A NEW TECHNIQUE FOR OBJECT DETECTION AND TRACKING AND ITS APPLICATION TO ANALYSIS OF ROAD SCENE

K. I. Kiy [1] *, D. A. Anokhin [1]

[1] Keldysh Institute of Applied Mathematics of Russian Academy of Sciences, 80626 Moscow, Russian Federation
konst.i.kiy@gmail.com

KEY WORDS: Computer Vision, Image Understanding, Object Detection and Recognition, Autonomous Vehicles, Road Scenes Recognition

ABSTRACT:

In this paper, a new technique for real-time object detection and tracking is presented. This technique is based on the geometrized histograms method (GHM) for segmenting and describing color images (frames of video sequences) and on the facilities for global image analysis provided by this method. Basic elements of the technique that make it possible to solve image understanding problems almost without using the pixel arrays of images are introduced and discussed.

A real-time parallel software implementation of the developed technique is briefly discussed. This technique is applied to solving problems of road scene analysis. The application to finding small contrast objects in images, like traffic signals and signal zones of vehicles is given. The developed technique is applied also to detecting other vehicles in the frame. The results of processing different frame of videos of road scenes are presented and discussed.

1. INTRODUCTION

Considerable progress in object detection and recognition has been achieved during the last decade. The application of CNN deserves special attention (Badrinarayanan, Kendall, Cipola, 2017), (Kendell, Gal, Cipola, 2018) (see also the references in these papers and multiple papers that cite these works). However, in all systems based on learning, there are some problems in automatic testing the validity of the obtained results and in revealing the interaction of the found objects (global image analysis). This means that the development of methods based on the classical pattern recognition approach is also of interest. This is especially true for real-time methods able to deal with HD-video. In addition, it is of special interest to develop methods that make the global image analysis possible, since it is not only interesting to detect objects, but it is important to study their interactions as well. In this paper, we propose several applications of the GHM (Kiy, 2018), (Kiy, Anokhin, Podoprosvetov, 2020), in object detection, recognition, using the ability of the method to perform the global image analysis.

2. PATTERN RECOGNITION METHODS BASED ON GEOMETRIZED HISTOGRAMS METHOD

2.1 Brief Description of GHM Capabilities for Describing Images

The GHM deals with images and frames of video sequences divided into parallel horizontal or vertical strips of the same width. The main operations of processing large pixel arrays of high-resolution images and frames of video sequences are performed separately for each strip. This idea makes it possible to parallelize easily the programs implementing the processing of pixel arrays. In addition, this allows us not to lose very small important contrast objects usually lost in classical segmentation methods. Instead of dealing with clusters in the image of image array points in a certain feature space to start the segmentation process (typically employed in classical segmentation methods), the GHM uses an approximate description of the geometry of the distribution of values of the function specifying the image. This makes this method able to detect very small contrast object in images like traffic lights, signal zones of vehicles, traffic signs, construction cones, etc., even from a large distance.

