Automated video analysis as a tool for analysing road user behaviour

Laureshyn, Aliaksei; Ardö, Håkan

Link to publication

AUTOMATED VIDEO ANALYSIS AS A TOOL FOR ANALYSING ROAD USER BEHAVIOUR

Aliaksei Laureshyn

Lund University, Faculty of Engineering,
Department of Technology and Society, Traffic and Road
Box 118, 221 00, Lund, Sweden, tel. +46 46 222 91 31, fax +46 46 12 32 72
aliaksei.laureshyn@tft.lth.se

Håkan Ardö

Lund University, Faculty of Engineering,
Centre for Mathematical Sciences, Mathematical Imaging Group
Box 118, 221 00, Lund, Sweden, tel. +46 46 22 275 34, fax +46 46 22 240 10
hakan@debian.org

Abstract: At Lund University an automated video analysis system is being developed that can be applied for studying the behaviour of road users in complex traffic environments. It is stressed that system must be capable of handling all the categories of road users, i.e. vehicles, pedestrians and cyclists. Common problems like detection and tracking of moving objects, occlusion by foreground objects, ground-plane co-ordinates estimation, smoothing of the scattered data and estimation of speed and acceleration profiles are discussed and some solution proposed.

Keywords: Automated video analysis, road user behaviour, measurement method.

INTRODUCTION

Video recording is a commonly used tool for behavioural studies in road traffic ([18], [19], [1]). It provides many advantages, such as lower interference with the traffic processes than with a roadside observer, possibilities to make longer observation periods, store and review important situations, etc.

Since the manual processing of video data is very resource-demanding work, there is a high demand for automation of this task. Even though a number of systems capable of automated video records processing and retrieval some traffic-related data have been developed ([13]), most often their application area is limited to the relatively simple traffic conditions and performed tasks (reading number plates, congestion detection, etc. – [5], [7], [12]). More advanced systems ([15], [4]) are still very focused on vehicle detection and ignore other road user categories.

Currently at Lund University, Faculty of Engineering there is a system under development which is primarily aimed at studying the behaviour of road users in complex traffic

environments (e.g. congested urban conditions). The set goals for the fully developed system (a four-year project) are to be able to detect and follow all the categories of road users (including pedestrians and cyclists), identify and quantify their behaviour and analyse and interpret it in terms of safety and efficiency. After two years of project work the system is capable of detecting and tracking large-size objects (vehicles) with sufficiently high accuracy and detecting objects of smaller size.

This paper addresses some methodological problems, encountered in the system development work, and discusses proposed solutions which were implemented. These problems are certainly common in the field and have been to some extent faced by developers of other similar systems; however, the discussion found in the literature seems to be quite unsystematic, often omitting the details and thus being unclear. We consider thus that our attempt to gather these problems in one paper may be quite valuable and can be an opening for further discussions on a higher level.

For readability reasons we avoid using mathematical formulas in the text but refer instead to the special literature where more detailed description of the mentioned methods and procedures can be found.

ROAD USER DETECTION AND TRACKING

A classical solution for detection and tracking objects in a video sequence is to use foreground/background segmentation, which is a generalization of the background subtraction method. Several such methods exists and they all are based on the same principle of estimating the background and then deciding what parts of the image currently shows the background and what parts shows something else, the foreground. An example is given in Figure 1 where the algorithm [17], which uses a multimodal background model representing dynamic backgrounds such as trees swaying in the wind, has been tested.

Figure 1. Foreground/background segmentation. To the left is the input image and to the right is the result after segmentation with the algorithm described in [17].

Once the foreground/background segmentation is done, adjacent foreground pixels can be clustered together into objects and then objects overlapping between adjacent frames can be clustered together into tracks. This becomes problematic when objects in the scene are close to each other and overlap in the image. The problem of multi-object tracking have been

studied for a long time and many classical solutions exist such as Kahlman-filtering, JPDAF, HMM and particle-filtering. An overview can be found in the introduction of [11].

