

Efficient Object Localization Scheme Based on Vanishing Line in Road Image for Autonomous Vehicles

Bongkyo Moon^{1*}, Jiwon Choi², Juehyun Lee³, Minyoung Lee³

¹Department of Computer Science and Engineering, Dongguk University, Seoul, South Korea

²Graduate School of Information Security, KAIST, Daejeon, South Korea

³Electronics and Telecommunications Research Institute (ERTI), Daejeon, South Korea

Email: *bongkyo.kmoon@gmail.com, eternalklaus@naver.com, juehyun@erti.re.kr, sharav@erti.re.kr

How to cite this paper: Moon, B., Choi, J., Lee, J. and Lee, M. (2021) Efficient Object Localization Scheme Based on Vanishing Line in Road Image for Autonomous Vehicles. *Journal of Computer and Communications*, 9, 85-97.
<https://doi.org/10.4236/jcc.2021.99006>

Received: September 2, 2021

Accepted: September 24, 2021

Published: September 27, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper proposes an efficient object localization method based on a vanishing line. The proposed method can be much improved in time efficiency since it requires scanning only vanishing line area. It requires the time complexity of $O(n)$ while the existing sliding window method requires the time complexity $O(n^2)$ for detecting all objects in the entire image. In addition, the range of detection area can be also remarkably reduced when compared with the sliding window method. As a result, the total range and times for searching in the proposed method can be significantly reduced by considering together the distance and position of the object. The experiment on the proposed method is performed with the virtual road data set known as SYNTHIA, and the competitive results are obtained.

Keywords

Object Localization, Image Preprocessing, Sliding Window, Vanishing Line, Autonomous Vehicle

1. Introduction

Recently the importance of vision-based semantic segmentation in urban scenarios has emerged due to the result of deep learning-based research on autonomous vehicle. In particular, it is important to accurately estimate the location of objects in road images before the objects are actually recognized. That is, the deep learning-based object recognition can be deployed after the localization process of estimating the location of objects in an image is completely achieved.

*Corresponding author.

In road images for autonomous vehicle, the first step in recognizing an object such as a person, a car, traffic light, or an extra obstacle is a localization method for determining existence of an object by using a bounding box. This process is a binary classification stage that predicts only whether an object exists or not in the specific location without recognizing the information about an object. Until now most previous works [1]-[10] on localization have been carried out based on identifying space boundaries by bounding boxes, which contain all regions of an image to recognize the extent of an object.

There are actually many different forms where the location of an object in the image is represented (e.g., by its center point, its contour, a bounding box, or by a pixel-wise segmentation). Object localization is often a very challenging issue because an object to be localized actually defines a category of objects with large variations. The object localization, called sometimes object detection, can be defined as finding the exact locations of objects in the image. Until now the object localization methods, which are based on creating a bounding box, have been focused for many years [1]-[18]. That is, these methods, called sliding window classifier, have been widely deployed and successful in finding objects. In these methods, an image is partitioned into a set of overlapping windows. Eventually object localization can be reduced to searching for a sub-window that covers as many positive features and as few negative features as possible.

However, traditional sliding window schemes [1] [2] [3] for object localization require a complex calculation in several image areas and take a large computation time in that they analyze the entire image. That is, these schemes make the computational cost greatly increased since they require the scrutiny of the extent of the entire frame. Furthermore, the entire image is scanned several times in order to adjust the range of localization. Therefore, these schemes have serious drawback of causing computing cost to increase greatly.

In this paper, we propose an efficient localization scheme called Vanishing Line based Object Localization (VLOL), which uses a vanishing line, a characteristic of urban road image. In this scheme, first, the linear components are extracted from the road image and then the vanishing point is located from these linear components. Second, the vanishing line is created by locating vanishing point [13] [18]-[25]. All objects are located along the vanishing line since they do not deviate from the region corresponding to the maximum height of the object. Third, the area where the vanishing line does not pass is excluded from searching process since it is considered as the outside of the object. That is, this scheme follows a linear search strategy in finding the locations of objects in road image since vanishing line is initially used to narrow the search range from the wide image area. Hence, the number of pixels included in searching process is reduced since computational resources can be concentrated on a specific area on the image.

It is possible to detect effectively objects from the outline of the image, moreover, since the different detection approach is deployed according to the close object or the long distance object. The proposed scheme does not require the

iterative scanning process, therefore, since the localization range can be adjusted precisely by scanning only a single image as considering the distance of the object. As a result, the total range and times for searching can be significantly reduced by considering together the distance and position of the object.