In each strip, the distribution of values of the vector function specifying color images ((R, G, B), (H, S, I), etc.) is approximately described by the geometrized histogram of this strip. Assume that the boundary lines of n-th strip $\boldsymbol{St}_n$ are parallel to the horizontal (vertical) axis **Os** of the image plane. The geometrized histogram is obtained by projecting pixels of $\boldsymbol{St}_n$ onto the lower boundary of the strip with separating colors and intensities. We can assume that the obtained projection belongs to **Os**. Due to the discrete nature of the pixel image the projection is determined by a set of localization intervals on **Os**. Each localization interval $\mathbf{Int}_k = [beg_k, end_k]$ on **Os** is described by the range and the mean value of hue $\Delta_H^k = [H_{min}^k, H_{max}^k]$ and $H_{mean}{}^k$, the range and the mean value of saturation $\Delta_S^k = [S_{min}^{kz}, S_{max}^k]$ and S_{mean}^k, and the range and the mean value of grayscale intensity $\Delta_I^k = [I_{min}^k, I_{max}^k]$ and I_{mean}^k. All pixels with these intensity-color characteristics lying in the strip over $\mathbf{Int}_k$ are projected on it. In addition, $\mathbf{Int}_k$ of the geometrized histogram has the cardinality $Card^k$, which is equal to the number of pixels with the described intensity-color characteristics projected on it (Kiy, 2018), (Kiy, Anokhin, Podoprosvetov, 2020). It is clear that the geometrized histogram approximately describes the intensity-color distribution in $\boldsymbol{St}_n$. The geometrized histogram contains too many intervals to solve image understanding problems quickly. Using clustering procedures, $\mathbf{Int}_k$ that have similar intensity-color characteristics and that are close as intervals on **Os** are joined (Kiy, 2018), (Kiy, Anokhin, Podoprosvetov, 2020). The results of this clustering are called color bunches. Each color bunch b is described by the localization interval $\mathbf{Int}_b = [beg_b, end_b]$ on **Os**, the range and the mean value of hue $\Delta_H^b = [H_{min}^b, H_{max}^b]$ and $H_{mean}{}^b$, the range and the mean value of saturation $\Delta_S^b = [S_{min}^b, S_{max}^b]$ and S_{mean}^b , and the range and the mean value of grayscale intensity $\Delta_I^b = [I_{min}^b, I_{max}^b]$ and I_{mean}^b. Additionally, each color bunch has the cardinality $Card^b$, which is approximately equal to the number of pixels in the strip with the

* Corresponding author

color characteristics within the limits of the threshold values attached to each color bunch that belong to the part of the strip bounded by the localization interval $[beg_b, end_b] \subset \mathbf{Os}$. An additional characteristic of b is its density $dens_b = Car_b$ / length($[beg_b, end_b]$). Color bunches of all strips of the whole image generate the graph of color bunches *STG*, which gives a concise image description. In the same strip, color bunches with adjacent localization intervals are joined by an edge. In adjacent strips, color bunches with intersecting localization intervals are also joined by an edge. This graph describes the geometry and the color distribution in the image rather exactly. Fig. 1 shows all color bunches of the image in the right-hand part of Fig. 1 superimposed on middle lines of strips of its grayscale component. Figure 1 demonstrate that *STG* contains the main information of the original image.

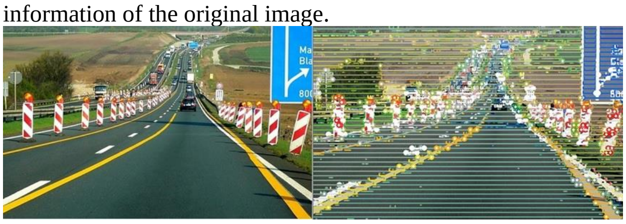

Figure 1. All color bunches of the right-hand image.

Definition. Color bunches that have a maximum possible density among all the bunches that passes through one of its points are called dominating.

It is clear that color bunches corresponding to parts of regions in the image are dominating in their parts of the strip.

To provide capabilities of global image analysis, a "search lattice" SearchLat(*STG*) is constructed on the set of dominating color bunches. This lattice makes it possible to perform finding adjacent objects for any object in the image and to compare the geometric and intensity color characteristics of the object and its surrounding objects (Kiy, 2018). All bunches of SearchLat(*STG*) are dominating and have the maximum density at one of their points among all bunches passing through this point. In each strip, elements of SearchLat(*STG*) determine a linearly ordered set of adjacent (in the sense of their localization intervals) color bunches that cover the corresponding strip in some sense.

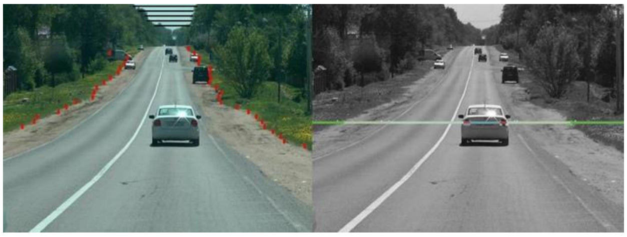

Figure 2. Bunches of SearchLat(*STG*) for a fixed strip.