ROAD PLANE POSITION ESTIMATION

Many indicators, used to describe the road user's behaviour are actually based on measuring distance to other road users, physical objects or virtually calculated points in the road environment (e.g. lateral position within the lane, headway to the in-front vehicle, distance to a conflict point, etc.), which put high requirements on the accuracy of the position estimation.

Rectification

Measuring distances between objects viewed by a camera is not easy as there is no simple transformation from distance measures made in the image in pixels to real world distances in meters. The problem is that when a two dimensional image is produced from a real world scene with three dimensions, information is lost. In the general case it is not possible to measure distances without additional information. But with some prior knowledge about the scene it might be possible. Consider for example the case where the camera is placed high above the centre of the intersection looking straight down. If the intersection is flat an image produced by such a camera would be very similar to a map in which case distances in meters can be calculated from distances in pixel by a simple scaling.

In practice it is very hard to place a camera straight above the centre of an intersection. A much easier situation would be to mount the camera on top of a nearby building. In that case it will still be possible to measure distances between points on the ground plane, presuming the pavement is at least approximately a plane. If the camera is mounted significantly higher than the height of all the road users they could be approximated as flat objects and it is possible to transform the images from the camera into images looking approximately as if they were produced by a camera looking straight down at the intersection. This transformation is called **rectification**.

In order to perform the rectification the relationship between the ground plane and the image plane of the camera has to be estimated. This gives a set of parameters, a homography, that are used in the rectification process. Methods for estimating two dimensional homographies can be found in [9]. In this case it can for example be performed by manually measuring all the six distances between four points in the intersection and marking those four points in the image. Typically the points should be in the far corners of the intersection to give the best numerical stability and thus suppress measurement noise as much as possible. An alternative method would be to wait until an object of know dimensions, such as for example a bus, passes the camera view and use the corners of that object for the calibration. By tracking vehicles and utilising the fact that most vehicles passing through the intersection will not change the physical dimensions it is even possible to fully automate the rectification process. This is investigated in [2].

The results of the rectification process are demonstrated in
Figure 2.

Figure 2. Rectification. To the left is the original image as produced by the camera and to the right is the same image rectified. Note that the road plane is rectified nicely while objects above the road get more and more distorted the higher they appear.

Stereo vision

Rectified image allows measuring distance between the objects on the road plane, but the problem is that some parts of an object are elevated above the road and the distances between them estimated from the image are not accurate. To know which distances are measured correctly one needs to restore some lost information about the spatial relations between the object parts. Without any special correction, the image processing algorithm assumes that all the parts are in the same plane (see Figure 3), which under some unfavourable (but quite realistic) conditions can give an error up to several meters in position estimation.

ABCD - the real objects' shape;

AD - the real ground position;
A'D - the ground position assumed by the algorythm (no corrections applied)
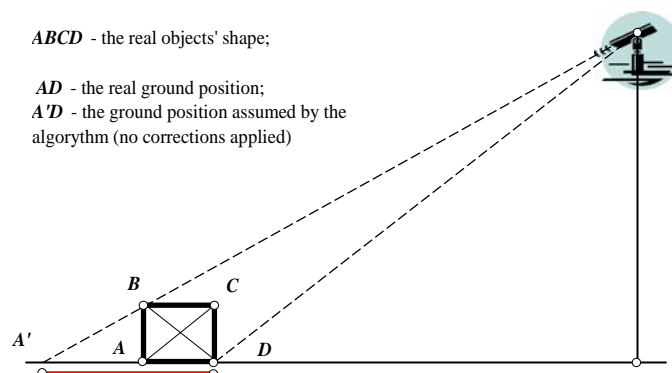
Figure 3. 2D-3D problem.

