Third Croatian Computer Vision Workshop

September 16, 2014, Zagreb, Croatia

CCVW 2014

University of Zagreb
Center of Excellence
for Computer Vision

PROCEEDINGS OF

Proceedings of the Croatian Computer Vision Workshop CCVW 2014, Year 2 ISSN 1849-1227
Publisher: University of Zagreb Faculty of Electrical Engineering and Computing, Unska 3, HR-10000 Zagreb, Croatia
http://www.fer.unizg.hr/crv/proceedings

CCVW 2014

Proceedings of the Croatian Computer Vision Workshop

Zagreb, Croatia, September 16, 2014

S. Lončarić, M. Subašić (Eds.)

Organizing Institution

Center of Excellence for Computer Vision,
Faculty of Electrical Engineering and Computing,
University of Zagreb, Croatia

Technical Co-Sponsors

IEEE Croatia Section
IEEE Croatia Section Computer Society Chapter
IEEE Croatia Section Computational Intelligence Chapter
IEEE Croatia Section Signal Processing Society Chapter

Donators

PhotoPay Ltd.
Visor Ltd.

Proceedings of the Croatian Computer Vision Workshop
CCVW 2014, Year 2

Editor-in-chief

Sven Lončarić (sven.loncaric@fer.hr)
University of Zagreb Faculty of Electrical Engineering and Computing
Unska 3, HR-10000, Croatia

Editor

Marko Subašić (marko.subasic@fer.hr)
University of Zagreb Faculty of Electrical Engineering and Computing
Unska 3, HR-10000, Croatia

Production, Publishing and Cover Design

Tomislav Petković (tomislav.petkovic.jr@fer.hr)
University of Zagreb Faculty of Electrical Engineering and Computing
Unska 3, HR-10000, Croatia

Publisher

University of Zagreb Faculty of Electrical Engineering and Computing
Unska 3, HR-10000 Zagreb, OIB: 57029260362

ISSN 1849-1227

Organizing Committee

General Chair

Sven Lončarić, University of Zagreb, Croatia

Technical Program Committe Chair

Marko Subašić, University of Zagreb, Croatia

Local Arrangements Chair

Tomislav Petković, University of Zagreb, Croatia

Publications Chair

Pavle Prentašić, University of Zagreb, Croatia

Technical Program Committee

Bart Bijnens, Spain
Hrvoje Bogunović, USA
Mirjana Bonković, Croatia
Karla Brkić, Croatia
Robert Cupec, Croatia
Albert Diosi, Slovakia
Hrvoje Dujmić, Croatia
Ivan Fratrić, Switzerland
Hrvoje Gold, Croatia
Mislav Grgić, Croatia
Sonja Grgić, Croatia
Andras Hajdu, Hungary
Edouard Ivanjko, Croatia
Bojan Jerbić, Croatia
Zoran Kalafatić, Croatia
Stanislav Kovačič, Slovenia
Zoltan Kato, Hungary
Josip Krapac, Croatia
Sven Lončarić, Croatia
Lidija Mandić, Croatia
Vladan Papić, Croatia
Renata Pernar, Croatia
Tomislav Petković, Croatia
Ivan Petrović, Croatia
Thomas Pock, Austria
Tomislav Pribanić, Croatia
Arnau Ramisa, Spain
Slobodan Ribarić, Croatia
Damir Seršić, Croatia
Darko Stipaničev, Croatia
Federico Sukno, Ireland
Siniša Šegvić, Croatia

Reviewers

Albert Diosi, Austraila
Arnau Ramisa, Spain
Edouard Ivanjko, Croatia
Federico Sukno, Argentina
Hrvoje Bogunović, USA
Karla Brkić, Croatia
Marko Subašić, Croatia
Robert Cupec, Croatia
Siniša Šegvić, Croatia
Stanislav Kovačič, Slovenia
Thomas Pock, Austria
Tomislav Petković, Croatia
Vladan Papić, Croatia
Zoltan Kato, Hungary
Zoran Kalafatić, Croatia

Real Time Vehicle Trajectory Estimation on Multiple Lanes

Kristian Kovačić, Edouard Ivanjko and Hrvoje Gold
Department of Intelligent Transportation Systems
Faculty of Transport and Traffic Sciences
University of Zagreb
Email: kristian.kovacic@fpz.hr, edouard.ivanjko@fpz.hr, hrvoje.gold@fpz.hr

