ..., p_n\}$
e.
$costset(P_{n-1}) \geq cosstset(P_n)$
No
Yes
P_{n-1}

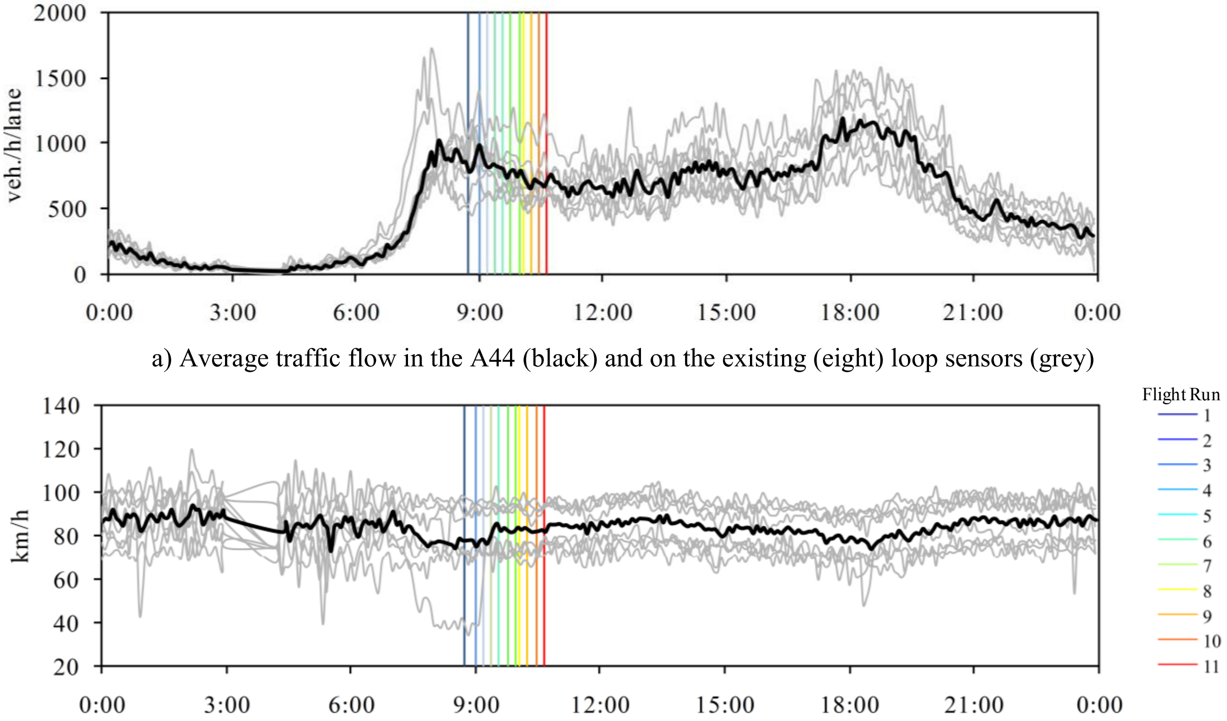
FIGURE 5 General framework

A light aircraft overflew the A44 eleven times (flight runs), between 8:45 and 10:45 AM (see FIGURE 6). Flight characteristics were optimized considering the atmospheric conditions on-site and a desired ground sample distance of 23cm. Images were orthorrectified using a 3D terrain model, the camera and lens characteristics and the precise flight positioning data recorded through differential GPS. Images were collected at an average rate of 0.5*Hz*, triggered by the fixed maximum image overlapping rate of 90%.

Vehicle Detection

The vehicle detection was carried out using colored background subtraction (see FIGURE 7). The reader should refer to (*23*) for the details on the vehicle detection algorithm:

