Traffic Monitoring System Development in Jelgava City, Latvia

Vitalijs Komasilovs[1], Aleksejs Zacepins[1], Armands Kviesis[1], Eliecer Pea[2], Felipe Tejada-Estay[2], Claudio Estevez[2]

[1]*Department of Computer Systems, Faculty of Information Technologies, Latvia University of Agriculture, Jelgava, Latvia*

[2]*Department of Electrical Engineering, Universidad de Chile, Santiago, Chile*

{*vitalijs.komasilovs, aleksejs.zacepins, armands.kviesis*}*@llu.lv*, {*eliecer.pena, felipe.tejada, cestevez*}*@ing.uchile.cl*

Keywords: Traffic Monitoring, Video processing, Multi-Object Tracking, Motion Modeling

Abstract: Smart traffic management and monitoring is one of the key aspects of the modern smart city. Traffic flow estimation is crucial for sustainable traffic planning in the city. Requirement for successful planning and optimization of traffic is vehicle counting on the streets. Surveillance video is suitable data source for precise vehicle counting. In this paper authors propose solution for real-time vehicle traffic monitoring, tracking and counting in Jelgava city, Latvia. It is based on motion detection using background modeling, which is enhanced by statistical analysis. The system demonstrates good performance and acceptable accuracy on given test cases (about 97% precision for regular traffic conditions).

1 INTRODUCTION

Smart city has no one absolute definition (Anthopoulos and Reddick, 2016). In general, the smart city is identified by the following characteristics: smart economy, smart people, smart governance, smart mobility, smart traffic, smart environment, and smart living (Mahizhnan, 1999). Smart traffic consists of several topics, which are smart traffic light management, smart accident management, smart public transportation systems, vehicle identification, vehicle tracking and counting.

Data about the vehicle number can be used in various ways, such as to synchronize traffic lights, assist drivers in the selection of routes, and assist governments in planning the traffic system expansion, building new roads and data for designing better solutions for the urban and road traffic (Barcellos et al., 2015).

To this moment in Jelgava city (Jelgava is the fourth largest city in Latvia, is historical center of Zemgales region, distance from Latvia capital Riga is 42 km) vehicle counting is made manually by two approaches. First: one person (operator) manually observing the road and writing down vehicle count, second: camera is located near the monitoring place and video file is recorded, afterwards operator is watching the video and counts the vehicles.

Authors approach to improve this process is to use IT solutions for real-time video analysis without attracting the operator. Google Scholar shows about 25 500 results on search phrase ”vehicle counting from”, which means that this topic is actual and it is not yet fully discovered.

The use of image-based sensors and computer vision techniques for data acquisition on the traffic of vehicles have been intensely investigated in the recent years, since traffic videos provide more information about the traffic of vehicles than other conventional techniques (e.g. inductive loops, sonar or microwave detectors), and sometimes such video based systems can expand their monitoring capabilities by taking advantage of the video cameras already installed on site (Tian et al., 2011). Conventional techniques have disadvantages such as the installation cost, traffic disruption during installation or maintenance, and usually these methods are unable to detect slow or static vehicles (Mandellos et al., 2011).

Nevertheless, that there are plenty of algorithms and systems for image processing (Zhu et al., 1996; Wu et al., 2001; Rad and Jamzad, 2005; Iwasaki and Itoyama, 2007; Lim et al., 2009), image counting in real situations from real-time video stream is not a trivial task and there are challenges to solve. Unfortunately, there are no one ultimate system, which can be applied in all cases. As well price of the commercial system can be a factor, which limits its application by government.

This paper describes software solution for vehicle tracking and counting using image processing technologies. The live video is ob-

Figure 1: An example of video frame with marked area of interest

tained from public JelgavaLV channel on YouTube (https://www.youtube.com/user/JelgavaLV) from fix camera positioned above the road on the building wall by the address 5 J. Cakstes Blvd., Jelgava, Latvia. Vehicle traffic occurs in the diagonal direction, from top right (farthest from the camera) to bottom left (closest to the camera), and vice versa. Video has FullHD resolution of
1920×1080 px at 30 frames per second.