***Abstract*—Today's road traffic management systems using intelligent transportation systems solutions need real time measurements of various traffic parameters like flow, origin-destination matrices, vehicle type, etc. Cameras combined with image processing algorithms are being more and more used as the sensor capable to measure several traffic parameters. One such parameter, also important for accurate simulation of road traffic flow and evaluation of traffic safety, is the driving aggressiveness factor which can be estimated from the vehicles trajectory. In this paper an Extended Kalman Filter based approach to estimate vehicle trajectories on multiple lanes using only one static camera is described. To test the accuracy of the implemented approach a synthetic road traffic environment is developed. Real time capabilities of the approach are tested using real traffic video footage obtained from Croatian highways.**

I. Introduction

Today's traffic in urban areas is starting to cause heavy load to the existing road infrastructure. As road infrastructure in many cases cannot be modified (lack of build-up space), different approaches need to be taken in order to optimize traffic flow. Such approaches consist of applying intelligent transportation systems (ITS) which main goal is to apply a holistic approach for solving traffic problems using information and communication technologies. For optimal traffic control, ITS based systems need high quality traffic data in real time. Needed traffic data consists of various parameters such as traffic flow, distance between vehicles, velocity of vehicles, vehicle classification, etc. which all can be obtained from various sensors. Mostly used road sensors are inductive loops and nowadays video cameras also.

Video sensors or cameras combined with image processing algorithms are becoming an important approach to todays road traffic monitoring and control. From the obtained video footage high level traffic information can be extracted, i.e. incident detection, vehicle classification, origin-destination (OD) matrix estimation, etc. This information is crucial in advanced traffic management systems from the ITS domain. Commercial solutions for traffic monitoring by video cameras provide vehicle detection and tracking in scenes where there is a small amount of overlapping between the tracked vehicle and other objects (infrastructure or other vehicles). Additional drawback is that they need one camera per lane which is making such systems rather expensive. Proposed system described in this work has the main goal to achieve vehicle detection and tracking using only one camera for several lanes. Such a system can have a large number of ITS applications where it could be implemented for driver aggressiveness factor analysis, incident detection, traffic violation detection, etc. So, more high level traffic parameters measurements can be made enabling development of advanced autonomic or cooperative road traffic control approaches.

In today's society, where the vast majority of people drive on a daily basis in order to reach their destinations, aggressive driving has become a serious issue. Aggressive driving behaviors include speeding, driving too close to the car in front, not respecting traffic regulations, improper lane changing or weaving, etc. Obtaining such information in most cases is done by some kind of survey (eg. telephone survey) or by placing humans to monitor the traffic of a problematic area for a short amount of time [5]. Classic road sensors like inductive loops can not measure such driver behavior. Another approach of analyzing aggressiveness of driver behavior consists of using computer vision algorithms which process videos obtained from video cameras. By this approach, data can be obtained in real time with high accuracy [6]. Interesting data in this case is the vehicle trajectory on a road segment. By processing images from traffic video cameras, traffic violation can also be detected in the image as described in [7]. This system in combination with other ITS services can be useful for traffic law enforcement in cooperation with other agencies.

This paper is organized as follows: the second section describes the algorithm which performs vehicle detection and localization in the image. The third section describes the vehicle tracking algorithm which computes the vehicle trajectory. The fourth section describes optimizations made to the proposed system to ensure real time capabilities. The fifth section describes testing results of the proposed system. Paper ends with conclusion and future work description.

II. Vehicle detection

The first step in every vehicle detection algorithm beside importing of an image from a video stream is image preprocessing. After an image is imported it contains a certain percentage of noise. Noise complicates the vehicle detection process and significantly reduces the accuracy of the proposed system so it needs to be minimized. In [3], a Gaussian blur filter is used for noise reduction in a video footage. It reduces the number of details in the image including noise. In the proposed system a 5×5 matrix is used for the implemented Gaussian blur filter. Workflow of image preprocessing wherein renderings are distributed between the Central Processing Unit

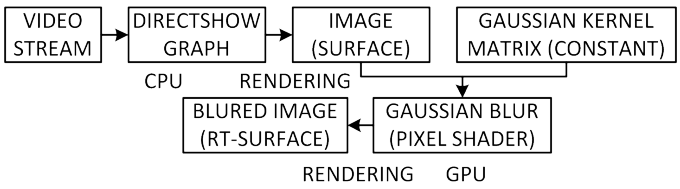

Fig. 1: Basic workflow of blur image preprocessing filter [1].

(CPU) and the Graphical Processing Unit (GPU) is given in Fig. 1.