In Fig. 2 the localization intervals of the bunches belonging to SearchLat(*STG*) for a fixed strip are superimposed on the middle line of the strip of the grayscale component of the image. Each localization interval is painted using the intensity color characteristics of the bunch. These bunches correspond to different parts of regions in the image (grass field, road, roadsides, body of the car, car signal zones). In (Podoprosvetov and Kiy, 2019), a program with graphical interface (and with a manual) that can construct the geometrized histogram and *STG* bunches for any BMP image file can be found and download.

2.2 Segmentation in *STG*

Segmentation performed in *STG* determines a concise description of real parts of the main objects in the image. Color bunches well represent parts of objects in the image in the corresponding strip. To reconstruct global objects in the image based on the set of color bunches, the concepts of left and right germs of global objects (left and right contrast curves) were introduced (Kiy, 2015). To segment the image, a bipartite graph of left and right germs of global objects or contrast curves *LRG* was determined (Kiy, 2015). If the image is divided into horizontal strips, then informally a left or right germ of a global object is a chain of dominating color bunches in adjacent strips with similar intensity-color characteristics. Note that for left (right) germs left (right) ends of the localization intervals of their color bunches vary continuously from strip to strip, and the left (right) adjacent (in the sense of localization intervals) bunches of members of the left (right) germ have contrast color characteristics with it. These chains are constructed bottom-up, passing from strip to strip (Kiy, 2015). By global objects we mean objects that belong to a chain of several adjacent strips. Figure 3 demonstrates examples of left and right germs of global objects.

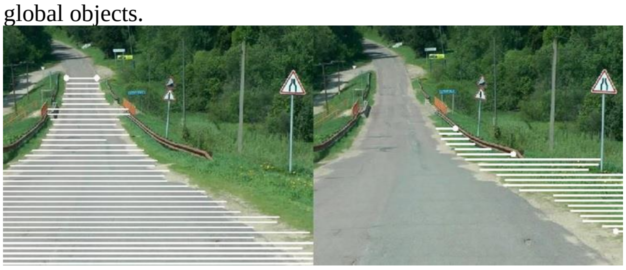

Figure 3. Continuous germs of global objects.

Another type of segmentation is applied for finding long thin objects like road marking. This type of segmentation involves small dominated objects and is less connected with similarity of intensity-color characteristics. The purpose of this is to follow road marking even in shadows. In more detail it is presented in (Kiy and Dosaev, 2019) and (Kiy and Dosaev, 2020).

2.3 Parallel Program Implementation of the GHM

The main operations of the segmentation in *STG* and image understanding systems based on it are executed without using the image pixel array — only using the designed *STG*.

In (Kiy, Anokhin, Podoprosvetov, 2020), a multithreaded implementation of *STG* was constructed, which allows one to enhance the operation speed, which was rather big already, up to the record standards. It is worth noting that even for a laptop of a mean price category, produced in 2017, having a processor Intel Core i7 with four physical cores, the execution time for processing one frame of a video sequence of HD of 1920x1080 pixels is about 30 ms. For this time, several image understanding tasks, such as detection of the road, road sides, road marking, finding sky region, detection other vehicles on the road, etc., are solved. This is explained by the fact that all these tasks are solved using *STG* with minimal involvement of pixel arrays. This means that we are able to solve image understanding problems with performance of 30 fps for frames HD-video. For multi-processor computers even the performance of 100 fps can be reached.

2.4 Vision Tasks that Can Be Solved by the Proposed Methods

Below, a list of tasks can be found. It is necessary to note that we include in the list only the tasks for which at least preliminary experimental encouraging results with a developed software system have been already obtained. Really the future list will be much more extended. It seems real to apply this technique in motion analysis. But it will be the topic of the next publications. The list of investigated (at least partially) problems is as follows:

1. Detection of the road region and roadsides.
2. Detection of the sky region.
3. Detection and understanding road marking (permanent and temporary).
4. Detection of traffic lights and traffic signs.
5. Detection of signal zones of vehicles and the analysis of their state.
6. Detection of vehicles on the road and integrated analysis of road scene.