The paper describes approach used for vehicle traffic counting on live stream video (see figure 1). Apart from other objects (e.g. wires, bridge, pedestrians, buildings etc.) video stream contains regular two-way (one line to each direction) road of Jelgava city.

Complexity of traffic counting task is increased by several aspects: vehicle occlusions occur due to camera position, area of interest includes parking lots on both sides of street, speed bumps on a street, bridge cables appear on foreground, semi-hidden turning under the bridge.

2 MATERIAL AND METHODS

Figure 2 shows basic workflow of solution for vehicle traffic counting on live stream video. Input frames are extracted directly from YouTube FullHD stream (1920×1080), cropped to area of interest (576×648) and pushed to further processing, described in subsections below. Solution is implemented and tested using Python 3.5.2 environment. OpenCV 3.2.0 library (Bradski, 2000) is used for low level image manipulations and processing.

2.1 Vehicle Detection

This stage includes background modeling and vehicle detection. Proposed solution uses background subtraction method for motion detection on subsequent frames. Authors used (KaewTraKulPong and Bowden, 2002) algorithm implementation available in OpenCV (*BackgroundSubtractorMOG*), which gave better results on test cases comparing with (Zivkovic, 2004) implementation (*BackgroundSubtractorMOG2*) as shown on figure 3 (a) and (b) respectively. The parameters used are as follows: *history=50*, *nmixtures=3*, *backgroundRatio=0.1*, *noiseSigma=10*.

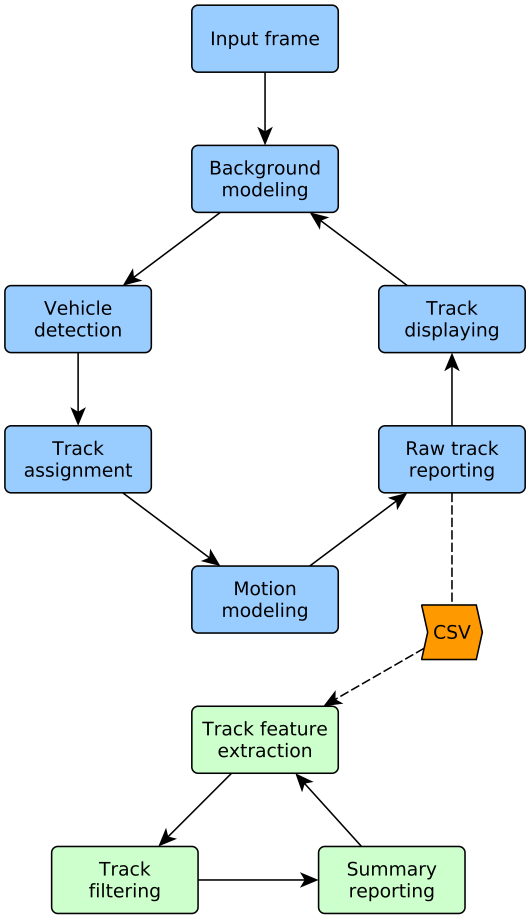
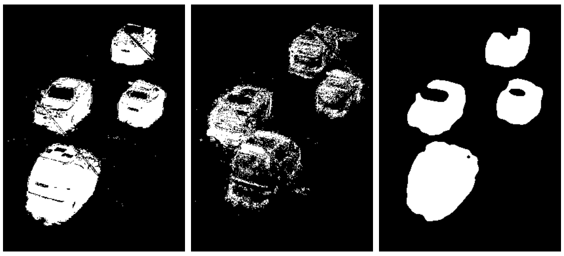


Figure 2: Principal process flow chart

For better moving blob segmentation additional mask operations are applied:

- Gaussian blur with kernel size 25×25 px and standard deviation 0;
- binary threshold on level 100;
- single erosion iteration with elliptical kernel 3×3 px.