After preprocessing of the imported image, various methods exist for vehicle detection. They can be divided into three types: optical flow methods; temporal difference methods; and background subtraction methods [9]. The system proposed in this work uses the background subtraction method. Workflow of the background subtraction method is shown in Fig. 2. Process consists of creating a background model of the scene and comparing computed background model with the latest preprocessed image imported from the video [2]. To create the background model the following equation is used:

$$BG_k = BG_{k-1} + \left\lfloor \frac{\sum_{i=1}^{n} sign(I_i - BG_{k-1})}{n} \right\rfloor, \qquad (1)$$

where BG_k represents the value of the specific pixel in the background model for the current frame and BG_{k-1} is the value of the specific pixel in the background model for the previous frame, I_i is the value of a certain pixel in i^{th} image, and n is the constant number of consecutive images stored in buffer ranging from the most recently acquired image k to the last image in the buffer $k - n + 1$. By comparing mentioned pixels in imported images, every pixel in the currently processed image can be classified. If the difference between the current image pixel value and the background model pixel value is larger than a specified threshold constant, the pixel is classified as a part of a foreground object. Otherwise it is considered as a part of the background model. The result of preprocessing and Fb/Bg image segmentation is given in Fig. 3.

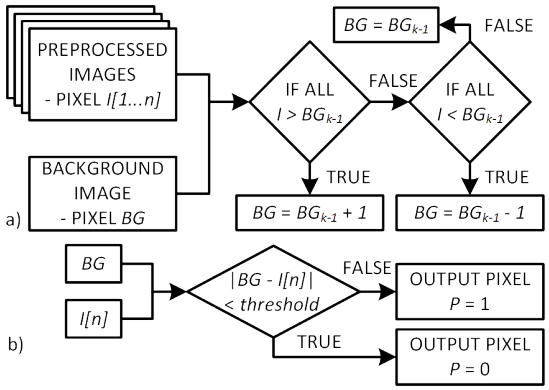

Fig. 2: Fg/Bg image segmentation workflow: a) background model creation, and b) background model and current image comparison [1].

Fig. 3: Original image (a) passed through preprocessing algorithm (b) and after Fg/Bg segmentation (c).

III. Trajectory estimation

When a moving vehicle is detected by the vehicle detection algorithm, its location in the image is obtained also. Detected vehicle location is given with (x, y) pixel coordinates and it contains information about the vehicle true location corrupted with noise. Noise disturbs vehicle tracking algorithm as measured vehicle location gets shifted for a certain value which is different for each image. This requires further processing of measured data.

The approach described in [4] uses data association and Kalman filtering for further processing of the object location. A data association algorithm is used to recognize the same object in series of consecutive images in order to track the respective vehicle or estimate its trajectory through time. As the measured object location contains noise, a Kalman filter is used to filter it. State model of the used Kalman filter is defined by object center (x, y) coordinates, area and velocity of an object. The system proposed in [8] uses genetic algorithms for data association and Kalman filter for trajectory estimation. Object detection is performed with background subtraction method based on mixture of Gaussian model [8].

The system proposed in this work processes object location using a modified data association algorithm mentioned in [4] and Extended Kalman Filter (EKF) for trajectory estimation. The first step in the data association algorithm is pixel clustering performed in the latest image obtained from the vehicle detection algorithm. Pixel clustering combines all adjacent pixels in the image into clusters. After all clusters are computed, they are compared with clusters from the previous image. If there is a positive match between two clusters in two consecutive images, both clusters are set to belong to the same object. If a cluster from the latest image has no match with any of the clusters in the previous image, it is considered to be a new object. If a cluster from the previous image has no match with any of the clusters in the latest image, it is considered that it has left the current scene. Matching criteria for cluster association in two consecutive images is given by the weights defined with the following equations:

$$w_{dist} = 1 - \frac{d - d_{min}}{d_{max} - d_{min}}, \qquad (2)$$

$$w_{area} = 1 - \frac{a - a_{min}}{a_{max} - a_{min}}, \qquad (3)$$

$$w_{cover} = \frac{a_{is}}{max(a_{obj}, a_{cl})}, \qquad (4)$$

$$w = \frac{w_{dist} + w_{area} + w_{cover}}{3}, \qquad (5)$$

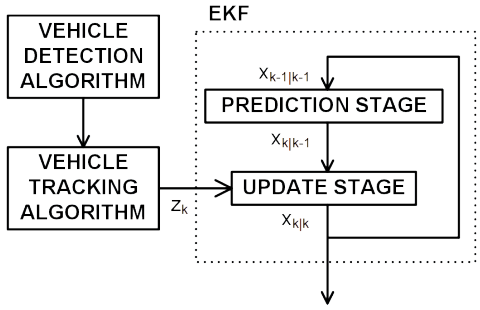

Fig. 4: Basic workflow of implemented EKF for vehicle trajectory estimation.

where d is distance between location of the specific cluster and estimated object location in pixels, d_{min} and d_{max} are minimum and maximum distance between all clusters and the processed object in pixels, a is difference between the cluster area (size) and the estimated object area, a_{min} and a_{max} are minimum and maximum difference between all clusters area and the estimated object area respectively, a_{is} is intersection area between cluster and object, a_{obj} is area of the object, and a_{cl} is the cluster area. All areas are expressed in pixels $[px]$.

To compute the distance between the location of a specific cluster and corresponding estimated object location their geometric centers are used. Cluster and object area are computed as their surrounding bounding box area. Matching gives a positive result only for cluster with the highest weight w and if $w_{cover} \geq \frac{2}{3}$.

EKF combines measured data with predicted state estimate. Result of this process can give more accurate trajectory then the one obtained by using measured data only. Basic workflow of the system is given in Fig. 4. The system first predicts state vector $x_{k|k-1}$ based on the state vector from the previous iteration performed by EKF in the update stage ($x_{k-1|k-1}$). Then the measurement obtained by the vehicle detection algorithm is combined with the latest state vector $x_{k|k-1}$ in the update stage. The obtained new state vector $x_{k|k}$ is used in the next iteration as input to the EKF ($x_{k-1|k-1}$). State vector can be defined with the following vector:

$$x = \begin{bmatrix} x_x \\ x_y \\ x_v \\ x_a \\ x_\phi \\ x_\omega \end{bmatrix}, \tag{6}$$

where x is state vector, x_x and x_y are vehicle x and y coordinates in the image in $[px]$, x_v is velocity in $[^{px}/_{s}]$, x_a is acceleration in $[^{px}/_{s^2}]$, x_ϕ is angle (direction) in $[rad]$ and x_ω is angular velocity of vehicle in 2D camera perspective in $[^{rad}/_{s}]$.

Measurement vector z can be defined with the following equation:

$$z = \begin{bmatrix} z_x \\ z_y \end{bmatrix}, \tag{7}$$

where z_x and z_y represent x and y coordinates of the vehicle in the image in $[px]$ obtained by vehicle detection algorithm.

Computation in the prediction stage is done by the following equations:

$$f(x) = \begin{bmatrix} x_x + x_v t \cos(x_\phi) + \frac{x_a[x_\omega t \sin(x_\omega t + x_\phi) + \cos(x_\omega t + x_\phi)]}{x_\omega^2} \\ x_y + x_v t \sin(x_\phi) - \frac{x_a[x_\omega t \cos(x_\omega t + x_\phi) - \sin(x_\omega t + x_\phi)]}{x_\omega^2} \\ x_v + x_a t \\ x_a \\ x_\phi + x_\omega t \\ x_\omega \end{bmatrix}, \tag{8}$$

$$x_{k|k-1} = f(x_{k-1|k-1}), \tag{9}$$

where $x_{k|k-1}$ is state vector and $x_{k-1|k-1}$ is state vector from previous iteration $k-1$ computed in update stage and t is interval (distance) between iteration k and $k-1$ expressed in the number of frames.

After the prediction stage is done, the predicted state vector is updated with the previous state $x_{k|k-1}$ and the latest measurements vector z_k. Computation of the new state vector $x_{k|k}$ is done using the following equations:

$$h(x) = \begin{bmatrix} x_x \\ x_y \end{bmatrix}, \tag{10}$$

$$y_k = z_k - h(x_{k|k-1}), \tag{11}$$

$$F_{k-1} = \left.\frac{\partial f}{\partial x}\right|_{x_{k-1|k-1}}, \tag{12}$$

$$H_k = \left.\frac{\partial h}{\partial x}\right|_{x_{k|k-1}}, \tag{13}$$

$$P_{k|k-1} = F_{k-1}\ P_{k-1|k-1}\ F_{k-1}^T + Q_{k-1}, \tag{14}$$

$$S_k = H_k\ P_{k|k-1}\ H_k^T + R_k, \tag{15}$$

$$W_k = P_{k|k-1}\ H_k^T\ S_k^{-1}, \tag{16}$$

$$x_{k|k} = x_{k|k-1} + W_k\ y_k, \tag{17}$$

$$P_{k|k} = (I - W_k\ H_k)\ P_{k|k-1}, \tag{18}$$

where y_k is innovation vector, $P_{k|k-1}$ is the covariance matrix of the predicted state estimate, F_{k-1} is the state transition matrix, $P_{k-1|k-1}$ is the covariance matrix of the predicted state estimate from the previous iteration, Q_{k-1} is the covariance