It is also worth noting that possible applications of the developed methods are not restricted to road scene analysis. This was only a convenient field for demonstration of capabilities of the developed methods.

Problems 1, 2 were considered in (Kiy, 2018) and (Kiy, 2018a), In (Kiy and Dosaev, 2019) and (Kiy and Dosaev, 2020), the investigation of Problem 3 with the application of the GHM was initiated. In the next section, we will briefly present some additional results on this topic, necessary for solving Problems 5 and 6, presented in the next section.

3. DETECTION OF VEHICLES ON THE ROAD AND INTEGRATED ROAD SCENE ANALYSIS

3.1 Detection of Small Contrast Objects in the GHM

Assume that we deal with a division of the image into horizontal strips. Since the GHM is based on a description of the geometry of value distribution of the vector function specifying the image, in each strip, any even very small contrast object different from a segment of a vertical line in a strip gives an interval of the geometrized histogram of this strip. If the resolution of the image is rather big, then even distant small contrast objects like traffic signals, signal zones of vehicles, signal lamps of flying vehicles, traffic signs, a part of road marking in the strip give intervals of the geometrized histogram of this strip. As a rule, all mentioned objects have a sufficiently big color saturation *S* (traffic signs, signals, colored road marking, signal lamps of flying vehicles, signal zones of vehicles) and the hue *H* lying within a certain interval. As a rule, they have a rather big intensity. For example, usually parts of white road markings in the strip have the intensity bigger that the intensities of the surrounding objects. It is important that the intervals of the geometrized histogram corresponding to these objects generate color bunches of *STG* to be involved in the process of detecting using only *STG* without dealing with pixel array of images. To provide the fulfilment of this requirement, intervals with local maximum saturation and intensity among all adjacent intervals, independently of their cardinality and density, are included in the seeds for the clustering procedure for constructing color bunches (Kiy, Anokhin, Podoprosvetov, 2020). Of course, they may determine dominated color bunches, as the traffic sign bunch in Fig. 3. However, using the search among all bunches, the bunches corresponding to traffic lights can be easily found. Methods for finding road markings (permanent and temporary) based on GHM and examples of their operation can be found in (Kiy and Dosaev, 2019) and (Kiy and Dosaev, 2020).

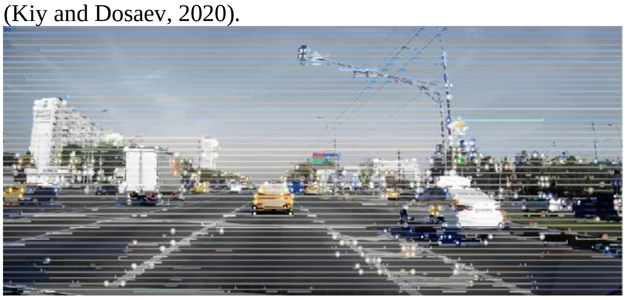

Figure 3. Color bunches corresponding to a scene with traffic lights.

Additional algorithms that investigate the surrounding objects of the segmentation on *STG* for a selected candidate for a traffic signal and its location conclude that the selected object represents the traffic signal and surround it by a rectangle of the corresponding color in the image. The result of operation of such an algorithm is presented in Fig. 4. The detailed publication on this topic is under preparation.

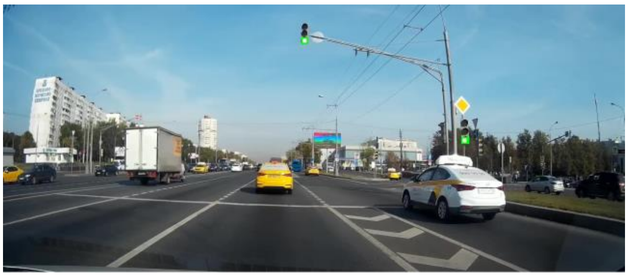

Figure 3. Traffic lights found in the road scene.