The resulting final mask after all manipulations is shown on figure 3 (c).



(a) MOG (b) MOG2 (c) final mask

Figure 3: Moving vehicle masks

At the next stage contour search is performed on mask. The list of contours is filtered by area threshold $t_{area} = 2000$: contours with area less then 2000 px^2 (number obtained experimentally for given frame and vehicle size) are considered as non-vehicle detections (e.g. birds, tree leaves moving by the wind, walking people) and excluded from further processing. Then center coordinates of each contour is calculated and together with contour area and bounding box coordinates is packed into tracking point data structure and pushed to further processing.

2.2 Vehicle Tracking

Vehicle tracking stage stands for the problem of following same vehicle across multiple subsequent frames and includes track assignment, motion modeling and raw track reporting steps. As an input this module uses list of moving vehicles (contours) detected on current frame and it maintains its internal state of tracked vehicles from previous frames.

First of all detected vehicles are assigned to tracks - the trajectory of the moving vehicle from previous frames. Authors use modification of Hungarian algorithm for linear sum assignment problem and minimize the sum of distances between detections from current and previous frames (1):

$$\min \sum_i \sum_j D_{i,j} X_{i,j} \tag{1}$$

where D is distance matrix between last tracked points and current detected points, X is binary assignment matrix.

There are cases when on two subsequent frames part of tracked vehicles left the current frame and new (yet untracked) vehicles appeared on the frame. For proper handling of such cases authors use distance threshold $t_{dist} = 300$ px and area change ratio threshold $t_{areaRatio} = 0.3$, which in taken into account during assignment step: detection is not assigned to track if distance between two is more than 300 px or if area has changed by more than $\pm 30\%$. As a result new vehicles are properly assigned to new tracks, while old vehicles are left unassigned.

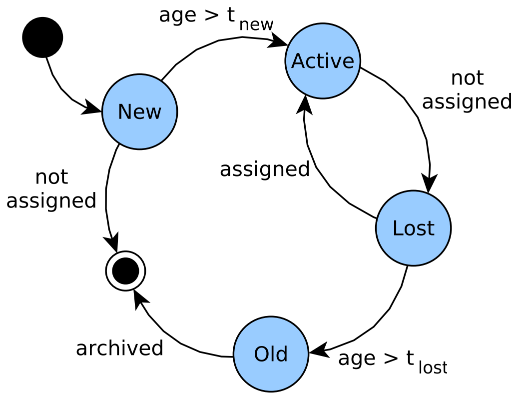


Figure 4: Track status transitions

In addition to increase stability of tracking and reduce false positive vehicle tracking cases authors use status modeling with several time thresholds (see figure 4):

- $t_{new} = 15$: time delay (frames) until newly tracked object becomes active, helps to eliminate shortly appearing detection blobs (e.g. shadows, car occlusions, clouds);
- $t_{lost} = 30$: grace period (frames) while lost track is kept among assignment candidates, helps to eliminate short gaps in detection (mostly car occlusions);

In order to smooth tracked vehicle trajectory and improve detection gap handling authors apply Kalman's filter for vehicle motion modeling and prediction. Internal state of the tracked vehicle is modeled using 6 dynamic parameters:

- x, y: object center coordinates on picture;
- v_x, v_y: object moving speed;
- s, v_s: object contour area and its changing speed.

As input for filter corrections x, y and s are provided from detection results, while speeds are maintained internally.

Finally active tracks are displayed on real-time monitor (see figure 5) and eventually *Old* tracks are archived to external database "as-is" (authors use separate CSV file for every 10 minutes of monitoring for easier handling).