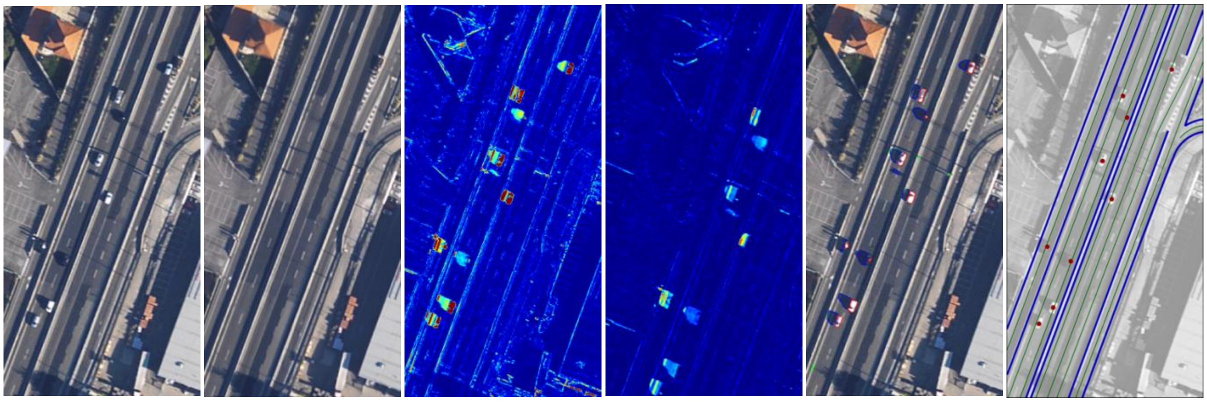
1. Images were locally rectified to minimize the terrain model and main orthorrectification errors. Each image was divided into grids, scaled and referenced automatically using the SIFT (Scale Invariant Feature Transform) method (*25*). The key points in successive images were then matched using the RANSAC (random sample consensus) algorithm (*24*).
2. For each flight run over the A44 a colored background was constructed using the median filter. A foreground of moving objects was extracted through background subtraction.
3. Shadows were filtered from the foreground using a spectral ratio technique and non-shadow moving pixels were used in a region-based analysis to extract blobs out of connected pixels and the vehicle candidate positions.



a) Average traffic flow in the A44 (black) and on the existing (eight) loop sensors (grey)

b) Average speed in the A44 motorway (black) and on the existing (eight) loop sensors (grey)

FIGURE 6 Loop sensor data collected on-site and flight run periods



a) rectified image b) background c) foreground d) spectral ratio e) blob analysis f) position extraction

FIGURE 7 Image processing steps (adapted from (*23*))

TRACKING RESULTS

In this section, we present the results of the cost function calibration and the statistics of the extracted trajectories for the A44 motorway case study.

Cost Function Calibration

The algorithm parameters ω^{l} and $\omega^{lc} = \left(1 - \omega^{l}\right)$ that represent the weights of the longitudinal acceleration variation and a lane change factor in the simple driver model used in this case study need to be tuned. Different combinations of weight pairs were tested against a manually reconstructed trajectory set with dense traffic situations. A set of specific measures of performance (MoP) were computed for both the manually extracted trajectories and those reconstructed by the proposed algorithm: mean (μ), standard deviation (σ), skewness (γ) and inter-quantile range (iqr) for speed, acceleration, headways, time-to-collision, lane gaps, etc. A few examples of the obtained root mean square percentage error (RMSPE) of a set of MoP are presented in TABLE 1. A description of this goodness of fit measures is presented in (*20*).

TABLE 1 RMSPE for different weights combination and MoP

$\omega^{lc} = (1 - \omega^{l})$	Detected trajectories (%)	Speed (%)				Headway (%)			Acceleration (%)		
		μ	σ	γ	iqr	μ	σ	γ	μ	σ	γ
0.500	116	0.32	6.95	38.4	7.55	0.42	0.37	27.9	5.37	8.33	11.2
0.750	53.8	0.16	3.68	68.8	2.46	0.21	0.52	1.93	2.68	6.37	3.47
0.800	38.2	0.15	3.26	77.1	2.53	0.19	0.42	1.98	2.77	7.02	3.80
0.850	25.3	0.15	2.88	12.4	2.50	0.16	0.01	0.10	2.07	5.96	3.20
0.900	14.0	0.15	2.68	10.5	2.70	0.14	0.01	0.08	1.85	4.68	1.46
0.925	6.99	0.15	2.65	10.7	2.45	0.13	0.02	0.08	1.90	4.82	2.17
***0.940**	**0.54**	**0.15**	**2.02**	**10.7**	**1.67**	**0.11**	**0.03**	**0.05**	**1.34**	**4.42**	**4.24**
0.950	6.45	0.25	1.18	3.99	0.6	0.45	0.22	30.5	1.78	1.85	2.30
0.960	12.4	0.22	1.18	4.11	0.77	0.42	0.50	30.5	1.55	1.46	2.37
0.975	22.6	0.26	2.00	4.40	1.33	0.45	0.27	29.9	2.07	4.00	4.25

The proposed method achieves very good results for higher weights of the longitudinal acceleration. However, the lane change also brings a non-negligible enhancement to the estimates of the mean (μ), standard deviation (σ), skewness (γ) and inter-quantile range (iqr) of longitudinal motion-based variables. ω^{l} and ω^{lc} were respectively set to 0.94 and 0.06 for the vehicle tracking of all flight runs.