Examples of processing video sequences using these methods can be found in (Podoprosvetov and Kiy, 2019).

Let us describe the approach for finding signal zones of vehicles based on GHM. First, we find candidates for bunches produced by signal zone of vehicles moving or staying in front among color bunches of *STG*. Second, we try to find approximately the vehicle contours that contain some of the candidates. For this purpose, the concepts of almost vertical and almost horizontal lines are introduced in the next subsection. To find the candidates that may represent signal zone of vehicles, we do the following:

1. At the first stage of the algorithm, isolated color bunches in strips with a sufficiently big saturation *S* and the hue within a certain range of red color are found.
2. At the second stage, continuous chains of these bunches in several adjacent strips are found.
3. At the third stage, those isolated bunches and chains of bunches are tested for being over the road or in its close vicinity.
4. At the fourth stage, it is tested whether there are special objects called almost vertical curves on SearchLat(*STG*) that locate over the road and in its close vicinity and have adjacent candidates for signal zones.
5. At the fourth stage, we find almost horizontal curves in their vicinity.

Since any vehicle is a rather big object, it induces several regions in the image, and, consequently, it induces in strips dominating color bunches. These bunches are included in SearchLat(*STG*) since without them the covering of the strip in *STG* is impossible. The right-hand image form Fig. 1 illustrates well this statement. Usually, a vehicle determines several regions contrast to the surrounding objects (road, roadsides, other vehicles, sky, vegetation, etc.). Horizontal and vertical boundaries of the vehicle in a sense are "almost" horizontal and vertical lines. This means that these boundaries should induce "almost" vertical (horizontal) lines of SearchLat(*STG*), which "covers" *STG*. In addition, the boundary lines (horizontal or vertical) induced by a vehicle have non-empty intersection and encompass characteristic places of the vehicle, such as signal zones, windows, etc. In addition, almost vertical boundary lines cannot start far from the vanishing point of the road. In the next subsection we try to axiomatize the following observations in terms of *STG*.

Note that taking into account signal zones of vehicles, we implicitly supposed that we deal with vehicles in front. For oncoming vehicles instead of signal zones, we take into account headlights and fog-headlights. They provide isolated color bunches with maximum local grayscale intensity. In the same way, we can construct the clustering procedure for finding color bunches, so that even small intervals of local maximum intensity determine color bunches.

3.2 Concepts of Almost Vertical and Horizontal Lines in SearchLat(*STG*)

Suppose that we deal with a division of the images into horizontal strips. Let us present an algorithm for constructing contours on SearchLat(*STG*) and define which contours are "almost" vertical lines on *STG*. These contours are generated by left (right) ends of localization intervals of basic color bunches generating the search lattice SearchLat(*STG*). To construct these contours efficiently, we generate for each strip of the division an array *LatLoc*[*i*] of length DimX/*l*, where DimX is the vertical dimension of the image array in pixels, and, *k* is a dimension reduction coefficient (e.g., *l* = 4). At each point *i*, *l*[*i*] = *b*, where *b* is the number of the basic color bunch from SearchLat(*STG*) whose localization interval passes the point *ki* of the copy of the horizontal axis **Os** on which the color bunches of these strip are located. Consider a certain strip St_n. Let b_k be a certain bunch, $b_k \in$SearchLat(*STG*) with localization interval [beg_{bk}, end_{bk}]. Strips are numbered bottom-up. Consider the next strip St_{n+1}. Using the array *LatLoc* [*i*] of the next strip, we find the basic color bunch that passes points beg_{bk} / l (end_{bk} / l) in the adjacent strip. Then, moving along the search lattice SearchLat(*STG*) in the corresponding strip, we find the basic b_{k+1}, such that the distance between beg_{bk} (end_{bk}) and $beg_{b(k+1)}$ ($end_{bk(k+1)}$) has a minimum value. If this distance is less than a certain constant bounding the inclination angle, the contour is extended to b_{k+1}. In extension, using left ends (right ends), left (right) contours are obtained. For a contour of length *d* two numerical characteristics are introduced: 1. the maximum deviation from the vertical direction $dmax = max_i$ abs($end_{bk+i} - end_{bk+i+1}$) and 2. total deviation from the vertical direction $dtot$ = abs($end_{bk} - end_{bk+d}$)/*d*.