2.3 Post-processing and Filtering

For better precision post-processing on reported raw tracking data is performed by separate process. This stage contains feature extraction, track filtering and

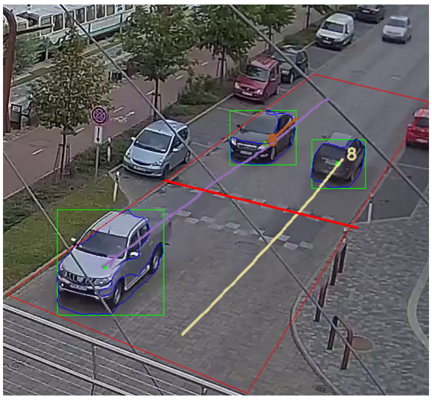


Figure 5: Real time monitor screenshot

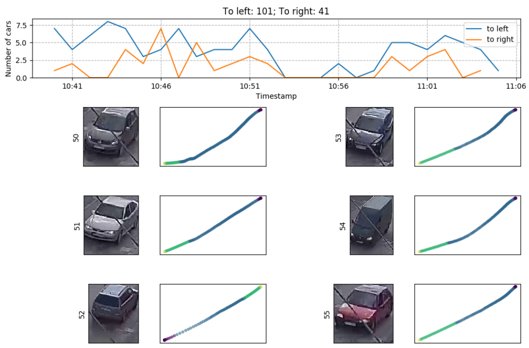


Figure 6: Summary monitor screenshot

reporting. First of all for each track following features are created:

- coordinates of tracking starting point;
- vehicle moving direction (*to left* or *to right*);
- number of recorded points in track and its length;
- imaginary mid-line crossing;
- linear motion trajectory interpolation (coefficients for line $y = ax + b$ and R^2).

Then statistical analysis is performed on resulting dataset and criteria are derived for filtering ineligible cases. Statistical analysis is described in details in next section.

Finally obtained statistics are periodically (every 5 seconds) displayed on summary monitor (see figure 6).

3 RESULTS AND DISCUSSION

This section describes results and their evaluation for proposed solution.

Table 1: Recorded video clips

Nr.	Time	Description
1	07:00	dawn
2	09:00	partly cloudy, sun reflections
3	11:00	cloudy
4	13:00	sunny
5	15:00	sunny, windy
6	17:00	rainy, congestion
7	19:00	partly cloudy
8	21:00	dusk, headlamp reflections

3.1 Experiments Setup

For testing and evaluation purposes eight 10 minutes long video clips were recorded from live stream at different time of the day, every 2 hours from 7:00 till 21:00 (see table 1 for details). On each clip vehicles are manually counted for ground truth reference. Then each clip is processed via proposed solution and results are collected.

For better statistical analysis another raw tracking dataset is gathered during several afternoons when most of traffic happens on the given street. In total it has about 10 000 recorded tracks.

3.2 Statistical Analysis

Raw vehicle tracking reports obtained from first stage of processing demonstrate poor precision: the system detects many false positives. Statistical analysis on recorded dataset were performed in order to develop suitable filtering conditions.

First of all tracks are filtered by moving direction and starting point coordinates in a way that tracks potentially cross imaginary mid-line. Observations confirm that majority of tracks started after mid-line by moving direction are either detection errors or duplicates of already tracked vehicles. Figure 7 shows spatial scatter plot of track starting points colored by moving direction.

Next track variation is analyzed by number of developed features (see figure 8). Number of recorded points in track is not correlating with track quality: there are correct tracks with small number of recorded points (fast moving vehicles) as well as with large number of recorded points (slow vehicles / congestion).

Contrary track length in pixels is good candidate for filtering condition. Whole class of false positives have short trajectories (vehicle occlusions, foreground wires, etc). Investigation shows that very long tracks are in essence tracks of two vehicles merged into one (track jumps between lines and goes in opposite direction). Such cases can be separated into two