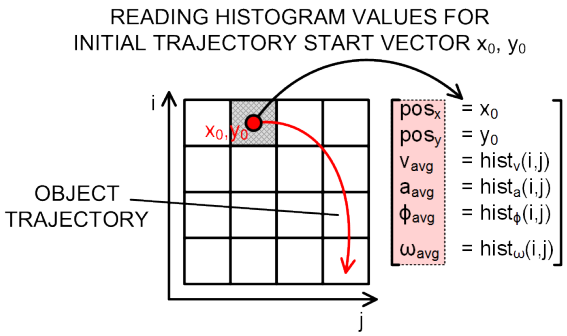

Fig. 5: Setting initial state values of EKF by using histograms.

Approach		Vehicle count		
		Total	Lane	
			Left	Right
Overlap check	Hits	126	65	61
	FP / FN	0/6	0/5	0/1
	Accuracy	95.6%	92.9%	98.4%
Trajectory check	Hits	129	68	61
	FP / FN	1/4	0/3	1/1
	Accuracy	96.2%	95.8%	96.8%
True vehicle count		132	70	62

Table 1: Counting results of the proposed system.

matrix of process noise, S_k is the innovation covariance matrix, H_k is the observation matrix, R_k is the covariance matrix of the observation noise, W_k is the Kalman gain matrix and I is identity matrix.

An important feature of the EKF is that before the first iteration can be computed, state vector $x_{k-1|k-1}$ and matrix $P_{k-1|k-1}$ need to be initialized. In the proposed system the initial values of vector $x_{k-1|k-1}$ are estimated by a histogram. Histogram is divided into $i \times j$ segments where each segment covers specific rectangular area of the image. Histogram is updated in every iteration of the EKF, where computed v, a, ϕ, ω components of a state vector $x_{k|k}$ from the EKF are added to the vector in the corresponding histogram segment. Histogram segments are determined by reading values of x and y components of a state vector $x_{k-1|k-1}$. Every component of a vector in the histogram segment represents the sum of all values in all previous iterations. If this sum is divided by the number of elements which were used in the sum, mean value can be obtained. In the first iteration of the EKF, values of x and y components of a state vector $x_{k-1|k-1}$ are set to values obtained directly from the vehicle detection algorithm and therefore they are not processed by the EKF. Other components of a state vector $x_{k-1|k-1}$ are set to mean values read from histogram as shown in the Fig. 5. After initialization, for every further EKF iteration, the state vector $x_{k-1|k-1}$ is computed only by the EKF (histogram values are ignored).

IV. Parallelization based speed-up

The first version of the implemented approach for vehicle detection has shown to be efficient from accuracy aspect according to the results given in Tab. 1. To ensure real time capabilities further development with aspect of using parallel computing abilities of today's CPU and GPU architecture has been done. Basic workflow of the proposed application which uses multi-threading (MT) and GPU support is shown in Fig. 6. Algorithms that process every pixel in the image can be time consuming for CPU even with use of Streaming SIMD Extensions (SSE) instructions support. Modern GPU architecture consists of many stream processors that can process data in parallel execution (SIMD instructions). This represents main reason for considering use of GPU in further development of current application. In the currently proposed system, image preprocessing and vehicle detection algorithm are entirely performed on GPU through pixel shaders. Pixel clustering is performed on CPU with MT support which improves performance of algorithm regarding execution time.

V. Results

The proposed system has been tested using real world road traffic video footage captured on a highway with two lanes near the city of Zagreb in Croatia. Camera was mounted above the highway and passing vehicles were recorded using a top view camera perspective as given in Fig. 7. Duration of the test video is $10\,[min]$. Obtained original video resolution is $1920 \times 1080\,[px]$ (RGB).

For vehicle detection results verification, two approaches for vehicle counting were tested. Both are based on markers (virtual vehicle detectors). Yellow and red rectangle markers are placed in the bottom part of the scene on each lane as shown in Fig. 7. Edges of markers are perpendicular to the image x and y axis. When a vehicle passes through a marker and a hit is detected, the counter for that marker is incremented. The first approach checks if an object is passing through a marker with its trajectory and the second approach performs check if an intersection between a marker and an object exists. Both approaches discard all objects whose trajectory direction is outside of a specific interval. In the performed test, all objects need to have their direction between $90 - 270\,[^\circ]$ in order not to be discarded. Objects also need to be on the scene for more than 30 frames. The value of the threshold constant used in Fg/Bg segmentation method is 10 and the number of consecutive images used when creating background model (n) is 105. Blue lines in Fig. 7 represent computed vehicle trajectory. Experimental results are given in Tab. 1. FP represents false positive and FN represents false negative hits. True vehicle count is acquired by manually counting all passed vehicles.