If the vehicle shapes were known that information could be used to give a better position estimate. Unfortunately, there are a lot of different vehicles and building a 3D model for each

of them is not trivial. Then the problem of identifying which of all the models a certain detection belongs to has to be solved, and as the number of models grows this problem will become more and more ambiguous. This approach is investigated in [8] and might be plausible for vehicles, but when it comes to pedestrians and cyclists that continuously change the shapes, this would require advanced dynamic models capturing this variability. It does not seem plausible to achieve all that in real time using reasonable amounts of today's hardware.

By using stereo vision (two cameras that are synchronized, i.e. exposing their pictures at exactly the same time) 3D-information can be extracted. This is done by identifying the same world point in the two images and then triangulating its 3D-coordinates from the 2D-coordinates in the two images. For this to work the geometry (relative position and rotation) of the two cameras has to be calibrated. This can be done by selecting a set of distinctive points spread out over one of the images and finding their corresponding positions in the other image.

Consider a point in the first image. To locate its 3D-world coordinates the same point has to be found in the second image. But it is not necessary to search the entire second image. The position of the point in the first image defines a 3D-world line along which the world point must be located, and since the cameras are calibrated this line is known. The image of this 3D-line in the second camera will be a 2D-line in the second image. This is called an epipolar line and it is only necessary to search for the point along this line. What's even better is that all points on an epipolar line in the second image correspond to a single epipolar line in the first image. That means that there is a one to one correspondence between epipolar lines in the first camera and epipolar lines in the second camera. All the points along an epipolar line in the first image will thus be found along the corresponding epipolar line in the second camera. This means that the 2D-problem of finding the points in the second image corresponding to all the points in the first image, can be reduced to several 1D-problems of finding the points along an epipolar line in the second image corresponding to all the points along an epipolar line of the first image.

It is very hard to find the correct match for every pixel. Typically some part of the images contains uniformly coloured regions with no texture, such as for example the roof of a car. As all pixels in those regions have almost the same colour, they match equally well to any pixel in the region. A classical solution here is to first ignore those uniformly coloured regions and only work on edges and then in a second step when the 3D-location of the edge points is known try to fill in the rest. To find these edge-pixels a 1D-sub-pixel edge detector ([3]) can be applied along each epipolar line and for every edge-point found a small patch centred on the detected edge-point can be extracted to be matched against the patches extracted from the edge-points on the corresponding epipolar line in the other image.

In order to measure distances in meters the calibration described above is not enough. It does allow for projective 3D-coordinates to be calculated, but they are only defined up to some unknown projective transformation. To estimate Euclidian 3D-coordinates a metric calibration has to be preformed. In [9] several methods for this are presented. Now it is no longer enough with four known points in the road plane. At least seven points are needed and they may not all be coplanar. In practise significantly more points are needed to give numerical stability and they need to be spread over the entire image and be located on different heights.

Recently services such as Google Earth and national equivalents have started to provide high resolution aerial images covering a significant part of the globe. By using such images of the intersection under study the calibration can be performed using manually selected point correspondences between the camera images and the aerial image. The aerial image gives two out of three world coordinates, and as the last one, the height, is zero for the points on the road plane, all three coordinates are available for these points. An affine calibration of the cameras can be found by treating the aerial image as an image from an affine camera looking straight down and using the method described in Section 9.4 of [9]. Then, if the principal point and the skew of the camera are known, the focal length can be estimated from the points on the ground plane, as all three world coordinates of these are known, using bundle adjustment ([9]). Finally the full camera calibration and the unknown point heights can be found also using bundle adjustment, with the affine camera and all intrinsic parameters (principal point, skew and focal length) fixed.