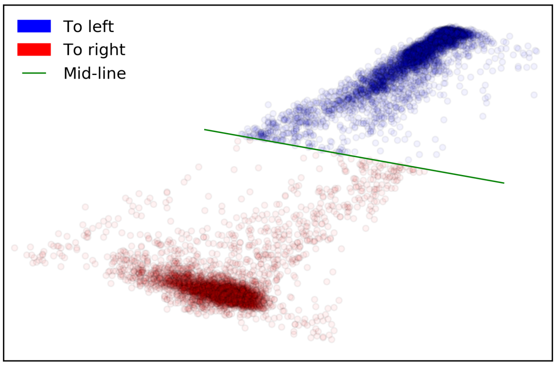


Figure 7: Track starting points and imaginary mid-line

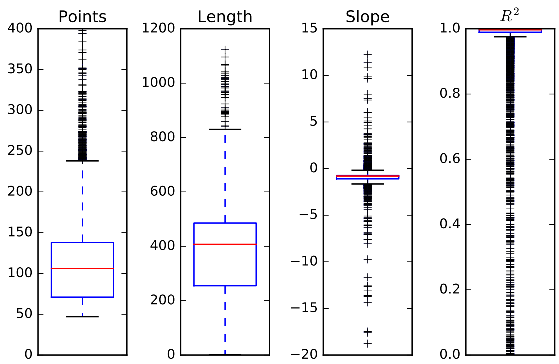


Figure 8: Track variation by selected features

independent tracks, but in practice these are very rare cases. Therefore authors count long tracks as single track.

Another set of features used for raw data filtering is trajectory linear interpolation results (a, b coefficients and R^2 value for line $y = ax + b$). Coefficient of determination R^2 shows how much given track is close to straight line. Despite the fact that R^2 variation is notably close to 1.0 there are a lot of outliners among acceptable tracks, and vice versa, wrong tracks with close to linear trajectory.

Coefficient b affects track pitch on the picture, which in traffic context relates to vehicle position on the street (left or right side of street). Due to presence of speed bumps in the middle of each line drivers tend to drive in-between of bums when there is no oncoming traffic. Therefore this coefficient is not suitable for any track selection.

Contrary a coefficient shows linear slope of the trajectory, which in turn should correspond to street direction for all vehicles. Figure 8 shows relatively low variation in a coefficient values, and selected outliner cases are clearly incorrect.

Figure 9 demonstrates samples of tracks selected by different criteria. After statistical analysis following criteria is used for selecting acceptable tracks out of raw tracking results:

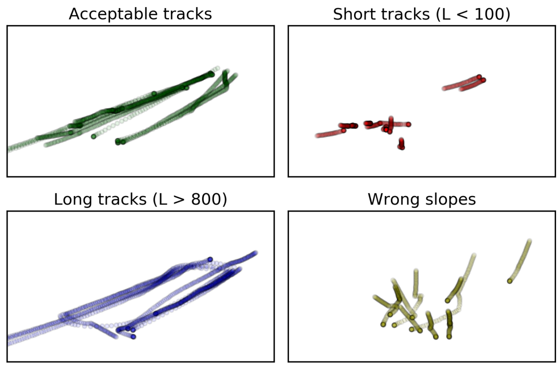


Figure 9: Track types selected by statistical analysis

- track starting point is before mid-line by vehicle moving direction;
- track length $L > 100$ px;
- slope of linear interpolation is in range $-2 < a < 1$

3.3 Precision Evaluation

Table 2 shows summarized precision evaluation results on all test cases. For each test case ground truth numbers are manually counted and considered as precision reference. Total number of raw tracks obtained from the first stage of processing are shown in the table (at this stage they are not divided by direction). Next numbers correspond to tracks left after analysis and filtering. Last two columns show evaluation of final precision of the system. Overall solution has acceptable precision for given task and conditions (about 97% precision for regular traffic conditions). There are two notable cases worth to highlight.

Test case 6 is recorded at 17:00 when traffic congestion happened on the given street. Proposed solution relies on motion detection via background modeling and due to congestion and minor vehicle movement it was not able to properly track separate vehicles (stopped vehicles were considered as background and movement phase was too short for starting tracking). As results show statistical analysis does not help to resolve these cases.