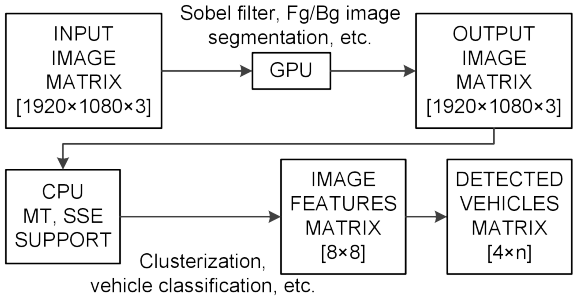

Fig. 6: Proposed workflow based on CPU/GPU computation distribution.

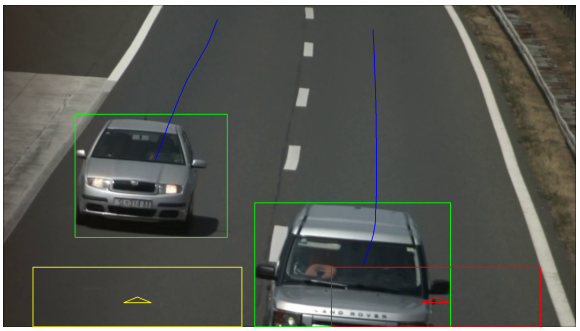

Fig. 7: Vehicle tracking and counting on two lanes.

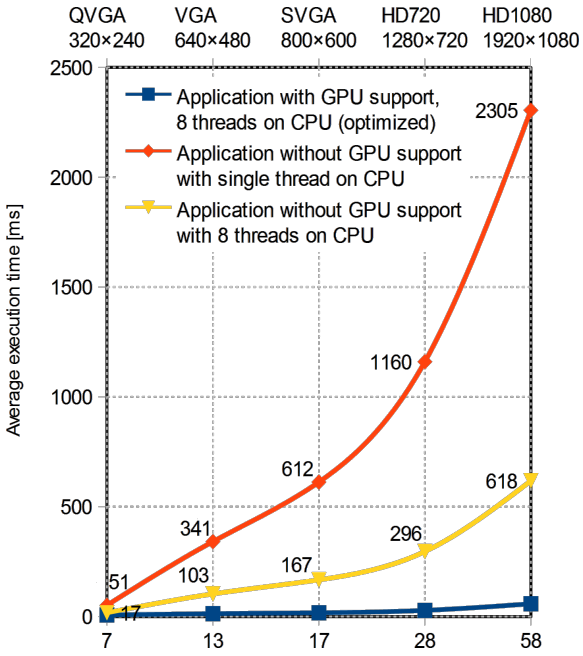

Fig. 8: Execution time of the proposed system.

In Fig. 8 execution time is given for various resolutions tested on a Windows 7 (64bit) computer with CPU Intel Core i7 - 2,4 GHz, GPU NVIDIA Quadro K1000M and 8 GB RAM. In the experimental testing, both approaches (overlap and trajectory check) for vehicle counting had the same execution time. From the acquired results it can be concluded that real time vehicle detection can be performed on SVGA $800 \times 600\,[px]$ resolution and lower using a standard PC computer. On SVGA resolution, $17\,[ms]$ is required to process a single frame. This enables maximum frame rate of $58\,[fps]$. At QVGA $320 \times 240\,[px]$ resolution, $142\,[fps]$ can be achieved with $7\,[ms]$ required to process a single frame. It can also be concluded that the approach with trajectory check gives better results regarding accuracy than the approach with overlap check.

For testing of the implemented EKF based vehicle trajectory estimation, a synthetic road traffic video was made in Autodesk 3ds Max. Video of synthetic environment simulates passing of one vehicle on a road with two lanes. As the true position of a vehicle in the synthetic environment is known, the implemented EKF can be tested for its trajectory estimation