In this model, a depth map of the image is first created and then a vanishing point is located. Next, the expected range of the object is reduced to the area of the vanishing line. Finally the image can be scanned along the vanishing line to locate the presence of the object. In order to validate the effectiveness of the VLOL method, moreover, we have used the images with various road conditions from the SYNTHIA dataset [26]. The text given in the image also provides a detailed analysis of the components of the VLOL. Eventually we obtained the competitive results from the experiment on VLOL scheme using the image set of urban scenarios. These results can be deployed for efficient object localization and detection in road image for autonomous vehicle.

In Section 2, related works are investigated, and the proposed model is introduced in Section 3. In Section 4, the experiment results are presented and conclusion is given in Section 5.

2. Related Works

In most object-recognition systems, objects are typically recognized in two main stages: first, the position of an object in the image is estimated, and second, the object is recognized and classified. At this stage, an efficient and accurate guess makes it possible to identify objects well. The object recognition is the next step for determining what the extracted object is.

2.1. Object Localization Approach Based on Sliding Window

The sliding window approach [1] [2] [3] creates rectangular sections across the image and examines every possible sub-image (sub-window) in the image. This method first trains a quality function (e.g. a classifier score) based on the extracted features from the training images. Due to the unknown size and pose about the object, however, it must check all the sub-windows with different sizes, which is computationally very expensive. In other words, as an $n \times n$ image has $O(n^4)$ sub-images, it makes billions of windows evaluated, and hence an exhaustive search is not practical.

In the sliding window approach, the classifier function has to be evaluated over a large set of candidate sub-windows. Lamport *et al.* [4] [6] propose an efficient subwindow search (ESS) method to speed up object localization using heuristic approach. ESS approach is very fast since it relies on a branch-and-bound search instead of an exhaustive search. Consequently, ESS is guaranteed to find the globally maximal region, independent of the shape of quality function's landscape.

The computational complexity of branch and bound method used in ESS [4] [6] varies widely from $O(n^2)$ to $O(n^4)$ for $n \times n$ image since the search algorithm

yields its worst computational complexity $O(n^4)$ by examining most of the sub-images when the object is not in the image. An *et al.* [5] present two significantly faster methods based on the linear-time Kadane's Algorithm with $O(n^3)$ in the worst case, where main idea is a combination of the key ideas from ESS and Bentley's algorithm with $O(n^2)$.

Harzallah *et al.* [7] present a combined approach for object localization and classification. This approach has two contributions: first, a contextual combination of localization and classification shows that classification can improve detection and vice versa. Second, object localization method combines the efficiency of a linear classifier with the robustness of a sophisticated non-linear one under an efficient two-stage sliding window.

Yeh *et al.* [27] extended the ESS algorithms to search for polygonal sub-windows that may not be rectangular. This consequently increases the computational complexity in the optimal sub-window search, however, since the shape of the sub-window such as a specified number of sides must be pre-specified. Zhang *et al.* [8] develop a new graph-theoretic approach for object localization with a free-shape sub-window, which can take any shape of polygon without a specified number of sides.

For detecting multi-class multiple objects such as person, car and traffic light, Ibrahim *et al.* [9] propose an efficient multi-object detection approach based on image superpixelization. They utilize image superpixels in two ways: first, if pre-processing stage is done once for an image, multiple detections could be fast. Second, if image superpixels could be used for identifying the promising candidate sub-windows, evaluating all sub-windows could be avoided.

2.2. Vanishing Point Detection Approach

Tai *et al.* [13] proposed a method that takes a different perspective to detect vanishing points. Instead of accumulating intersection points, they compute the probability of a group of lines passing the same point. This approach provides a probability measure for discriminating between competing hypotheses irrespective of the size of the vanishing group. In addition, its performance also degrades gracefully in noisy environments.

Kong *et al.* [20] decomposed the road detection process into two steps: first, the estimation of the vanishing point is associated with the main straight part of the road, and second, the segmentation of the corresponding road area is based on the detected vanishing point. They proposed a novel approach called adaptive soft voting scheme, which is based on variable-sized voting region using confidence-weighted Gabor filters that compute the dominant texture orientation at each pixel. That is, a new vanishing point-constrained edge detection technique is used for detecting road boundaries.