Another notable test case 8, which was recorded at 21:00 shortly after rain. Vehicle headlamps are reflecting from wet street surface, which leads to very high number of false positives in raw tracking results (almost all vehicles going left are counted twice: headlamp reflection and vehicle itself). However statistical analysis and raw result filtering helps to significantly improve results: from 58% to 94%.

Table 2: Precision evaluation summary

	Ground truth		Raw tracks		Statistical analysis		Final precision	
Nr.	to left	to right	count	precision	to left	to right	to left	to right
1	25	13	39	97%	25	13	100%	100%
2	63	25	96	91%	61	24	97%	96%
3	55	23	82	95%	54	23	98%	100%
4	50	54	107	97%	50	55	100%	98%
5	52	30	85	96%	51	32	98%	93%
6	80	71	188	75%	61	73	76%	97%
7	49	29	82	95%	47	31	96%	93%
8	37	18	78	58%	39	19	95%	94%

4 CONCLUSIONS

Traffic flow monitoring solution for real-time vehicle counting on live stream is proposed, developed and tested for Jelgava city in Latvia. It is based on background modeling, multi-vehicle tracking and statistical analysis. The experimental results show that proposed solution is suitable for variety of traffic and weather conditions, except congestions.

To extend system functionality it is planned to implement vehicle classification module (e.g. car, van, bus, truck, motorcycle, etc). Also for better congestion handling it is planned to apply object detection methods in addition to background modeling.

Camera positioning is crucial factor for precise traffic monitoring and it should be mounted in more elaborate location without any interfering objects (e.g. wires, pillars, bridges, etc).

ACKNOWLEDGMENTS

Scientific research, publication and presentation are supported by the ERANet-LAC Project Enabling resilient urban transportation systems in smart cities (RETRACT, ELAC2015/T10-0761).

REFERENCES

Anthopoulos, L. G. and Reddick, C. G. (2016). Understanding electronic government research and smart city: A framework and empirical evidence. *Information Polity*, 21(1):99–117.

Barcellos, P., Bouvié, C., Escouto, F. L., and Scharcanski, J. (2015). A novel video based system for detecting and counting vehicles at user-defined virtual loops. *Expert Systems with Applications*, 42(4):1845–1856.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Iwasaki, Y. and Itoyama, H. (2007). Real-time vehicle detection using information of shadows underneath vehicles. In *Advances in Computer, Information, and Systems Sciences, and Engineering*, pages 94–98. Springer.

KaewTraKulPong, P. and Bowden, R. (2002). An improved adaptive background mixture model for real-time tracking with shadow detection. *Video-based surveillance systems*, 1:135–144.

Lim, K. H., Ang, L., Seng, K. P., and Chin, S. W. (2009). Lane-vehicle detection and tracking. In *Proceedings of the international multiconference of engineers and computer scientists 2009 Vol II IMECS 2009, Hong Kong*.

Mahizhnan, A. (1999). Smart cities: the singapore case. *Cities*, 16(1):13–18.

Mandellos, N. A., Keramitsoglou, I., and Kiranoudis, C. T. (2011). A background subtraction algorithm for detecting and tracking vehicles. *Expert Systems with Applications*, 38(3):1619–1631.

Rad, R. and Jamzad, M. (2005). Real time classification and tracking of multiple vehicles in highways. *Pattern Recognition Letters*, 26(10):1597–1607.

Tian, B., Yao, Q., Gu, Y., Wang, K., and Li, Y. (2011). Video processing techniques for traffic flow monitoring: A survey. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1103–1108. IEEE.

Wu, W., QiSen, Z., and Mingjun, W. (2001). A method of vehicle classification using models and neural networks. In *Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd*, volume 4, pages 3022–3026. IEEE.

Zhu, Z., Yang, B., Xu, G., and Shi, D. (1996). A real-time vision system for automatic traffic monitoring based on 2d spatio-temporal images. In *Applications of Computer Vision, 1996. WACV'96., Proceedings 3rd IEEE Workshop on*, pages 162–167. IEEE.

Zivkovic, Z. (2004). Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE.