

## GROUND TRUTH GENERATING TOOL FOR TRAFFIC VIDEO DETECTOR

SZYMON BIGAJ ANDRZEJ GŁOWACZ<sup>1</sup> JACEK KOŚCIOW ZBIGNIEW MIKRUT<sup>2</sup> PIOTR PAWLIK<sup>2</sup>

<sup>1</sup> AGH University of Science and Technology, Faculty of Computer Science,  
Electronics and Telecommunications, Department of Telecommunications  
aglowacz@agh.edu.pl,

<sup>2</sup> AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics,  
Computer Science and Biomedical Engineering, Department of Automatics and Biomedical Engineering  
{zibi,piotrus}@agh.edu.pl

**Abstract.** The paper presents an application for generating ground truth data for the purposes of video detection and justifies its use in systems which analyze road traffic videos. The usefulness of described application in the development of video detection software is presented - especially during scene configuration and comparative analysis of video detection results versus ground truth data. The latter is possible due to simplicity of the result text files generated in a similar way both by the presented application and by the video detection algorithm. Two exemplary applications of the tool designed to generate ground truth data are presented, together with a discussion of their construction, functionality and abilities.

### 1 Introduction

Ground truth represents the required results of the action of an algorithm or an application. It makes it possible

to verify the precision of the designed solutions. Ground truth can be generated manually, or by using an automated gold standard which provides highly accurate results [14]. It can provide information on the object's location, its current condition, or duration in the current position or condition. Generating ground truth is essential in the verification of results generated by algorithms. It is used in a wide range of fields, such as medical diagnostics, road traffic analysis, crowd behavior analysis, and verification of algorithms used in computer games [2]. Ground truth makes it possible to calibrate systems, as well as improving the understanding of their mechanisms of action and detecting potential errors. Video analysis systems aim to facilitate decision making, in certain cases making these decisions with no human input. They are also differentiated by their purpose for detection, tracking and analysis of objects and people (for example during accidents or fights) [2, 4]. In this paper the software, used primarily to collect ground truth traffic data (vehicle counts, speed, calculation of the length of queues), is presented. After

saving ground truth data the software compares it with the results generated by the video detector. This software also supports the process of scene configuration. GTGT (Ground Truth Generating Tool) is closely matched to the video detector algorithm. For this reason, in the next section a coarse description of the algorithm will be presented. In the following sections other methods of generating ground truth will be discussed. Article is complemented by the description of the GTGT capabilities and examples of GTGT practical utilization.

## 2 Traffic video detector: the algorithm

The term 'video detector', used in this article, means a software for the analysis of digital video stream in order to calculate the selected traffic parameters (measurements). It is a traditional name, used in many commercial systems [1] and in the papers related to the area of ITS (Intelligent Transportation Systems) [8, 18]. In this section the general description of video detector algorithm is presented. This description is necessary to understand the impact of GTGT on the modification of video detector software and scene configuration. The details of the algorithm can be found in [11]. The task of video detector is to compute several traffic measurements. These should be conducted automatically on the basis of a sequence of images from cameras. The aim of the analysis is to calculate:

- length of queues of waiting vehicles (in meters or as a number of vehicles),
- number of vehicles leaving the intersection in a certain direction,
- vehicle speed.

The video detector supported by GTGT uses the Horn-Schunck method to calculate optical flow (OF). The input of the video detector algorithm is a binary image. It is

obtained by binarization of optical flow modulus calculated from every third frame. Next, after typical image preprocessing (morphological operations, deleting small objects) the segmentation is carried on. It was found that after segmentation, there is no guarantee that one vehicle complies with a single object. As a result of vehicles having uniform areas (which do not generate significant OF values) or becoming obscured by fixed background objects, the vehicle image can split into two or more objects. In turn, vehicles obscuring one another while moving along adjacent lanes may result in images of two or more vehicles merging into a single visual object. The video detector attempts to combine objects that were split incorrectly and divide those that were merged incorrectly. The algorithm operates on a list of objects (representations of vehicles) containing data such as surface area, bounding box, and several recent locations of the centre of gravity. These object are traced on subsequent frames and following cases are examined:

1. a single object in the list corresponds to single visual object on subsequent frame,
2. a single object in the list corresponds to two or more visual objects on subsequent frame (splitting).
3. two or more objects in the list correspond to single visual object on subsequent frame (merging).
4. two or more objects in the list correspond to two or more visual objects on subsequent frame (splitting and merging),

where the term "corresponds" is understood as the objects containing a sufficiently large intersection on two consecutive frames. The first case is the simplest one: object is easily tracked. The fourth case occurs very rarely. It is not analyzed, since the outlay required to study it exceeds the frequency of its occurrence many times over. The other cases (2 and 3) involve additional computations such as

distance analyze, directional dilation, new vehicle detection in order to correct merging and splitting. Traced vehicles are counted on user-defined virtual areas of crossroad. Algorithm estimates their speed and measures length of queue ahead of traffic signals. Measured quantities need to be verified by objective ground truth data. Such data should be generated manually (using GTGT), by clicking single points: the appropriate procedures are described in Section 4. The other GTGT capabilities are presented in Section 5.

### 3 Related works

Manual generation of data for video detection is extremely time-consuming. For example, a video sequence lasting 10 minutes at 30 frames per second (fps) contains a total of 18,000 frames. The process of analyzing and recording the results in order to verify the operation of the video detector would be too time consuming without a supporting tool. It is also important to maintain precision by enlarging and playing a fragment of the recording, or defining a region of interest. There are many tools which significantly accelerate and facilitate processing of a given recording. Almost always the ground truth data are generated by a human operator (annotator). The exception to this rule are studies conducted in the U.S. in the mid-2000s. In 2003 and 2004, two test beds were constructed at signalized intersections in Noblesville and West Lafayette, Indiana [18]. Three commercially available and widely used video detectors: Autoscope (version 8.10), Peek UniTrak (version 2), and Iteris Vantage (Camera CAM-RZ3) were evaluated on the same traffic conditions. To identify potential errors such as missed presence calls or false presence calls, the output from each of the video detectors was compared to the output of an inductive loop detector. The video detectors were each configured by vendor representatives to replicate the loop detector zones as closely as possible. Whenever there was

a discrepancy between the loop and video, a digital video was observed to determine the cause of the discrepancy. Similar experiments, provided in Illinois, are described in report [15]. Ground truth data were generated in the same way but the experiments concerned effects of configuration changes and illumination conditions in the performance of above mentioned video detection systems, located side-by-side.

The other tools for generating ground truth, described in this section, have been developed for video annotation provided by a human operator - manually or semi-automatically. The contractors of CAVIAR (Context Aware Vision using Image-based Active Recognition) project [10] propose generating ground truth manually as bounding boxes defining the individuals and groups, as well as a semantic description of the situation (behavior). Due to the variability of the ground truth observed e.g. during generating of ground truth for PETS04 video sequences three different individuals were involved in. The comprehensive description format of data, saved in the form of custom grammar, enables precise yet time-consuming analysis of the video surveillance algorithms operation.

In [12] the ODVIS (Open Development for Video Surveillance) application is presented, enabling generation of ground truth, observation of the video detector operation, and error detection. Data is generated in the form of vectors which define the motion of objects detected by the video detection algorithm. Errors are presented in three ways: by text, as graphs, or superimposed directly onto the recording: this facilitates the analysis of the operation and capturing critical situations when the video detector fails.

Authors of [7] propose the GTVT (Ground Truth Verification Tool) for creating ground truth data and to compute the performance of different video surveillance algorithms. When applied to traffic video detection GTVT makes it possible to select the area occupied by a vehicle

and the actual class of the object. Data is generated when the user defines a bounding box or points out the existing one. The latter is generated automatically by using blob tracking method. The results are recorded as text. There is also an option to calculate the final accuracy of the surveillance system when given the ground truth. Verification of the results of video detection is more difficult if several vehicles are being tracked at the same time.

Another tool [13] allows generation of semantic descriptions semi-automatically. The software supports the segmentation stage and labeling prediction. Annotating consists of identification and naming of objects (key objects) and identifying associated actions (events). These descriptions are stored as MPEG-7 XML files.

In [9, 16, 17] ViPER (Video Performance Evaluation Resource) tool is presented. It consists of several separate tools. ViPER-GT is a toolkit (written in Java and available under GPL) for visual data ground truth generation and storing them in custom XML format. The area covered by the annotation possibilities starts at medium-level visual tasks (object detection) and ends at high-level image understanding (events). The exemplary applications, taken from [16], are: text detection and extraction, face detection and identification based on landmark points, and a rule based event detection system. The ground truth data can be compared with a result data using ViPER-PE (Performance Evaluation) tool.

The VIA (Video Image Annotation tool) [5, 6] is a tool with more extensive functionality. Several sophisticated components like object recognition based on low-level features, face detection and motion were built into. SVM (Support Vector Machine) was used for classification. The learning process is preceded by a manual generation of ground truth data. It consists of description of automatically separated segments or indicating and annotating them manually.

Video analysis systems vary greatly and tools for generating ground truth are almost always kept as universal as

it is only possible. But still there is a shortage of universal applications that can be adapted easily to specific tasks. Most tools generates the ground truth of a high level of abstraction, using dedicated grammars, semantic descriptions or predefined scenarios. In addition high level of description corresponds with at most middle level of image analysis: motion detection and prediction or segmentation. There are no tools dealing with measurements based on one click pointing. So there is a need to create a new ground truth generating tool (GTGT), a tool kept as simple as possible, dedicated for a video detector calculating measurement of road traffic [11]. This tool is one of the key elements of the INSIGMA project [3].

## 4 Usefulness of GTGT in the development of video detection software

GTGT is an ancillary tool created in Matlab [4] for the purpose of supporting the development of video detection software. Road traffic data generated by the video detector must be interpreted and compared against ground truth generated by a human operator. This makes it possible to assess the accuracy of results and indicate those fragments of the algorithm that need to be corrected.

The process of improving the video detector and the role of GTGT is presented in Figure 1. It should be emphasized that GTGT is not only ground truth data generator. It also allows to configure the measurement points and can analyze result data from video detector software. Thanks to this GTGT becomes a tool for supporting a video detector development. A cooperation between GTGT and video detector can be divided into several stages (see Fig. 1):

Stage A: View selection

A key element affecting the results generated by the video detector is selecting the appropriate view. The camera should be placed suitably high so that the recorded

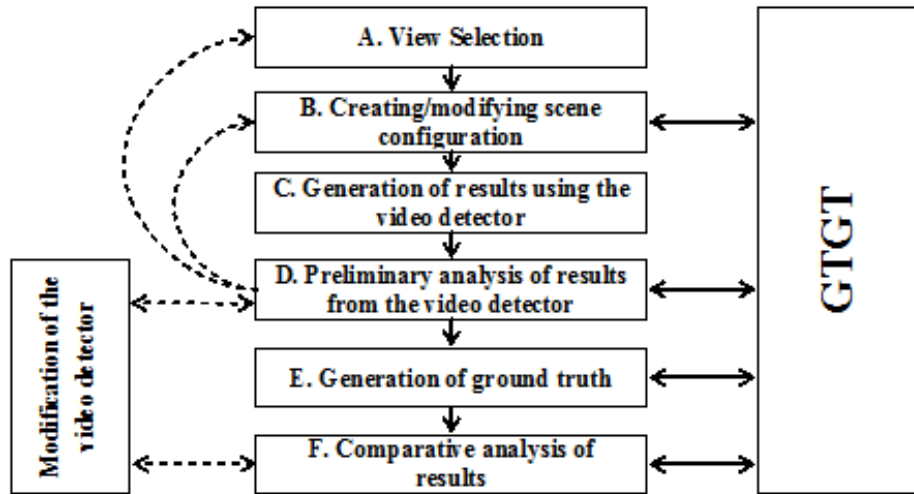


Fig. 1: GTGT in the development of video detection software. Stages A..F are described in the text

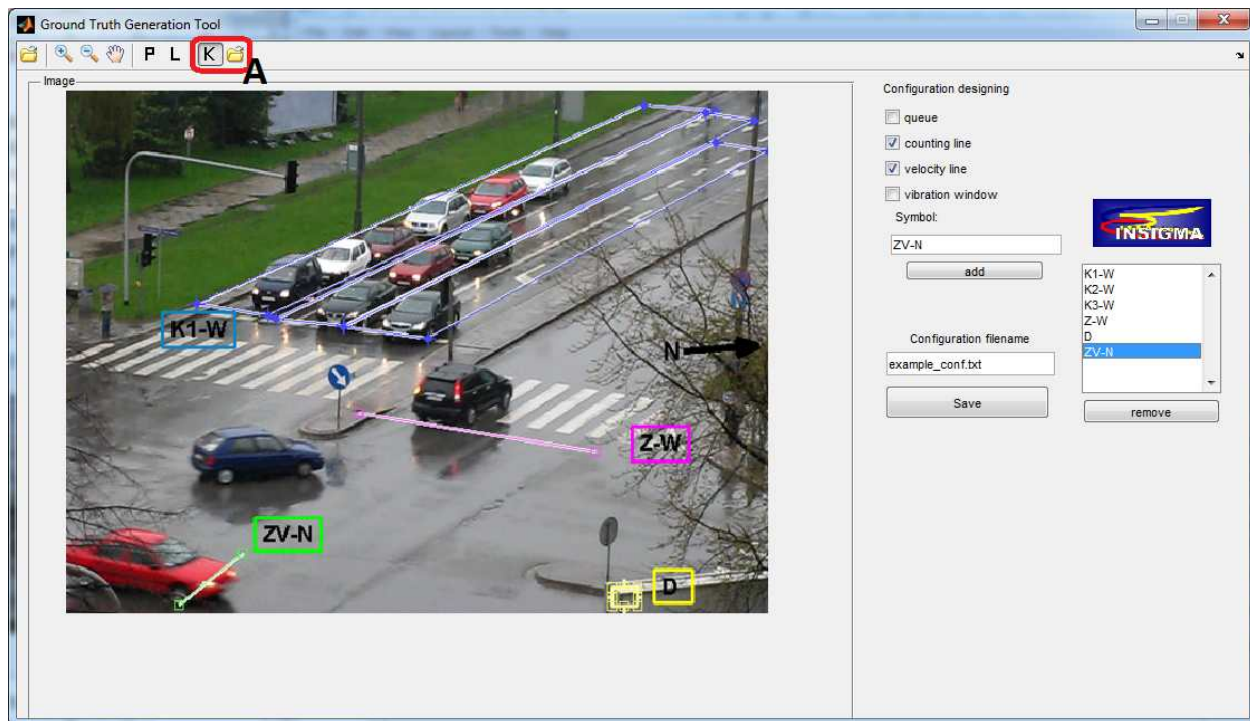


Fig. 2: A screenshot of the GUI of the GTGT when creating/modifying the scene configuration

scene includes the entire junction or a significant part of it. This makes it possible to differentiate between lanes. The image needs to be of suitably high quality, and the camera must not vibrate. The results can be affected by lighting, weather and many other factors. GTGT is not used at this stage, although if necessary it makes it possible to review the recording frame by frame or by incrementing every few frames. Next the appropriate scene configuration needs to be prepared.

#### Stage B: Creating/modifying scene configurations

A configuration is a collection of appropriately arranged lines and measuring fields. On its basis, traffic data is calculated by the video detector and the operator of GTGT. The configuration includes (see Fig. 2):

- Lines for counting vehicles (Z). When a vehicle crosses the given line the appropriate counter is incremented. If the line is positioned correctly, it is possible to count vehicles moving in a given direction after leaving the junction.
- Lines for measuring vehicle speed (V). Multiple detection of characteristic points for the vehicle crossing a given line makes it possible to calculate its speed.
- Fields marking the traffic queue area along the given lane (K). The vehicle furthest away from the centre of the junction along the queue marks its length. The point is sought within each field.
- Rectangles for detecting camera vibrations (D). Calculating and interpreting the difference between frames makes it possible to detect camera vibrations.

An example configuration is shown in Fig. 2. Lines Z and V are usually merged. The name of each element (apart from fields D) contains geographical direction defining where the vehicle is moving from or to.

GTGT makes it possible to create new and modify existing configuration simply and easily. Its elements, such

as the end of the line and points defining polygons, can be freely moved, changed, added or removed. The configuration is written to a text file and read by the video detector and the GTGT. A screenshot of the GUI of the GTGT during creating/modifying scene configuration is shown in Fig. 2. Transition between the configuration mode and data generation mode is possible at any point while using the tool.

#### Stage C: Generation of results using the video detector

A video detector is a program using the configuration and scene analysis algorithms to calculate traffic data. It saves measured values (number of vehicles on line Z, coordinates of the center of gravity of a vehicle crossing line V, and coordinates of the point marking the end of a queue in field K - see Fig. 2) in the text file. This file is then analyzed by the GTGT: visualized in stage D and compared with ground truth in stage F.

#### Stage D: Preliminary analysis of results from the video detector

The results generated by the video detector need to be visually analyzed. GTGT is able to filter events connected with each of the configuration elements. Depending on the element, GTGT superimposes on film appropriate results (events, as described in stage C), thus enabling the assessment of the accuracy of the location of the configuration elements. Moreover it is possible to calculate the number of vehicles counted in a given frame interval, velocity at a given moment or mean velocity, and length of queue at a given moment. By eliminating perspective, the distance measured in pixels can be converted to obtain queue length in meters and velocity in km/h. GTGT facilitates these operations: the operator can define a "dummy queue area" covering the appropriate region of the screen. Locations of the defined points supplemented by the real distances between them are used to compute the perspective transform.

At this stage, prior to the time-consuming stage of generating ground truth, it is possible to return to stage B, that



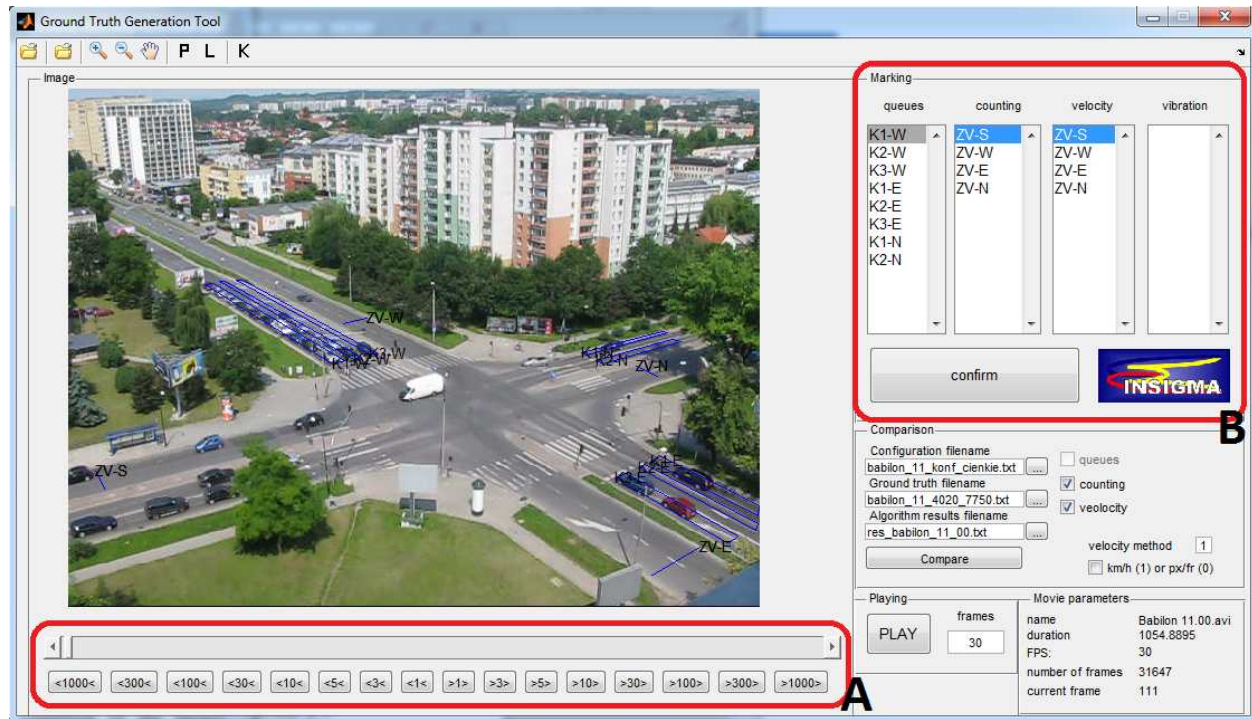


Fig. 3: GUI of the GTGT (configuration elements listed in frame B are visible)

is modifying configuration. If the result is unlikely at any configuration element (e.g. when a measuring line is too long), it is possible to modify the configuration or change the view. It is essential to set the configuration elements correctly, since they can have a significant impact on the video detection results.

#### Stage E: Generation of ground truth

The main purpose of GTGT is to enable the human operator to generate ground truth of traffic parameters simply and quickly. By reviewing a fragment of the recording frame by frame, the operator is able to generate ground truth by performing several clicks following accepted norms. The intuitive and easy to use interface makes it possible to skip between frames (field A, Fig. 3), zoom a region of interest, and play back a sequence of frames. The elements of the configuration are superimposed on the film, as shown in Figs. 3, 4, 6. Ground truth

is generated by selecting the appropriate configuration element from the list and confirming by clicking on the correct point within the frame (see Fig. 3, panel Marking, described as B). Generating ground truth occurs according to the following rules:

- the vehicle is counted when it leaves line Z,
- velocity is measured by clicking the same distinctive point of a vehicle twice: when it arrives on line V and when it leaves,
- queue length is measured when the last vehicle in field K stops. The user clicks the point the furthest from the centre of the junction.

Generated ground truth is written to file similarly to the results from the video detector, making it possible to compare them.

#### Stage F: Comparative analysis of results

The next functionality provided by the GTGT is comparing data from the video detector with ground truth. This can be done graphically or statistically. Graphs generated for each of the configuration elements separately present corresponding data for the selected range of frames. Statistics encompasses mean and/or maximum values. Comparative analysis can be conducted by the GTGT interface (panel Comparison in Fig. 3) or by a dedicated script expanding the abilities of the GUI. The script makes it possible to select the frame range, configuration elements, and method of calculating velocity. The GUI makes it possible to superimpose the video detection results and ground truth directly on film frames while they are being viewed. This wide range of methods of presenting the comparison results makes it possible to pinpoint the location when the results do not agree and identify a cause quickly. Examples of how the tool can be used are described in Chapter 5. The full set of ground truth data is not required for the correct operation; it can be generated for selected fragments of the film and for selected configuration elements. It helps to save time if, for example, just one function of the video detector needs to be tested. If the results obtained at stages D and F are not satisfactory, the video detector needs to be adjusted and checked whether the changes affect the results by returning to stage C. If the configuration has not been changed, stage E is omitted (Fig. 1).

## 5 Experimental results

The usefulness of the GTGT will be presented using two examples. The first describes a situation when the video detection algorithm unnecessarily splits a vehicle into several parts. The second describes methods of calculating vehicle velocity. The situation presented in Fig. 4 shows a bus driving across a measurement line. As well as configuration elements, the frame shows the vehicle's real velocity (above the bus) and the current state of the measuring

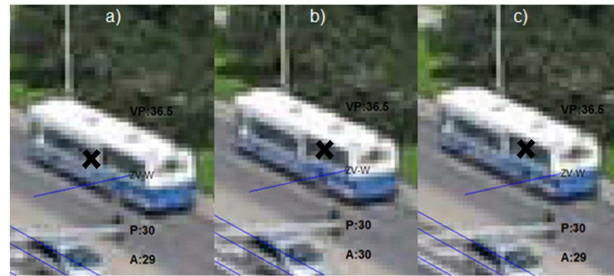


Fig. 4: Stages of counting the bus in frames: a) 6069, b) 6072, c) 6075 (P - ground truth data, A - video detector results)

line ZV-W (number of vehicles counted by the video detector and by the operator). As the bus is long and is of uniform color, the video detection algorithm (using optical flow) detects it as several separate objects. As a result each object increases and decreases the vehicle counter (Fig. 5, frames a, b, c). This is due to the merging of objects that are close together. The graph shows that the video detector has counted one fewer vehicle than the operator generating ground truth. This is likely due to two vehicles crossing the measuring line at the same time. By analysis of a situation where the video detector incorrectly splits or merges vehicles, it is possible to modify the configuration or the algorithm itself in order to reduce the number of such events.

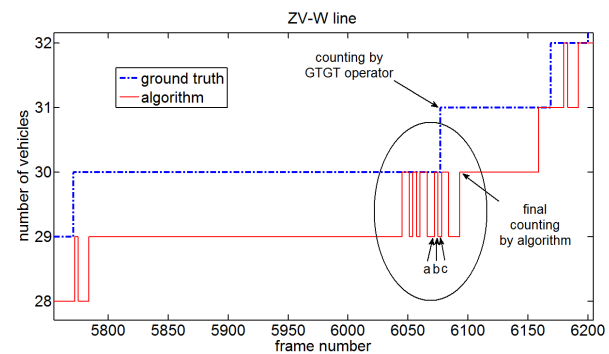


Fig. 5: Comparative graph: counting the bus in frames: a) 6069, b) 6072, c) 6075

The next situation is shown in Fig. 6. The figure shows six frames associated with velocity measurements. In



the first and fourth the sign '+' corresponds to ground truth generated by GTGT (clicking a characteristic point). Frames two to six present the center of gravity of detected object calculated by the video detector on every third frame (starting from frame 6267 - 'x' signs). The video detector tracks the vehicle on every third frame because the algorithm is capable to analyze about 8 frames per second. The VA value in Fig. 6 is the current vehicle velocity calculated using data from the video detector, while VP is the mean actual vehicle velocity based on ground truth. When the vehicle is on the measurement line video detector calculates consecutive positions of its center of gravity. The registration of coordinates of the vehicle's center of gravity on subsequent frames makes it possible to calculate the vehicle's velocity. But the final calculation can be performed in different ways (Fig. 7). GTGT software compares these methods and facilitates the selection of the best.



Fig. 6: Comparing the data needed for velocity calculation: x - tracking the center of gravity by the video detector, + - characteristic point localizations obtained from GTGT

The first method of calculating velocity available in the GTGT is a temporary one (vertical dotted red lines terminated by a circle - see Fig. 7). This is the distance between two consecutive measurements of centers of gravity

of vehicles divided by three (because the measurements are performed every third frame). It is calculated and presented always because it is the only information about velocity generated by video detector. Based on these data GTGT makes it possible to calculate mean velocity for a given vehicle. To calculate it, temporary velocities are grouped for each vehicle. The velocity can be calculated either as an arithmetic mean (horizontal pink dotted line in Fig. 7) or median (horizontal green dot-dashed line in Fig. 7).

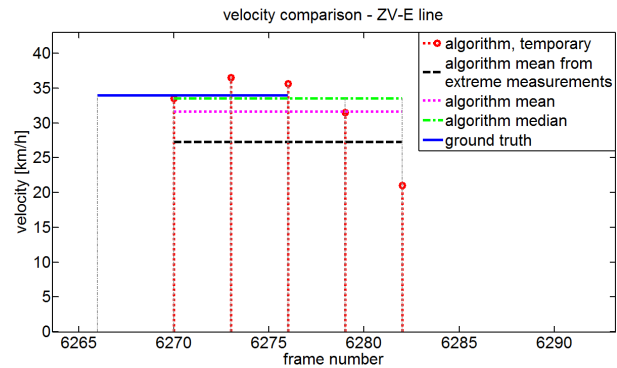


Fig. 7: Comparative graph: calculating vehicle velocity in subsequent frames (6267-6282)

Another method for calculating mean velocity is calculating the difference between the extreme measurements obtained for a given vehicle and dividing them by the number of frames between them (horizontal dashed black line in Fig. 7). Real velocity is calculated from ground truth data using the same method. The above description concerns the velocity calculated as pixels per frame. Knowing the FPS (frame per second) parameter of the recording/stream and eliminating perspective make it possible to convert this value to kilometers per hour. The visual comparison of the results of calculating vehicle velocity (see Fig. 7) allows to investigate which is the most accurate for the given view or the given measurement line, and applying it as appropriate. Using graphs such as those shown in Fig. 6 and 7 makes it possible to track the results of video detection algorithm. Looking at a graph

of velocity for a given configuration element, it is easy to pinpoint situations when the velocity calculated using video detector differs significantly from actual velocity. By analyzing the recording frame by frame, it is possible to investigate the shifting of the vehicle's centre of gravity and find potential reasons for this difference. Using the specific example shown in Fig. 6 and 7, the most accurate result was obtained for a median (of temporary velocities) as a method of calculating mean velocity. Differences between mean values and actual velocity (shown in Fig. 7) come from shifting back of the vehicle's centre of gravity (Fig. 6 e-f), due to occlusion of front of the vehicle by a billboard. Correction of the configuration element position will result in a reduced error value. In spite of error detection and attempts to eliminate them, there are discrepancies between the results obtained by the algorithm and the GTGT. They come from the specificity of operation of the algorithm itself. The vehicle's centre of gravity is detected on every third frame, while the GTGT marks a carefully selected distinctive point of the vehicle twice (see Fig. 6). Both measurements are graphically presented by GTGT thus allowing to find causes of errors.

## 6 Summary

In the paper a method and a tool for generating ground truth for video detection algorithms were presented. The concept and implementation of a Matlab application for generating ground truth and comparing the results against measurement obtained using the video detection algorithm were discussed. In comparison with other reviewed tools, the GTGT makes it possible to preview video recordings quickly and easily, define and modify regions of interest (scene configuration), generating the ground truth data and observe results of comparison superimposed on movie frames. Analysis of graphs comparing the results of the algorithm and the ground truth allows to easily locate the parts of the film in which the dis-

crepancies were detected. The presented examples show usefulness of the GTGT. The modular construction of the application makes it easy to expand and adapt to the verification of other tasks, e.g. choosing suitable frame resolution, dealing with camera vibrations, lighting changes or bad weather conditions (heavy rain, fog). In order to reduce its dependence on MATLAB, there are plans to rewrite the code in C++.

## Acknowledgment

The work was financed by the AGH UST grant no. 11.11.120.612 and co-financed by the European Union from the European Regional Development Fund, as part of the Innovative Economy Operational Program INSIGMA no. POIG.01.01.02-00-062/09.

## References

- [1] -, (2015). Autoscope, <http://www.imagesensing.com/products/video-detection.html> (last access: March 2015)
- [2] -, (2012). Hitech Projects, <http://www.hitech-projects.com/euprojects> (last access: July 2012)
- [3] -, (2015). INSIGMA Project, AGH, Krakow, 2012, <http://insigma.kt.agh.edu.pl> (last access: March 2015)
- [4] -, (2015). MATLAB and Simulink for Technical Computing, MathWorks, <http://www.mathworks.com> (last access: March 2015)
- [5] -, (2009). CERTH-ITI Concept extraction, [http://vitalas.ercim.eu/images/stories/pdf/web\\_cert\\_iti.pdf](http://vitalas.ercim.eu/images/stories/pdf/web_cert_iti.pdf) (last access: March 2015)
- [6] -, (2010). Video Image Annotation tool (VIA) from CERTH-ITI <http://sourceforge.net/projects/via-tool/> (last access: March 2015)

- [7] Ambardekar, A., Nicolescu, M., Dascalu, S. (2009). Ground Truth Verification Tool (GTVT) for Video Surveillance Systems, *Advances in Computer-Human Interactions ACHI '09*. Second International Conferences on, 354-359
- [8] Bonneson, J., Abbas, M. (2002). Intersection Video Detection Manual, *Report FHWA/TX-03/4285-2*, Texas Transportation Institute, The Texas A&M University System, College Station, Texas 77843-3135
- [9] Doermann, D., Mihalcik, D. (2000). Tools and techniques for video performance evaluation, *Pattern Recognition, 15th International Conference on*, 4, 167-170
- [10] Fisher, R.B. (2004). PETS04 Surveillance Ground Truth Data Set, *Sixth IEEE Int. Work. on Performance Evaluation of Tracking and Surveillance (PETS04)*, 1-5
- [11] Glowacz, A., Mikrut, Z., Pawlik, P. (2012). Video Detection Algorithm Using an Optical Flow Calculation Method, *In: A. Dziech and A. Czyżewski (Eds.): Proc. MCSS 2012, CCIS 287*, Springer-Verlag Berlin Heidelberg, 118-129
- [12] Jaynes, C., Webb, S., Steele, R., Xiong, Q. (2002). An Open Development Environment for Evaluation of Video Surveillance Systems, Metaverse Lab, Dept. of Computer Science, University of Kentucky
- [13] Lin, C. Y., Tseng, B. L., Smith, J. R. (2003, July). VideoAnnEx: IBM MPEG-7 annotation tool for multimedia indexing and concept learning. In *IEEE International Conference on Multimedia and Expo* (pp. 1-2)
- [14] Maggio, E., Cavallaro, A. (2011). Video Tracking: Theory and Practice, Wiley and Sons
- [15] Medina, J.C., Benekohal, R.F., Chitturi, M. (2009). Evaluation of video detection systems, Volume 1: Effects of configuration changes in the performance of video detection systems Volume 2: Effects of illumination conditions in the performance of video detection systems. *Research Reports ICT-08-024 2008, ICT-09-046 2009*, University of Illinois at Urbana-Champaign
- [16] Mihalcik, D., Doermann, D., Ahuja, R. (2011). ViPER: Video Performance Evaluation Resource, University of Maryland, Institute for Advanced Computer Studies, Language and Media Processing Laboratory, <http://lampsrv02.umiaccs.umd.edu/projdb/project.php?id=18> (last access: March 2015)
- [17] Mihalcik, D., Doermann, D. (2003). The design and implementation of ViPER. *University of Maryland*
- [18] Rhodes, A.E., Smaglik, J., Bullock, D.M. (2006). Vendor Comparison of Video Detection Systems, *Publication FHWA/IN/JTRP-2005/30. Joint Transportation Research Program*, Department of Transportation at Purdue University, West Lafayette, Indiana