# Building a Robust Vehicle Detection and Classification Module

Anton Grigoryev[a,b], Timur Khanipov[b], Ivan Koptelov[b], Dmitry Bocharov[b], Vassily Postnikov[c], Dmitry Nikolaev[b]

[a]Moscow Institute of Physics and Technology, Moscow, Russia
[b]Institute for Information Transmission Problems, Moscow, Russia
[c]Cognitive Technologies, Moscow, Russia

## ABSTRACT

The growing adoption of intelligent transportation systems (ITS) and autonomous driving requires robust real-time solutions for various event and object detection problems. Most of real-world systems still cannot rely on computer vision algorithms and employ a wide range of costly additional hardware like LIDARs. In this paper we explore engineering challenges encountered in building a highly robust visual vehicle detection and classification module that works under broad range of environmental and road conditions. The resulting technology is competitive to traditional non-visual means of traffic monitoring. The main focus of the paper is on software and hardware architecture, algorithm selection and domain-specific heuristics that help the computer vision system avoid implausible answers.

**Keywords:** intelligent transportation systems, vehicle classification, vehicle detection, computer vision, automatic vehicle classifier

## 1. INTRODUCTION

In this paper we describe the major engineering challenges encountered in building a highly reliable recognition module using a set of well-studied (as it seemed at the beginning) scientific tools. The system we explore is the recognition core (AVC core) used in the Automatic Vehicle Classifier [1] currently deployed at scale on recently constructed toll roads of Russia. Many of the ideas and techniques we used to cope with these challenges are not bound to the specific domain of toll vehicle classification and might as well be used in developing other computer vision applications.

The AVC core detects and classifies vehicles passing through the gate near the camera. The system determines vehicle direction and measures a number of vehicle characteristics which are used to classify it according to a set of predefined rules (classification table). Correct direction has great importance since entry and exit events are processed differently by the upstream system. The computed vehicle parameters include the wheel axle count and height, which is either the maximum height of a vehicle or its height directly above the first axle depending on a particular classification table. The axle count is of course determined implicitly using only a side-view image and counting the number of visible wheels.

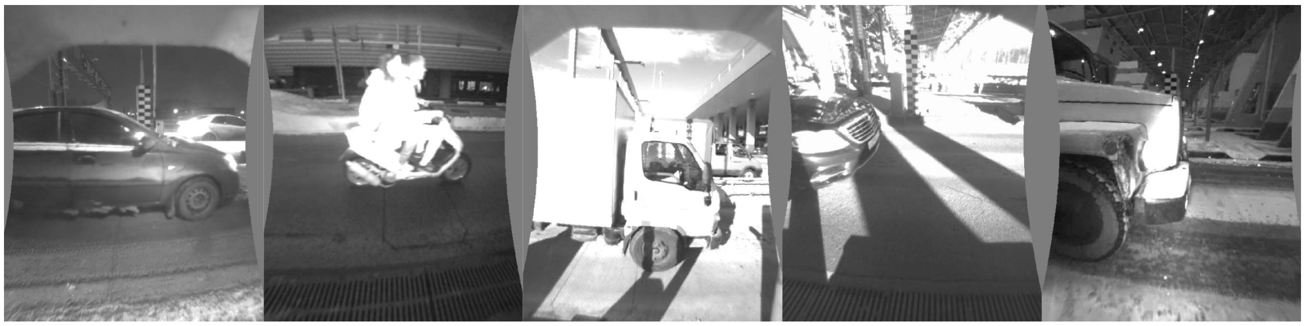
Figure 1. Examples of images processed by the Automatic Vehicle Classifier

Vehicle speed near the camera can be anywhere from 0-40 km/h (0-60 for designated fast lanes), but typical speed outside of traffic jams is 15-25 km/h. “Reversive” passes are possible, with a vehicle entering the classification zone and then backing out. This situation is common when the preceding vehicle cannot complete the transaction and needs to reverse out as well. Therefore there exists a problem of occasional strongly nonuniform motion and the direction estimation functionality must be resilient to it. The system is operating 24/7 and should work in all reasonable weather conditions, including strong sunlight, rain and snow. At night the artificial overhead lighting is available, but may be insufficient. See Fig. 1 for a representative sample of observation conditions. The performance goals are very strict: <0.1% vehicle detection and direction errors and <0.3% classification errors, with “entry” and “exit” events latency <150 ms and classification latency <500 ms.

In this paper we try to explain the key engineering aspects of this system by taking the “new-vs-old” approach to lower the cognitive load of multiple redundant subsystems: in Section 2 we describe the minimal functionally complete state of the system, whereas in Section 3 we explain additional parts that are responsible for reaching the final system robustness.

## 2. INITIAL DESIGN

This section introduces the overall AVC core architecture and describes the basic ideas we relied on in the beginning. The project started in late 2010 and the initial design described here roughly corresponds to its one year old state: after real data was collected and both software and hardware requirements became clear, but before the major functional flaws were identified. We must note, however, that the following description serves mostly as an exploratory device rather than being a precise historic account.

Since the maximum distance to the lane is limited by the traffic island width and most vehicles cannot fit into a single frame, we decided to use a wide-angle lens in portrait mode to cover the necessary height range with one camera and to let the vehicle scan itself as it passes in front of it. For the scanning scheme to work with fast vehicles we chose to do the entire processing at 40 fps and this choice proved to be good in the long-term perspective.

We used IDS UI-5221SE-M-GL (752x480 grayscale IR-sensitive) cameras with Fujinon YV2.7x2.2SA-SA2L or similar wide-angle lens with DC controlled iris. To deal with insufficient night lighting we used IR lights mounted below cameras. The first prototypes were running on various desktop PCs with quad-core Intel i7 CPUs of 2011 with the plan to use industrial APU-based fanless computers like eBox620-110-FL-T56N-1.65G (1.65 GHz dual-core CPU with integrated Radeon HD 6310 GPU) and offload most image processing operations to the graphics core.

The initial design is shown in Fig. 2 and is described below.

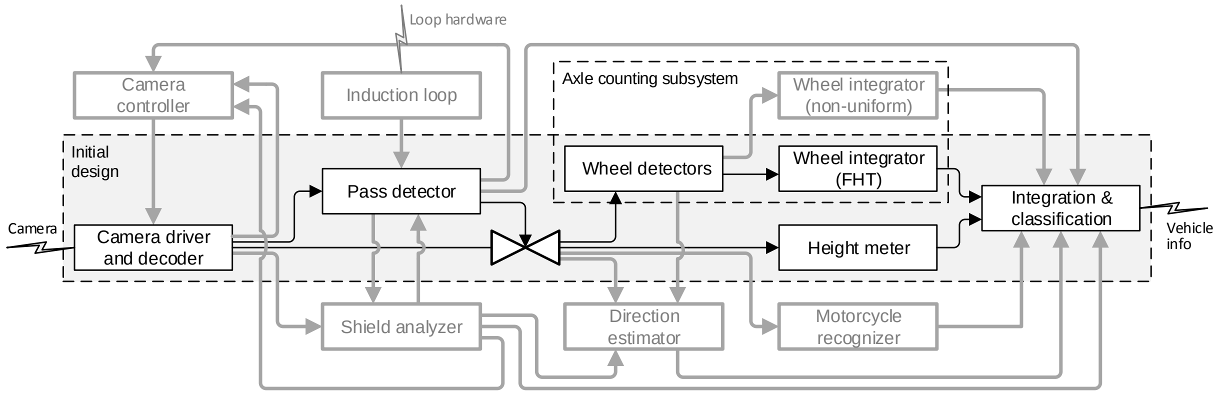

Figure 2. AVC Core architecture. Modules and links not present in the initial design are shown in gray. The schematic valve is opened by the pass detector during the detected pass.

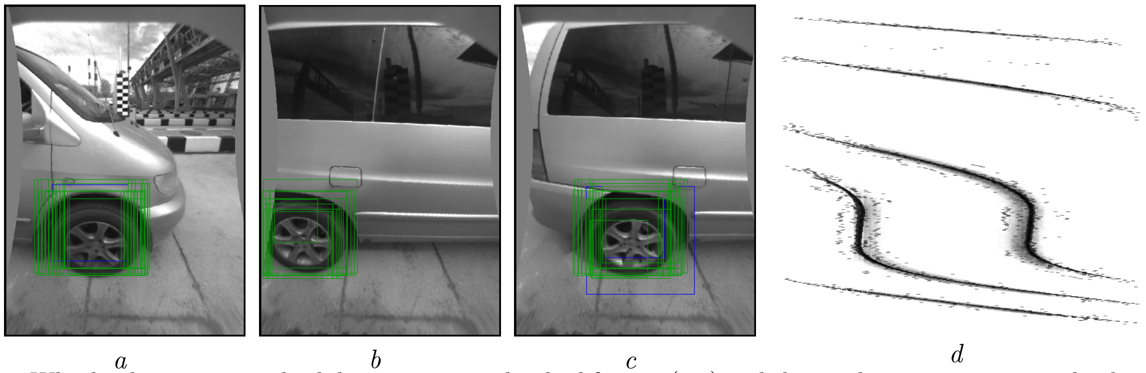

Figure 3. Wheel axle counting: wheel detections on individual frames (a-c) and the resulting $x-t$ image with wheel tracks (d).

**Camera driver and decoder** manages the image acquisition process and feeds two asynchronous video processing threads: the detection and classification thread and the **data recorder** (not shown) which can be enabled to save the videostream and metadata (induction loop and hints from the upstream system). The recorded data can later be read from disk to reproduce and debug errors or for QA & testing. The camera runs in auto-gain auto-exposure mode to allow for changing scene illumination.

The **pass detector** is responsible for indicating vehicle presence before camera, generating "entry" and "exit" events which toggle the operation of subsystems performing actual vehicle recognition and classification. The first successful implementation of the pass detector was based on simple single-mode exponential background modeling [2] and comparison with current frame using Pearson correlation coefficient, which is fast, simple and rather robust to the exposure adjustment due to its invariance to linear brightness transforms. Of course in presence of strong shadows it generates false positive detections, but without sun it performs well and therefore should be enough for indoor applications.

The **axle counting subsystem** estimates the number of axles by counting wheels moving across the frame. Each frame is fed to **wheel detectors** which use both machine learning-based (modified Viola-Jones [3]) and model-based detectors (elliptic curve detector similar to the one described by Levashov and Yurin [4]). This hybrid approach allows to reach a very good detection quality on typical wheels while maintaining acceptable quality in unusual cases by falling back on the model-based detector. Detections for each frame of the current vehicle are accumulated by the **wheel integrator** which analyzes them in spatiotemporal ($x-t$) space, counting tracks on the $x-t$ image (Fig. 3) using the Fast Hough Transform (FHT) [5].

The **height measurement** algorithm is based on simple background modeling like the pass detector, but instead of exponential running average it computes the mean image of a thin vertical band of the frame since the "entry" event, and estimates the top boundary of the vehicle. The bottom bound is obtained as a byproduct of wheel detection, by taking the median vertical coordinate of the detected wheels. The conversion from pixel boundaries to the actual height is performed with the help of a simple geometric model which uses pre-calibrated internal camera parameters, camera height and horizon position which is configured separately for each lane.

The **integration & classification** subsystem manages the class assignment according to simple rules which associate classes with height and axle count ranges. There is a special class "not a vehicle", which causes the system to reject a pass, assuming that it was toggled by a non-vehicle. Also it includes an heuristic to reject high axle counts ($< 3$) for vehicles with low estimated height, since such cases were often caused by false positive wheel detections. This is the place where most of the expert knowledge resides and it has grown to be the most complicated piece of code in terms of logic complexity.

The **system services** (not shown as they are connected to pretty much everything) include logging, system health monitoring, and visualization features. They allow monitoring and debugging a live instance and enable postmortem incident analysis and focused gathering of training and testing data.

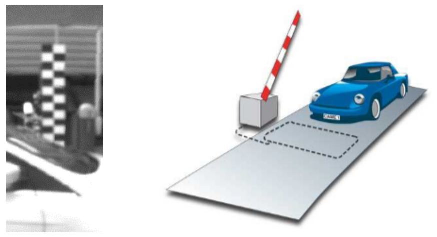
Figure 4. Additional hardware: shield and induction loop

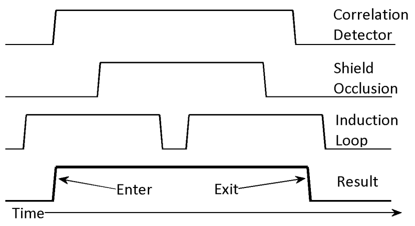

Figure 5. Multisensor vehicle pass detection

## 3. ENHANCED DESIGN

In the duration of about two years after initial AVC version rollout, quite a few systemic problems were discovered. The foremost problem was wheel detection performance. The OpenCL acceleration of Viola-Jones algorithm on the AMD T-56N proved unsuccessful due to memory bandwidth bottlenecks [6], and we have decided to switch the target platform to AAEON AEC-6637 (with a quad-core Intel Sandy Bridge CPU), which could run all four AVC cores at full 40 fps with two of them in pass detection mode and the other two (active) in recognition mode. Other problems concerned vehicle detection and classification quality. This section describes the architectural additions to the AVC core that allowed us to reach the target classification quality of 99.7%. The new parts are shown in figure 2 in gray.

### 3.1 Introducing Shields

After analysis of the problems it was deemed that a software-only solution for all the pass detection and height measurement problems, while scientifically interesting, would make no economic sense, and we resorted to augmenting the system with hardware components. The most important of them was the installation of “shields” which are narrow ($0.2 \times 2.0$ meters) checkered boards installed in front of cameras on the opposite sides of the lane (Fig. 4). The known texture of the shield allows to robustly check for occlusions by comparing image intensities at positions of adjacent black and white cells: the inter-cell border is definitely occluded when “white” cell turns to be darker than the adjacent “black” cell. Since the position and dimensions of shields are known, it can be easily identified in the image using a grayscale hit-miss transform (to avoid dealing with binarization), thus providing a form of auto-calibration outside of vehicle passes (hence the link from pass detector to shield analyzer). The **shield analyzer** checks the shield for occlusions and feeds the occlusion area and boundaries to pass analysis subsystems, notably the pass detector (a vehicle presence signal biased towards false negatives) and the classifier, where it corrects the vehicle boundary from the height meter.

### 3.2 Shield-Aware Camera Controller

Quality of the input image is of foremost importance for any recognition system, and for an all-weather 24/7 system automatic camera adjustment can be tough to get right. Initially the AVC used the builtin auto-exposure and auto-gain provided by the IDS camera driver, but after summer testing it became clear that we need to control the iris. The camera did not have builtin support for iris control, so we had to build a custom circuit to convert the general-purpose PWM signal from the camera to DC values controlling the iris. Another concern was that when a large vehicle filled the whole frame, the camera readjusted itself on the vehicle and not on the environment, which led to problems with exit event detection. Thus we designed an improved control for camera parameters to normally adjust exposure, gain and iris only outside of passes, while detecting strong over- and underexposure also within passes to protect from pass detector deadlocks. Outside the pass the goal is to keep the shield image within strict brightness limits. The shield thus aided significantly in controlling the camera since it is located in the same area as the passing vehicle and thus allows to optimize the image within the needed region, notwithstanding local over- and underexposures in less relevant areas.

Another significant feature of the camera controller is the adaptive setting of shield brightness levels themselves. Ideally, we need to stabilize the brightness of passing vehicles, most importantly their wheels, while we can adjust the camera only outside passes to avoid interference with the recognition. Therefore we measure the mean wheel area brightness for each pass, and if it is outside the normal range we adjust the shield target brightness, thus creating an indirect feedback loop that compensates for ambient illumination changes. This is important at dawn and dusk because night illumination is nonuniform and the ratio of vehicle and shield brightness changes significantly.

## 3.3 Multisensor Pass Detection

Pass detection errors are costly for our system since in many cases the AVC is used to control the lane barrier, and in the worst case either the barrier would strike a vehicle or several vehicles would pass without due payment. In addition to old correlation-based sensor and the shield occlusion signal, the modern pass detector also utilizes a signal from metal-detecting induction loops (Fig. 4), which were already installed at AVC operation sites and thus added no cost. In terms of reliability the new detector outperforms each of the individual sensors: the correlation sensor breaks under complex illumination, the shield tends to break passes e.g. in the case of trailers, and the loop has poor spatial resolution, tending to merge passes. Using a rule-based decision process the weak properties of individual detectors cancel each other (Fig. 5) [7].

## 3.4 Direction Estimation Subsystem

Related to pass detection, there is also a motion direction estimation problem, because the upstream processing for entering and exiting vehicles is necessarily different. In fact, one direction error is equivalent to two pass errors. The initial core relied on the wheel integrator to obtain the direction, but since it turned out that the required direction reliability was much higher (99.9%) we began working on an independent subsystem to estimate direction independently from axle counting process. The resulting direction estimator computes a mode of the histogram of solutions to one-dimensional optical flow assuming the vehicle moves horizontally. It uses a hint from the shield analyzer to determine when it is safe to gather statistics on background noise level.

## 3.5 Enhanced Wheel Detectors

The elliptic object detector has remained without change as an effective measure of last resort. Viola-Jones-based detector, however, has underwent several enhancements, mostly to combat high-contrast shadows obscuring the field of view.

The first addition was the diagonal edge-image-based feature vector, where the Canny edge image was filtered to leave only diagonal edge segments and label them $+1$ or $-1$ depending on orientation. This immediately allowed us to create a robust detection cascade for truck wheels since they have enough details to be recognizable from the edge image.

The second attempt at redesigning the wheel detectors was focused on detection time, since the number of cascades required for good detection has grown too big to sustain 40 fps. The result was our generalization of Viola-Jones as the decision tree [8]. This version also included a unified training infrastructure that combines multiple features (image, gradient, diagonal edges) into a single detecting tree.

For the final version we had only really difficult error cases, and we decided that they were beyond the power of Viola-Jones framework. Thus we built a hybrid classifier using a convolutional neural network (CNN) instead of several final layers of cascade. This yielded a very trainable classifier, and the computational cost of the CNN was incurred only in candidate regions selected by Viola-Jones.

## 3.6 Improved Wheel Integrator

The current wheel integrator is described in detail in our previous work [9]. Briefly, it became clear that strongly non-uniform motion constitutes a significant fraction of traffic ($\approx 10\%$), thus we have made two improvements: firstly, a new method based on connected component analysis of the x-t image was implemented, and secondly, the old FHT-based method was augmented with connected-component-aware wheel track erasure. For slow nonuniform motion there is plenty of time to recognize wheels, thus wheel tracks can be counted as connected components. The FHT method is more robust to missed wheel detections and thus more appropriate for fast vehicles, but with connected component erasure it doesn't break when a long vehicle enters the classification area at high speed and then slows down. The decision which method is more reliable is based on various heuristics like pass duration and wheel track slope on the x-t image and is made by the classification subsystem.

## 3.7 Problem-Specific Heuristics

When approaching high classification quality it becomes clear that among errors there is a large number of near extreme cases which would take huge effort to correct by adjusting the algorithms and yet may be solved easily by introducing a few relatively simple heuristics. For example, under low lighting conditions when camera controller fails to set the optimal picture parameters or external lights fail, vehicle wheels may be almost invisible and would not be detected at all leading to a lost vehicle (the system would consider it a non-vehicle). At the same time we know that if a vehicle both occludes the shield and triggers the induction loop it is a big metal object, i.e. a real vehicle with high certainty. Hence if we emit a vehicle with some default number of axles in this situation we prevent many lost vehicles (leading to costly queue errors) and in many cases we even guess the correct class.

The AVC has a few such domain specific heuristics. These form "the last mile" of the recognition and classification process targeted for dealing with especially difficult cases. The aforementioned heuristics gives 12% recovered falsely ignored passes. Another heuristic demotes a vehicle which has exactly one detected axle (by default extrapolated to 2 axles) to a non-vehicle with 0 axles if the induction loop has not been triggered. This accounted for 1% error reduction.

The other type of heuristics deals with wrong classification. The "axle-distance" rule computes the distance between the first two axles and merges them into one if the distance is too small. This allows to suppress false wheel detections on various truck parts, reducing errors for trucks by 26%.

## 3.8 Experimental evaluation

In developing of the AVC we have devised the following three indicators: queue quality (QQ), classification quality (CQ), and integral quality (IQ). QQ measures how well consistency between the real and the virtual vehicle queues is maintained by the AVC core and is penalized if either pass is detected incorrectly (lost, split, etc) or its direction is wrong. CQ shows AVC classification performance for vehicles which move forward and are correctly detected. It is worth noticing that CQ might even drop when QQ increase, because the base for CQ increases without necessarily increasing the number of correct cases. The integral quality (IQ) is used as a total AVC performance measure, combining QQ and CQ with QQ having higher impact to better simulate the apparent functioning of the AVC.

The effect of disabling various subsystems described above is shown in table 1. The usefulness of additional modules is clearly observable, but some important effects are in eliminating whole classes of systematic errors which cannot be seen without much deeper dataset analysis.

Fig. 6 shows the evolution of the three quality metrics over time since July 2014 to July 2015. The 1.0.0 version already includes most enhancements except the motorcycle recognizer (first appeared in 1.7.1). The graph shows the improvement in recognition quality from incremental enhancement and tuning of various subsystems. Unfortunately we cannot show the "prehistoric" versions with less enhancements since the testing interface has changed significantly around 1.0.0, yet the general trend is also characteristic of earlier development.

| **Modification** | **QQ** | **CQ** | **IQ** |
|---|---|---|---|
| Basic (all features) | 99.23% | 98.66% | 98.56% |
| Without direction estimator | 99.09% | 98.61% | 98.40% |
| Without induction loop | 96.75% | 98.59% | 96.07% |
| Image-only pass detector | 94.42% | 98.90% | 93.90% |
| FHT-only wheel integrator | 99.21% | 90.18% | 94.34% |
| Height estimator without shield | 99.23% | 98.30% | 98.38% |

Table 1. The effect of disabling the AVC core algorithm enhancements on the quality metrics QQ (queue quality), CQ (classification quality) and IQ (integral quality).

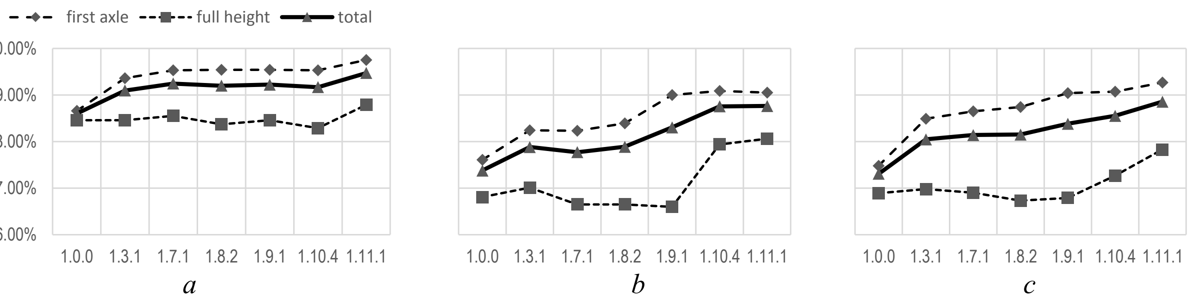

Figure 6. Historic quality plots: queue quality (a), classification quality (b), and integral quality (c). Dashed lines represent metrics for different classification schemes (axles + full height and axles + height over first axle).

## 4. CONCLUSION

This paper gives a superficial description of the whole recognition core of the Automatic Vehicle Classifier and its development over time. Both hardware and software enhancements are covered demonstrating the importance of a complex method of dealing with computer vision challenges. We believe that the problems and solutions highlighted here are typical for development of many industrial recognition systems and thus the analysis of our case could help both invent new ideas and prevent some design oversights in other systems.

In addition to improving the basic algorithmic subsystem, introduction of heuristical knowledge was crucial for reaching the desired performance characteristics of the AVC core. This knowledge fuses the algorithmic modules together, leading to a coherent and robust system.

**ACKNOWLEDGEMENT** The applied scientific research is supported by the Ministry of Education and Science of Russian Federation (project No. RFMEFI57914X0011).

## REFERENCES

[1] T. Khanipov, I. Koptelov, A. Grigoryev, E. Kuznetsova, and D. Nikolaev, "Vision-based industrial automatic vehicle classifier," in Seventh International Conference on Machine Vision (ICMV 2014), 944511–944511, International Society for Optics and Photonics (2015).

[2] J. W. Taylor, "Smooth transition exponential smoothing," Journal of Forecasting **23**(6), 385–404 (2004).

[3] P. Viola and M. Jones, "Robust real-time face detection," International Journal of Computer Vision **57**(2), 137–154 (2004).

[4] A. Levashov and D. Yurin, "Accurate and reliable framework for fast parametric curves detection," (2011).

[5] D. Nikolaev, S. Karpenko, I. Nikolaev, and P. Nikolayev, "Hough transform: Underestimated tool in the computer vision field," 238–243 (2008).

[6] S. Gladilin, A. Grigoryev, A. Kotov, and D. Nikolaev, "Viola-Jones algorithm implementation in OpenCL," Trudy ISA RAS **64** (2014). In Russian.

[7] D. Bocharov, D. Sidorchuk, I. Konovalenko, and I. Koptelov, "Vehicle passes detector based on multi-sensor analysis," SPIE, Seventh International Conference on Machine Vision (ICMV) **9445** (2015).

[8] A. Minkina, D. Nikolaev, S. Usilin, and V. Kozyrev, "Generalization of the Viola-Jones method as a decision tree of strong classifiers for real-time object recognition in video stream," SPIE, Seventh International Conference on Machine Vision (ICMV) **9445** (2015).

[9] A. Grigoryev, D. Bocharov, A. P. Terekhin, and D. P. Nikolaev, "Vision-based vehicle wheel detector and axle counter," in 29th European Conference on Modelling and Simulation, ECMS 2015, Albena (Varna), Bulgaria, May 26-29, 2015. Proceedings., V. M. Mladenov, G. Spasov, P. Georgieva, and G. Petrova, eds., 521–526, European Council for Modeling and Simulation (2015).