Definition 1. A contour is close to be perpendicular to **Os** or is an "almost" vertical line in SearchLat(*STG*) or an "almost" vertical lines in the image, if *dmax* and *dtot* are bounded by a relatively small constant.

Definition 2. If we deal with a division into vertical strips and **Os** is the vertical axis, then "almost" vertical lines in the corresponding SearchLat(*STG*) are called "almost" horizontal lines in the image.

Figure 4 presents an example of *STG* image with contours constructed on SearchLat(*STG*) for horizontal **Os**. Points of contours are painted blue.

If two "almost" vertical lines bound a real object in the image, they have to be connected in space, e.g. their vertical projection have to have a rather strong intersection. Let us introduce a technique that allows one to determine vertical lines with intersecting vertical projections. Let *L* = {1, …, N} be the set numbering the strips of the image division. We assign to each contour *K* on SearchLat(*STG*) a segment $I_K \subset L$, the ends of which are determines by the first and last strips of this contour *fs*(*K*) and *ls*(*K*).

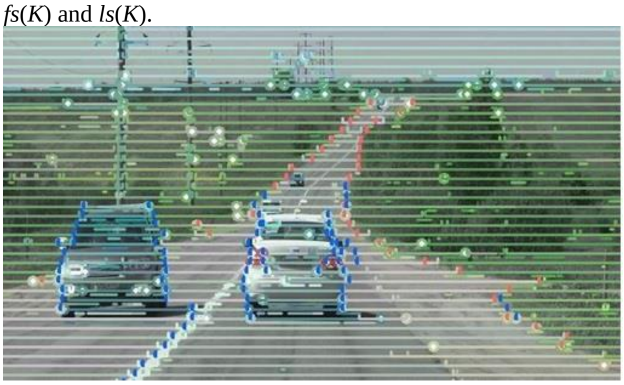

Figure 4. Contours in the *STG* image.

For any two contours K_1 and K_2, the segment of their intersection is $I_{12} = I_{K1} \cap I_{K2}$. Let us attach to intersecting contours the following two numbers: 1. c_1 is the fraction of I_{12} in I_{K1}; 2. c_2 is the fraction of I_{12} in I_{K2}. There are four types of intersection: 0. $c_1 \geq ½$, $c_2 \geq ½$; 1. $c_1 \geq ½$, $c_2 < ½$; 2. $c_1 < ½$, $c_2 < ½$; 3. $I_{12} = \varnothing$. Denote by d_{12} the distance along the axis **Os** between the projections of first elements of contours onto **Os**. Contours are numbered bottom-up and left-right according to their first elements. It I_{12} is not empty, then contours are intersecting. For any contour *K*, we find the first and last contours that intersect *K* (in bottom-up motion) *f*(*K*) and *l*(*K*), the closest left and right neighbors *cll*(*K*) and *clr*(*K*) in the sense of distance d_{12} with the intersection of type 1, and the left and right neighbors *mll*(*K*) and *mlr*(*K*) that have maximum intersection with *K* in the sense of c_1. To take into account the fact that the contour was broken because of segmentation errors the upper and lower closest neighbors *up*(*K*) and *lw*(*K*) are also determined. Using this, we can eliminate isolated contours that cannot bound objects like vehicles.

Then contours that have candidates for signal zones in close vicinity to the right or left of them are considered. The connections between signal zones for cars and trucks are different. Cars typically have signal zones on the main body, while trucks usually have them below it. This is taken into account in connections between "almost"vertical lines and candidates for signal zones.