The road plane is easily located by manually selecting a few points on it, and then all points located less than some threshold from the road plane could be identified as belonging to the road and the rest of the points represent the objects in the scene. This includes vehicles, pedestrians and cycles as well as traffic lights and lampposts.
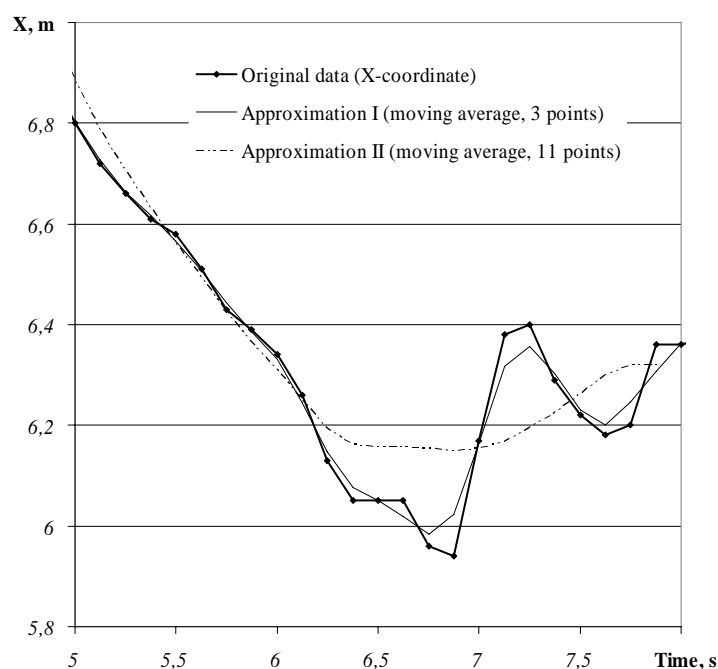

Figure 4. The effects of the chosen bandwidth on the resulting profile. Note that Approximation I follows the character of the original data, but still is too scattered. Approximation II is very smooth, but loses extreme values.

SCATTERED DATA SMOOTHING

Due to some stochastic errors, the raw position measurements from the video are always scattered around the "true" trajectory, which requires application of some smoothing procedures before the data can be practically used. However, the smoothing quality depends

on how the used algorithm suits the data. If it is too "strict", there is a risk to loose important profile features, such as presence or exact position of the local extremes or inflection points. On the other hand, if the algorithm is too "forgiving", one might get false extreme indications that have merely stochastic character.

Simple smoothing techniques, such as moving average, usually use a fixed bandwidth (interval or amount of points used to calculate the smoothed value) over the whole dataset. However, some parts of the dataset might differ from the other in character and thus it is hard to choose the bandwidth optimal for the entire dataset. Figure 4 gives an example on how the estimated data points depend on the chosen bandwidth.

Another problem, arising due to inaccuracy of the raw data, is the overestimation of the derivative values. As an example, let us consider calculation of speed from the trajectory data. The distance between two measured points is systematically biased towards longer distances, which results in speed overestimation (see Figure 5).
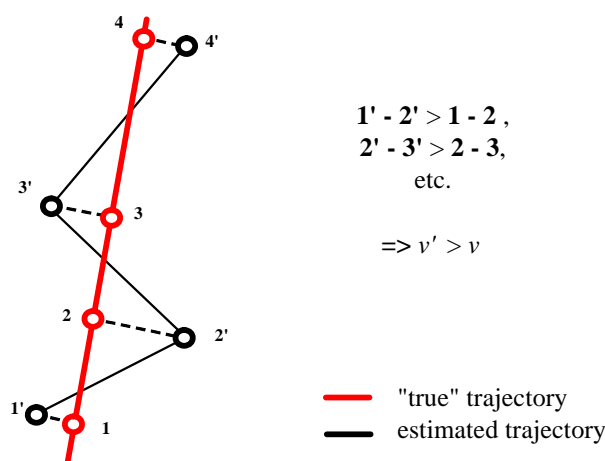

Figure 5. Overestimation of the derivative from the scattered data.

One possible solution is to first smooth the trajectory line and then calculate the speed based on the travelled distances along this line ([1]). However, trajectory smoothing does not completely remove the error in speed estimation. Calculation of the second or third derivatives (acceleration and jerk) thus still remains very inaccurate.

To overcome these problems we used local kernel polynomial regression algorithm ([6]). One advantage of this method is that the size of the bandwidth is an adjustable parameter, calculated dynamically depending on the data character in the current region. The other advantage is that the method allows separate estimation of the derivatives directly from the raw data.

This method is also used for smoothing the raw speed measurements obtained directly from the video data by using sub-pixel correlation, which is described in the following section. An example of the data proceed with the local kernel polynomial regression algorithm is presented in Figure 6.
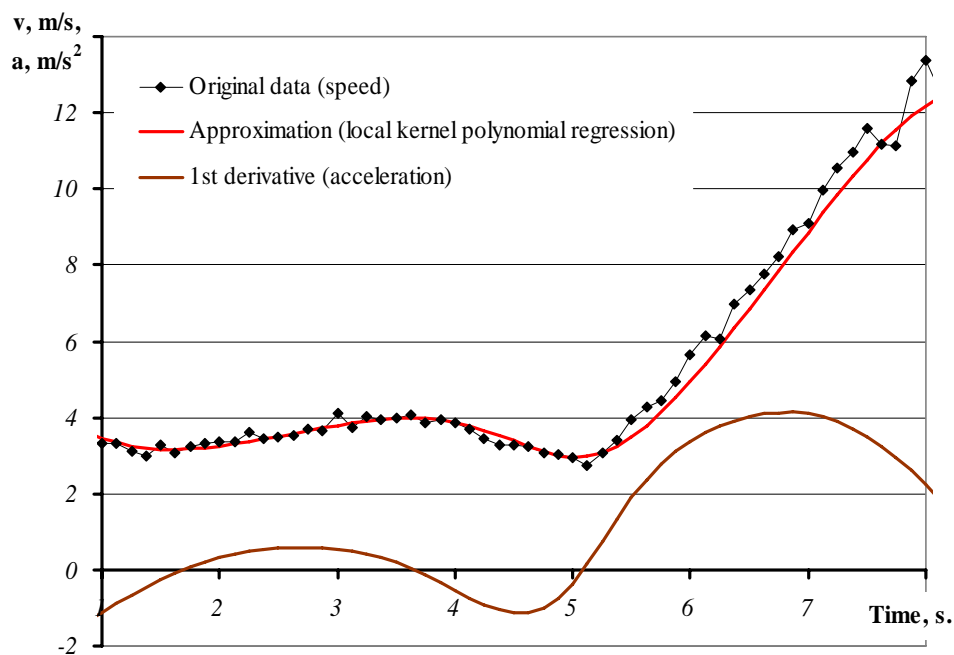
Figure 6. Data processed with local kernel polynomial regression algorithm.

DIRECT SPEED ESTIMATION

Speed and acceleration of road users are the parameters, describing the dynamics of a traffic situation. Detailed speed trace (speed profile) of a road user can provide important information such as clearly indicate when braking or swerving occurs, as well as can be used together with position data for calculation of other important traffic parameters (time gaps, collision point position, TTC – Time-to-Collision, etc.). The braking intensity can serve as an indicator for a conflict situation and a measure of its severity ([10], [14]); moreover, the derivative of the acceleration (jerk) was found to be an even better indicator for conflict situations then the hard braking ([16]).

Figure 7. Two adjacent frames used to estimate car velocity.

We have discussed the accuracy problems when estimating the speed values from the raw position data. There is thus a potential to improve speed estimation quality if speed measurements can be done directly from the video data, for example, by using displacement

between two adjacent frames. The problem is that this displacement is very small (typically only 1-5 pixels, see Figure 7) so the estimate has to be done with sub-pixel precision.

To do that the image is interpreted as a sampled version of a two dimensional function. According to the sampling theorem it is possible to reconstruct the original function as it was before the sampling by using sinc-interpolation between the sample points (pixels). This assumes that the original function contains no frequencies higher than half the sampling frequency, which is usually guaranteed by the optical properties of the camera lens.

This means that it is possible to calculate the intensity values for positions in the image between the pixels and thus it is possible to extract patches from the image located at sub-pixel positions. Such patches can be extracted around the position indicated by the tracker in the adjacent frames and then be compared. By also looking at the derivatives of those patches it is possible to deduce in what direction the patch should be moved in order to find a better match. Thus the displacement can be found using standard optimization techniques such as Newton-Rapson ([3]).

OBSTACLES BETWEEN CAMERA AND SCENE

Another problem is that there might be obstacles between the camera and the road user. In the case shown in Figure 8 it is the lamp of a lamppost and the system measures the speed of the lamp instead of the speed of the car as the lamp is closer to the camera. To solve this, the background/foreground segmentation data can be used to mask out the pixels that belong to the moving objects and remove all pixels belonging to the static background (including the lamppost in front of the car). Then the sub-pixel correlation described above only considers the pixels belonging to the objects and a better speed estimate is achieved.
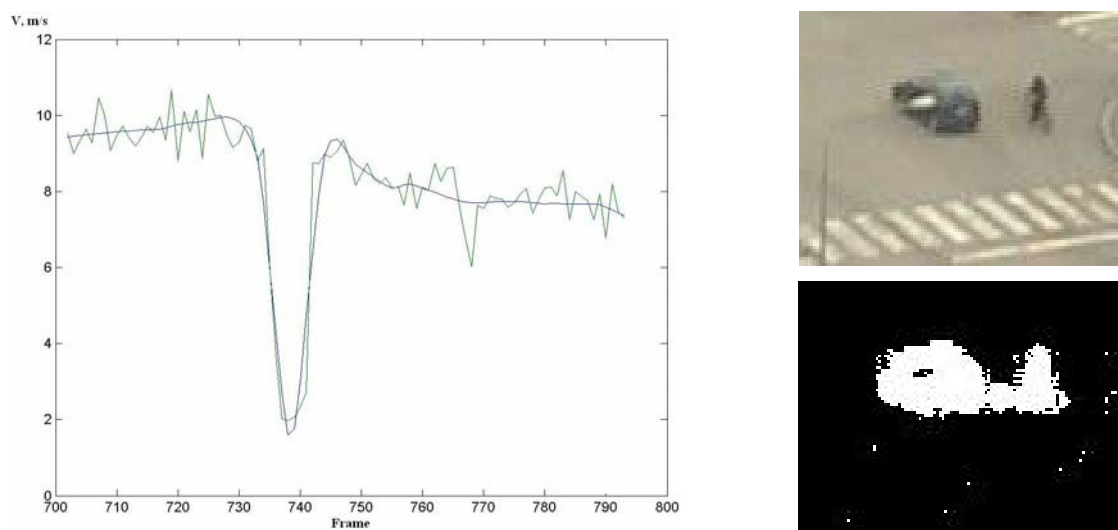

Figure 8. Effects of an obstacle on speed estimation. The diagram to the left is a speed profile of a car passing the intersection with fairly constant speed. The images to the right are the frame 737 and the corresponding foreground mask. The estimated speed drops significantly as the car passed beneath the lamp if the mask is not used.

CONCLUSIONS

The paper presents some typical problems faced in applying automated video analysis techniques for traffic behaviour analysis purposes and proposes some solutions to them. The intention is not to advocate for some "best" method or solution, but rather to create a research framework, report on experiences gained during our first years of system development and define the problematic areas which have to be further studied and filled in.

The practical work experience with automated video analysis system at Lund University shows a great potential of the tool. The observations are already continuously performed over several months, which was hardly feasible before in any behavioural study. Some part of the original video data is stored and can be used for other studies that were not even planned when observations started. The continuous trajectory and speed data, provided by the system, open a new dimension in describing the road user behaviour, which might affect the formulation of the future research questions as well as the choice of tools and methods used for such data analysis.

The future work should include detection and tracking accuracy improvement for objects of different sizes and shapes, increase robustness of the used procedures under less favourable work conditions (jammed traffic, twilight or night conditions, precipitations), as well as development of the analytical part for the further data analysis and interpretation of the results in terms of safety and the efficiency of the observed traffic system.

REFERENCES

[1] Andersson, J. Image processing for analysis of road user behaviour – a trajectory based solution. Lund Institute of Technology, Department of Technology and Society, Traffic Engineering, Bulletin 212, 2000.

[2] Ardö, H. Learning based system for detection and tracking of vehicles. Department of Mathematics, Lund University. 14th Scandinavian Conference on Image Analysis, 2005.

[3] Åström, K., A. Heyden. Stochastic Analysis of Image Acquisition, Interpolation and Scale-space Smoothing, Department of Mathematics, Lund University. Advances in Applied Probability 31 (4), 1999, pp. 855-894.

[4] Atev S., H. Arumugam, O. Masoud, R. Janardan, N. P. Papanikolopoulos. A Vision-Based Approach to Collision Prediction at Traffic Intersections. IEEE Transactions on Intelligent Transportation Systems 6, No. 4, 2005, pp. 416-423.

[5] Blythe, P. T. Congestion charging: Technical options for the delivery of future UK policy. Transportation Research Part A 39, 2005, pp. 571-587.

[6] Bratt, H., E. Ericsson. Estimating speed and acceleration profiles from measured data. In "Transport and air pollution: 8th international symposium: including COST 319 final conference. Graz, Austria 31 May – 2 June 1999". Technical University Graz, Institute for Internal Combustion Engines and Thermodynamics, Report 76.

[7] Coifman, B., D. Beymer, Ph. McLauchlan, J. Malik. A real time computer vision system for vehicle tracking and traffic surveillance. Transportation Research Part C 6, 1998, pp. 271-288.

[8] D. Koller, K. Daniilidis, T. Thorhallson, and H.-H. Nagel. Model-based object tracking in traffic scenes. In Proceedings of ECCV '92. Springer-Verlag, 1992.

[9] Hartley R., A. Zisserman. Multiple View Geometry in computer vision. Cambridge University Press, 2000.

[10] Hydén, C. Traffic Conflicts Technique: State-of-the-art. In: Topp H. H. (Ed.) Traffic Safety Work with Video-Processing. University Kaiserlauten. Transportation Department, Green Series No 43, 1996.

[11] Isard, M. A. Visual Motion Analysis by Probabilistic Propagation of Conditional Density. Robotics Research Group, Department of Engineering Science, University of Oxford, 1998.

[12] Ji, X., Zh. Wei, Y. Feng. Effective vehicle detection technique for traffic surveillance systems. Journal of Visual Communication and Image Representation 17, 2005, pp. 647-658.

[13] Kastrinaki, V., M. Zervakis, K. Kalaitzakis. A survey of video processing techniques for traffic applications. Image and Vision Computing 21, 2003, pp. 359-381.

[14] Malkhamah, S., M. Tight, F. Montgomery. The development of an automatic method of safety monitoring at Pelican crossings. Accident Analysis and Prevention 37, 2005, pp. 938-946.

[15] Messelodi S., C. M. Modena. A Computer Vision System for Traffic Accident Risk Measurement: A Case Study. ITC-irst Technical Report T05-06-07, 2005.

[16] Nygård, M. A method for Analysing Traffic Safety with help of Speed Profiles. Master's thesis. Department of Civil Engineering, Tampere University of Technology, 1999.

[17] Stauffer, C., W. E. L. Grimson. Adaptive Background Mixture Models for Real-time Tracking, CVPR99, Vol. II, 1999, pp. 246-252.

[18] Summala, H., E. Pasanen, M. Räsänen, J. Sievänen. Bicycle accidents and drivers' visual search at left and right turns. Accident Analysis and Prevention 28, No. 2, 1996, pp. 147-153.

[19] Svensson, A. A method for analysing the traffic process in a safety perspective. Department of Traffic Planning and Engineering, Lund Institute of Technology, Lund University, Bulletin 166, 1998.