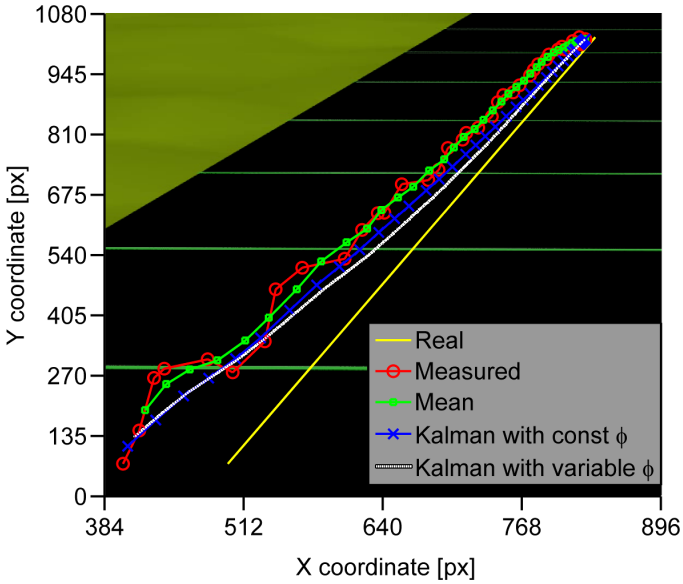

Fig. 9: Comparison of real, measured, mean and EKF vehicle trajectory.

accuracy. In Fig. 9 different trajectories obtained by various methods are compared. The real trajectory represents movement of vehicle geometric center defined during development of the synthetic video. The measured trajectory is computed by taking data (vehicle trajectory) from the vehicle detection algorithm and adding noise to it in order to simulate values which would be obtained by processing real world road traffic video. Noise is defined by standard uniform distribution in the interval $[-2.5, 2.5]$ and it is added to each vector of the vehicle trajectory. The mean trajectory is computed by taking the last 3 x and y coordinates of the measured trajectory and computing the mean value of them. So measurement noise can be reduced without significantly affecting vehicle location estimation accuracy. EKF trajectories are obtained by using two different state models. The first model is already described in the previous section. The second model is based on the first model with the angular velocity component removed.

In Fig. 10, x-axis represents number of frame for which error is computed and y-axis represents amount of error in

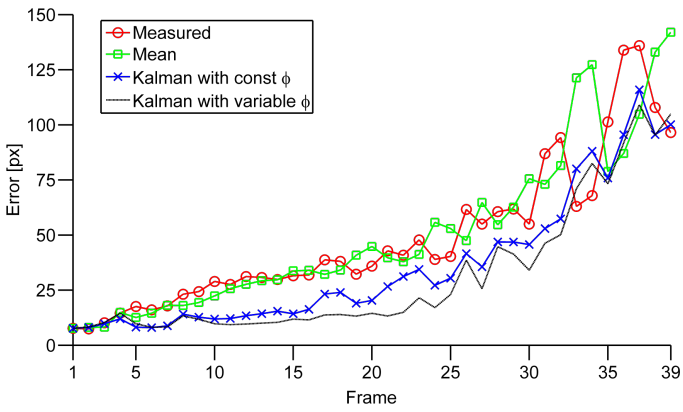

Fig. 10: Comparison of error in measured, mean and EKF vehicle trajectories.

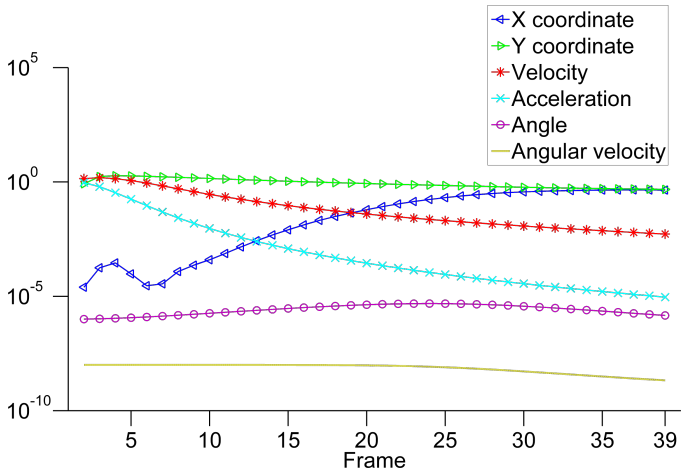

Fig. 11: Change of covariance matrix $P_{k|k}$ components.

$[px]$. Vehicle position error can be computed for any frame using the following equation:

$$err(k) = \sqrt{\left(x_r^{(k)} - x_f^{(k)}\right)^2 + \left(y_r^{(k)} - y_f^{(k)}\right)^2}, \qquad (19)$$

where k is the frame number, $x_r^{(k)}$ and $y_r^{(k)}$ are real measured values in $[px]$ of x and y coordinates for specific trajectory vector, $x_f^{(k)}$ and $x_f^{(k)}$ are filtered values in $[px]$ (using mean value method or EKF) of x and y coordinates for specific trajectory vector. Mean error value for the measured trajectory is $48.4\,[px]$, for trajectory obtained by mean method is $50.2\,[px]$, for trajectory obtained by EKF with constant vehicle direction angle ϕ is $35.9\,[px]$ and EKF with variable vehicle direction angle ϕ is $31.2\,[px]$.

Covariance matrix $P_{k|k}$ changes through 39 frames as shown in Fig. 11, where blue and green lines are x and y coordinates, red line is velocity component, cyan is acceleration, purple is angle and yellow is angular velocity component of the uncertainty matrix $P_{k|k}$. From the Fig. 11 it can be concluded that estimate covariances for y coordinate, angle and angular velocity have no rapid change in their values over whole vehicle trajectory. Opposite to that estimate covariances for velocity and acceleration decrease rapidly over time. Estimated covariance for x coordinate increases rapidly till it approximately reaches the value of estimate covariance for y coordinate. Vehicle velocities and acceleration can be estimated more accurately than the (x, y) vehicle coordinates. This can be explained that tracked vehicle moves in the video and enlarges during tracking. So, location measurements are more accurate when the vehicle is detected (vehicle enters the scene) than when it leaves the scene.

In Fig. 7 estimated trajectories of several vehicles on multiple lanes can be seen. The presented processed traffic scene proofs that the implemented system can successfully simultaneously detect and track vehicles on multiple lanes in real time.

VI. Conclusion

In this paper a system for vehicle detection and tracking on multiple lanes based on computer vision is proposed. The developed system uses only one camera to detect and track vehicles on multiple lanes. It solves drawbacks of the currently available commercial systems that use one camera per road lane. The first vehicle detection results are promising with an accuracy of over 95%.

Additionally, vehicle trajectory estimation has been added to the existing system. Because the trajectory of a vehicle contains a large ratio of noise, trajectory is filtered by EKF. For testing of the implemented EKF filter, synthetic environment was developed in which groundtruth vehicle trajectory is known. From the testing results in which the groundtruth data is compared with the computed data, it can be concluded that EKF can improve trajectory estimation accuracy. As the proposed system is computationally expensive it was optimized by implementing ability to execute specific image processing algorithms (preprocessing, Fg/Bg image segmentation) on GPU. The algorithm for pixel clustering which was too complex to execute on GPU was optimized by implementing CPU MT support.

Future work consists of developing a tracking system which would be able to perform license plate recognition and vehicle type classification of detected vehicles. So additional data can be obtained from which further analysis of the road traffic video footage could be made.

Acknowledgment

This work has been supported by the IPA2007/HR/16IPO/001-040514 project “VISTA - Computer Vision Innovations for Safe Traffic” which is co-financed by the European Union from the European Regional and Development Fund and by the EU COST action TU1102 - “Towards Autonomic Road Transport Support Systems”. Authors wish to thank Dominik Kovačić for developing the synthetic road traffic environment.

References

[1] K. Kovačić, E. Ivanjko, H. Gold, Real time vehicle detection and tracking on multiple lanes, WSCG, 2014, Czech Republic

[2] K. Kovačić, E. Ivanjko, S. Varela, Real time vehicle country of origin classification based on computer vision, ISEP, 2014, Slovenia

[3] V. Braut, M. Čuljak, V. Vukotić, S. Šegvić, M. Ševrović, H. Gold, Estimating OD matrices at intersections in airborne video - a pilot study, MIPRO, 2012, Croatia

[4] J. M. Jeong, T. S. Yoon, J. B. Park, The specific object tracking algorithm based on Kalman filter in an environment where similar objects are existing, ICCAS, 2013, Korea

[5] E. Wells-Parker, J. Ceminsky, V. Hallberg, R. W. Snow, G. Dunaway, S. Guiling, M. Williams, B. Anderson, An exploratory study of the relationship between road rage and crash experience in a representative sample of US drivers, Accident analysis and prevention, vol. 34, Social Science Research Center, Mississippi State University, 2002, USA

[6] H. B. Kang, Various approaches for driver and driving behavior monitoring: a review, ICCV Workshops, 2013

[7] Y. C. Chung, J. M. Wang, S. W. Chen, A vision based traffic light detection system at intersections, Journal of Taiwan Normal University: Mathematics, Science & Technology, 2002, China

[8] H. Wang, M. W. Ren, J. Y. Yang, Object tracking based on genetic algorithm and Kalman filter, CIS, 2008, vol. 1, pp. 80-85, China

[9] Y. Xia, S. Ning, H. Shen, Moving targets detection algorithm based on background subtraction and frames subtraction, ICIMA, 2010, vol. 1, pp. 122-125, China