Tracking Results

With the proposed method a total of 1855 trajectories for all flight runs were successfully collected. Levels of service remained between A and D, except for the weaving area in the South-North direction, where levels of service F and E were observed during the first two flight runs (see FIGURE 6 and the trajectory sample from flight run nº1 in FIGURE 8).

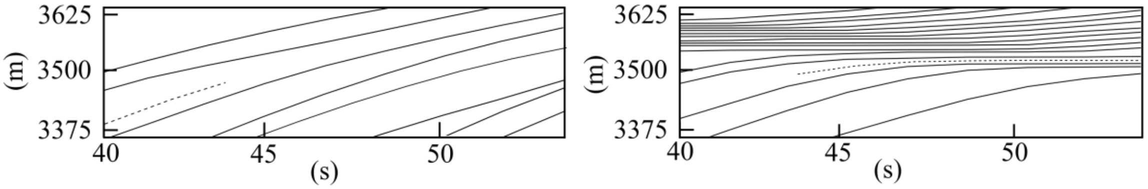
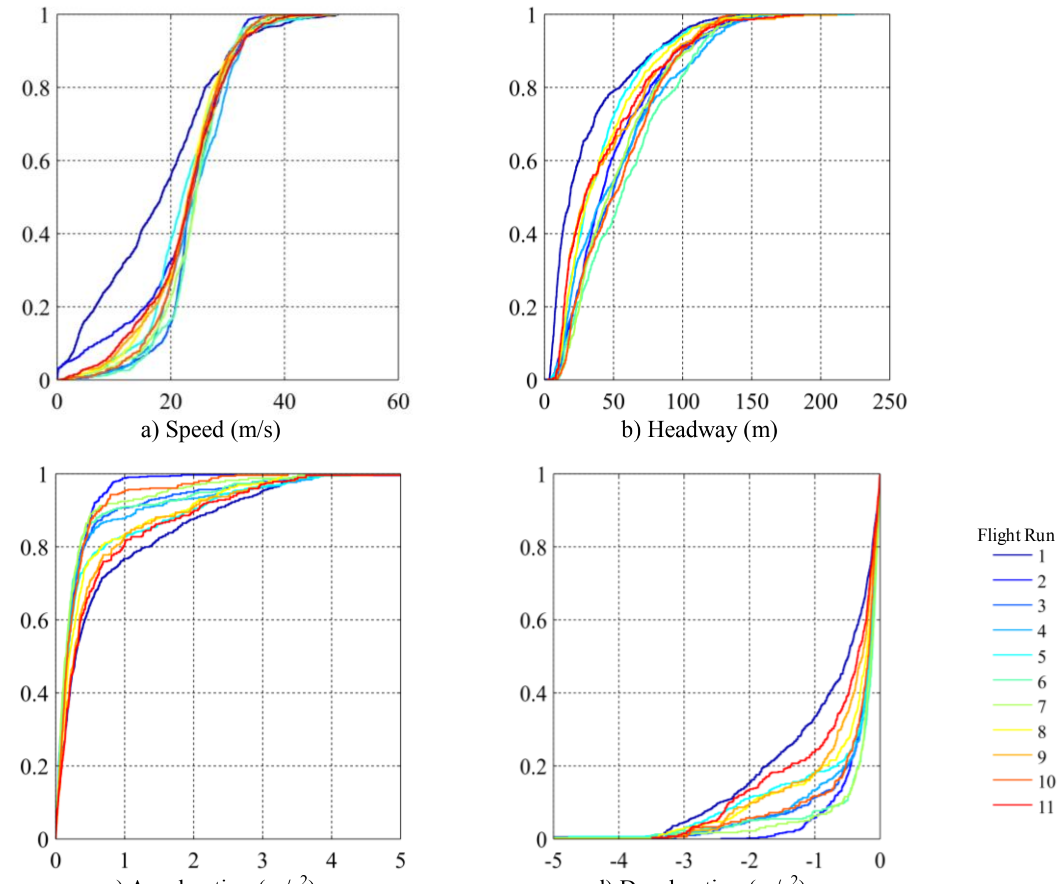


FIGURE 8 Sample of vehicle trajectories extracted using the proposed algorithm for the left and right lanes in the South-North direction of the A44 motorway

For the specific purpose of the simulator calibration, key traffic variables from smoothed (*26*) trajectories extracted using the proposed approach were computed. In FIGURE 9, the empirical cumulative distribution functions (CDF) for some of these variables are presented. Speed and headway may be approximated to a truncated normal distribution. It is worth noting that for the earliest flight run (flight run number 1), low speed values were still collected in some sections of the A44, resulting in a bimodal nature of its distribution (see FIGURE 9 a). Acceleration and deceleration follow a half-normal distribution with the typical low upper and lower range values for non aggressive manoeuvres.



a) Speed (m/s) b) Headway (m)

c) Acceleration (m/s^2) d) Deceleration (m/s^2)

FIGURE 9 CDF of motion variables for the different flights

CONCLUSIONS AND DISCUSSION

The trajectory reconstruction of independent object detection is a known difficult step in the road vehicle tracking problem. Recent developments in the application of graph-based algorithms to this specific problem have brought several computational and formal advantages against previous optimization solutions. By extending the k-shortest disjoint path algorithm to motion based optimization, we show in this paper that the integration of dual graphs in the general k-shortest path algorithm can provide a formulation free of image-based patterns and characteristics, by incorporating motion dynamics in the optimization problem. The proposed approach relies on the specification of a motion-based optimizing function, which can be easily modelled for different traffic scenes. A simple model based on longitudinal acceleration and lane change tags result in robust and fast results, even with images collected from a moving observation point and with low ground sampling distances.

Besides the algorithm parameters, the performance of the proposed method still depends on two key inputs: the vehicle detection results and the image shooting frequency. The vehicle detection used for the candidate positions generation (*23*) still lacks for an overall performance assessment for different lighting and weather conditions, and image acquisition system configurations. The influence of different image sampling rates on the optimization parameters should also be assessed in future works. Finally, it is worth noting that the original specification of the Suurballe algorithm applied to dual graphs may not always converge to the true optimal solution. In fact, if no dependence is created between graphs, the approach allows for node-joint paths in the primal graph for the final solution as all transformations and shortest path calculations are made using the dual graph. A possible solution is to use an Integer Programming (LP) formulation, as proposed by Berclaz et al (2011), instead of the graph-oriented formulation of Suurballe, ensuring that the constraint matrix exhibits a property known as total unimodularity, but at the expense of higher computer processing time, especially under dense traffic conditions. Future work will focus on additional enhancements to the proposed method to account for graph dependence, on different motion-based optimization criteria and on testing the algorithm to different image sets.

ACKNOWLEDGMENTS

Research contained within this paper benefited from the support of Prof. João Paulo Costeira, Dr. Manuel Marques and from the computational resources of their home institution: the Institute for Systems and Robotics of the *Instituto Superior Técnico*, University of Lisbon, Portugal. The authors would like to thank InfoPortugal S.A. for the collection of the raw aerial images.

REFERENCES

1 Hranac, R., R. Margiotta and V. Alexiadis. *NGSIM Task E.3: High-Level Data Plan*. Publication No. FHWA-HOP-06-011. Cambridge Systematic, Inc., Cambridge MA, USA, 2004.

2 Kesting, A. and M. Treiber. Calibrating Car-Following Models using Trajectory Data: Methodological Study. . In *Transportation Research Record: Journal of the Transportation Research Board*, No. 2088, Transportation Research Board of the National Academies, Washington, D.C., 2008, pp 148–156.

3 Punzo V., B. Ciuffo, and M. Montanino. Can results of car-following model calibration based on trajectory data be trusted? In *Transportation Research Record: Journal of the Transportation Research Board*, No. 2315, Transportation Research Board of the National Academies, Washington, D.C., 2012, pp. 11-24.

4 Laureshyn, A. *Application of automated video analysis to road user behaviour*. Ph.D. thesis, Lund University, Sweden, 2010.

5 Lenhart, D., S. Hinz, J. Leitloff and U. Stilla. Automatic traffic monitoring based on aerial image sequences. *Pattern Recognition and Image Analysis*, No. 3, Vol. 8, 2008, pp. 400–405.

6 Hoogendoorn, S. P., H. J. V. Zuylen, M. Schreuder, B. Gorte and G. Vosselman. Microscopic traffic data collection by remote sensing. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1855, Transportation Research Board of the National Academies, Washington, D.C., 2003, pp. 121–128.

7 Angel, A., M. Hickman, P. Mirchandani and D. Chandnani. Methods of Analyzing Traffic Imagery Collected From Aerial Platforms. *IEEE Transactions on intelligent transportation systems*, Volume 4, Issue 2, 2003, pp. 99-107.

8 Cheung, S. and R. Kamath. Robust background subtraction with foreground validation for urban traffic video. IS&T/SPIE's Symposium on Electronic Imaging. San Jose, CA, USA, 2004.

9 Yilmaz, A., O. Javed and M. Shah. Object tracking. *ACM Computing Surveys*, No. 38, Vol. 4, 2006, pp. 13.

10 Veeraraghavan, H., O. Masoud and N. Papanikolopoulos. Computer vision algorithms for intersection monitoring. *IEEE Transactions on ITS*, No. 4, Vol. 2, 2003,pp. 78–89.

11 Saunier, N. and T. Sayed. A feature-based tracking algorithm for vehicles in intersections. In *Proceedings of the 3rd Canadian Conference on Computer and Robot Vision*. IEEE, Quebec, Canada, 7-9 June, 2006.

12 Ismail, K. A. *Application of computer vision techniques for automated road safety analysis and traffic data collection*. Ph.D., UBC, 2010.

13 Bhattacharya, S., H. Idrees, I. Saleemi, S. Ali and M. Shah. Moving object detection and tracking in forward looking infra-red aerial imagery. *Machine Vision Beyond Visible Spectrum* McMillan, R. W. (Ed.), Springer, 2011, pp. 221–252.

14 Haag, M. and H. H. Nagel. Combination of edge element and optical flow estimates for 3D-model-based vehicle tracking in traffic image sequences. *International Journal of Computer Vision* No. 35, Vol 3, 1999, pp. 295–319.

15 Buch, N., S. Velastin and J. Orwell. A Review of Computer Vision Techniques for the Analysis of Urban Traffic, *IEEE Transactions on Intelligent Transportation Systems*, Volume 12, Issue 13, 2011, pp. 920-939.

16 Berclaz, J., F. Fleuret, E. Türetken, and P. Fua. Multiple Object Tracking using K-Shortest Paths Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Volume: 33, Issue: 9, 2011, pp. 1806-1819.

17 Song, X. and R Nevatia. Detection and Tracking of Moving Vehicles in Crowded Scenes. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, 2007, pp. 4.

18 Suurballe, J. W. Disjoint Paths in a Network. *Networks*, Volume 4, 1974, pp. 125–145.

19 Winter, S. and A.Grünbacher. Modeling Costs of Turns in Route Planning, *Geoinformatica*, Volume 6, Issue 4, 2002, pp. 345-361.

20 Hollander, Y. and R. Liu. The principles of calibrating traffic microsimulation models, *Transportation*, Vol. 35, No. 3, 2008, pp. 347-362.

21 Bellman, R. On a routing problem. *Quarterly of Applied Mathematics* , Vol. 16, Issue 1, 1958, pp. 87-90.

22 Ford, Jr. and R. Lester. *Network Flow Theory*, RAND Corporation, 1956, Paper P-923.

23 Lima Azevedo, C., J. L. Cardoso, M. Ben-Akiva, J. P. Costeira and M. Marques. Automatic vehicle trajectory extraction by aerial remote sensing. *Procedia - Social and Behavioral Sciences* 00: 16th Euro Working Group on Transportation, September, 2013. Porto, Portugal.

24 Fischler, M. A. 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.

25 Lowe, D., 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60 (2), 1981, pp. 91–110.

26 Toledo, T., H. Koutsopoulos and A. Kazi. Estimation of Vehicle Trajectories with Locally Weighted Regression. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1999, Transportation Research Board of the National Academies, Washington, D.C., 2007, pp. 161–169.