Since straight contours are very important, algorithms for detecting them is proposed. For example, trucks very often have straight vertical boundaries of their main body. In addition, false contours produced by road marking or parts of the road or roadside (as in Fig. 4) may be straight. In (Kiy, 2018), a method based on the analysis of histograms of inclination angles of segments joining points of a chain points in the adjacent strip (like our contours) stable to segmentation errors was developed. This method also describes the total inclination angle of the contour. Using this method, we eliminate straight contours with rather big inclination angle.

Frequently due to uneven surface of the vehicle, there are "almost" vertical lines that are connected with signal zones located inside the vehicle body (as in Fig. 4 near the license plate). It is necessary to find the boundary “almost” vertical lines that bound the body of the car. To solve this problem, the search lattice SearchLat(*STG*) is employed. For this purpose, we use the structures on the search lattice *traceL*$_K$(SearchLat(*STG*)) and *traceR*$_K$(SearchLat(*STG*)), on which for each left or right contour *K* the trace of it (color bunches of the search lattice through which *K* passes) is specified.

The algorithm consists of two stages: 1. At the first stage, moving along the several strips adjacent to the strip, on which the bunch of a signal zone is located, and finding possible other contours, we find the extreme left of right contours connected with signal zones using *traceL*$_K$(SearchLat(*STG*)) or *traceR*$_K$(SearchLat(*STG*)); and 2. Additional almost vertical lines inside the vehicle body that are not extreme left and right lines connected with signal zones are eliminated.

At the step 2, we use possible metric relations for distances between the extreme left and right almost vertical lines explained in the next subsection.

To restrict the possible region of the vehicle, we cut a small image containing the region confined by a pair of “almost” vertical lines and construct “almost” horizontal lines in this cut image. As the rule, these lines bound upper and lower part of the vehicle. They may bound car windows or parts of the truck body. The lower part Fig. 5 presents a set of “almost” horizontal lines of the image in the upper part of this figure.

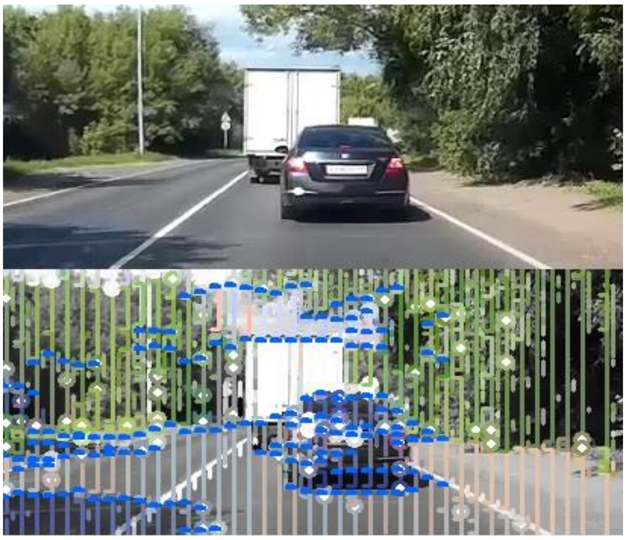

Figure 5. “Almost” vertical lines of the image.

3.3 Integrated Analysis of Road Scenes

Since, before finding vehicles, several additional tasks, such as finding the road, roadside, the road markings on the road, the sky region have been solved, we have additional information that can help in finding possible vehicles on the road. It depends on the situation on the road, and we cannot guarantee that all these problems were correctly solved completely, because of occlusion and inhomogeneous illumination of the road scene. We have to test that the solutions fit each other. Of course, in many cases, especially, if the condition of the road marking is rather good or the road region and roadside boundaries were found correctly, we can find the “vanishing” point of the road and the approximate width of the lane in pixels in each strip before our car. The boundary of the sky region and the “vanishing” location should fit each other. Figure 6 demonstrates an example of processing in which the all component of the found data (temporary road marking, boundary of the green roadside, and the boundary of sky region) fit each other. Therefore, we can conclude about the approximate width of the vehicle in front. This is a hint for joining “almost” vertical lines that can bound it. We can also eliminate from the consideration “almost” vertical lines that start sufficiently higher than the strip of the “vanishing point”. As an additional argument we can investigate segmentation objects located between the selected “almost” vertical lines and to study their color and shape. Therefore, we can conclude about the approximate width of the vehicle in front at the strip where the first points of the selected pair of vertical lines are located. This is a hint for joining “almost” vertical lines that can bound it. We can also eliminate from the consideration “almost” vertical lines that start sufficiently higher than the strip of the vanishing point. As an additional argument we can investigate segmentation objects located between the selected “almost” vertical lines and to study their color and shape.

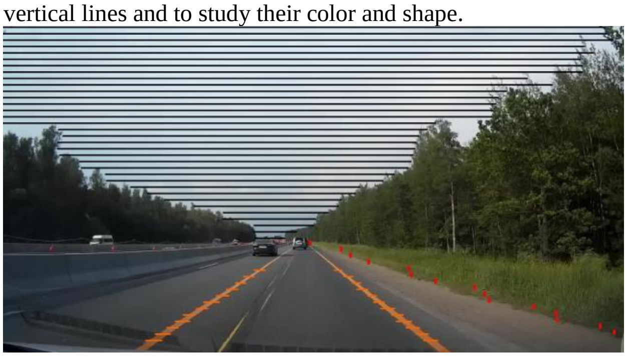

Figure 6. Integrated analysis of a road scene.

4. SOFTWARE IMPLEMENTATION AND THE FUTURE WORK

The algorithms described in this paper have been implemented by a program complex written in C++ and operating under Windows and Linux environments. The structure of the software complex was described in (Kiy, Anokhin, Podoprosvetov, 2020), where a table, estimating the performance of this software system can be found. This performance makes it possible to solve image understanding tasks in real time for HD-video on standard multi-core (multiprocessor) computers. The examples in figures presented in this paper were obtained by a special version of the software complex with a visualization program using MFC, or taken from protocols of processing different video sequences of road scenes. Examples of protocols of processing different video sequences of road scenes can be found in (Podoprosvetov and Kiy, 2019). At present, an additional block that will be able to make conclusions of the qualitative nature of vehicles in front and to reveal the future behavior of it (analysis of braking and turn signals) is being debugged and tested, which will the subject of the next publication.

REFERENCES

Badrinarayanan, V., Kendall, A., Cipola, R., 2017. Segnet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. PAMI*, 39(12), 2481–2495.

Kendell, A., Gal Y., Cipola, R., 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *IEEE conference on computer vision and pattern recognition* (ICVPR-2018), 7482-7491.

Kiy, K.I., 2015. Segmentation and detection of contrast objects and their application in robot navigation. *Pattern Recognit. Image Anal*., 2015, 25(2), 338–346.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLIV-2-W1-2021-119-2021 | © Author(s) 2021. CC BY 4.0 License.

doi.org/10.1134/ S1054661815020145.

Kiy, K.I., 2018. A new method of global image analysis and its application in understanding road scenes. *Pattern Recognit. Image Anal*., 2018, 28(3), 483–494. doi.org/10.1134/S1054661818030100.

Kiy, K.I., Anokhin, D.A., Podoprosvetov A.V., 2020. A software system for processing images with parallel computing. *Programming and Computer Software*, 46(6), 406-417. doi.org/10.1134/S0361768820060043.

Kiy, K., Dosaev, R., 2020. Global image analysis: detection and recognition of basic informative elements of road scenes. *CEUR Workshop Proceedings*, 2665, 100-104.

Kiy, K., 2018a. An image understanding system based on the geometrized histograms method: finding the sky in road scenes. *CEUR Workshop Proceedings*, 2210, 291-299.

Podoprosvetov, A. V. and Kiy K.I., 2019. AvtoNiva Project, 2019.
http://project1054516.tilda.ws/ (1 March 2021).