Lezama *et al.* [22] presented a method for the automatic detection of vanishing point in urban scenes, which is based on finding point alignments in a dual space. In this method, when converging lines in the image are mapped to aligned

points, the PClines transformation based on parallel coordinates is used. Eventually, a post-processing step discriminates relevant from spurious vanishing point detections with two options: using a simple hypothesis of three orthogonal vanishing points (Manhattan-world) or the hypothesis that one vertical and multiple horizontal vanishing points exist.

3. Proposed Scheme

In this section, an effective object localization scheme is proposed, which uses the characteristics of road images deployed in autonomous vehicle. Here, the road image has the following two common features. First, it has the same camera calibration. That is, all objects are reduced to the same depth ratio by the perspective on the image. Thus, objects in the class of the same depth always appear at the same or similar size in the image. Second, it has a vanishing point. In other word, the road image that represents three-dimensional physical world in two dimension always has a vanishing point.

Hence, it is possible to effectively extract the objects with these features without using a sliding window method. The proposed localization scheme works in two stages. First, we need to find the vanishing point in the image and then create a vanishing line based on it. Next, the objects are detected by tracking along the generated vanishing line. In order to detect all objects in the entire image, the detection range should be smaller than that of the conventional sliding window method since it corresponds to the vanishing line. Therefore, the time required for the object detection becomes much less.

3.1. Vanishing Point and Vanishing Line

There exist a number of straight lines mutually parallel in three dimension (3D). Under perspective projection, these lines meet at a common point known as the vanishing point (VP). A vanishing point is the point where all the straight lines in the physical space meet. Regardless of what plane is actually in physical space, they all have the same vanishing point if they are parallel to each other in three-dimensional space [4] [16] [19] [21] [22] [23] [24] [28] [29]. The disappearance line is a group of vanishing points and a set of vanishing points of straight lines belonging to the same plane. **Figure 1** shows a vanishing point seen at the far end of the railroad and vanishing lines.

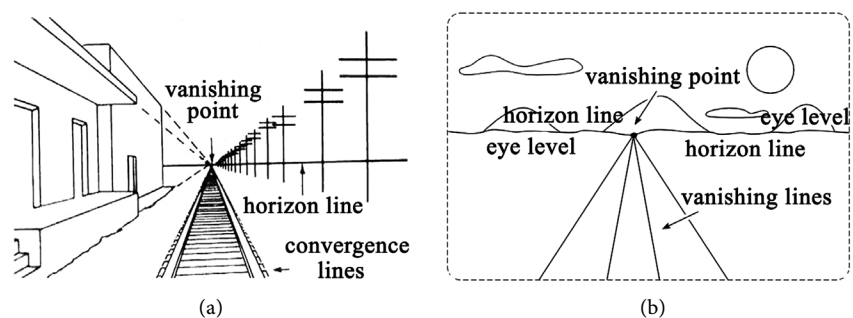


Figure 1. (a) A vanishing point seen at the far end of the railroad and (b) vanishing lines.

An obvious approach to locating VPs is to exploit directly the property that all lines with the same orientation in 3D converge to a VP under perspective transformation. Thus the task of VP detection can be treated as locating peaks in a two dimensional array where the intersections of all line pairs in an image plane accumulate [29]. Hence, the extraction method of VPs, which performs accumulation of line pair junctions, is actually most popular. This method causes problem on implementation, however, since the line pairs can intersect anywhere from points within an image to infinity.

3.2. Image Pre-Processing

3.2.1. Image Transformation to Depth Map

A depth map is an image that contains information about the distance from the viewpoint to the object's surface. The depth of an object in the real world can be predicted based on an image projected on each eye in a slightly different position. Likewise, the depth information of an object can be reconstructed based on an image taken by two cameras at the same time [30]. The depth can be generally expressed in black and white even though it is displayed in various ways. The nearest object is represented in black, and the farther the object is, the closer to white it is. **Figure 2** shows that the left image and the right image are taken by two cameras and the depth map reconstructed based on both road images. The proposed algorithm uses a depth map as an input image for localization.

3.2.2. Generating Vanishing Line by Finding Vanishing Point

Many straight line components can be detected in a plurality of planes since a large number of complex objects exist in a road scene image. Hence, there are no distinct VPs or multiple VPs may be detected. The texture of the road image or other unnecessary objects can make the scene complicated and thus cause many segments detected false. Typically, the lane of the road is the most valid straight line components in the road image. Hence, a high-resolution vanishing point can be found by detecting the linear component of the lane in the image.

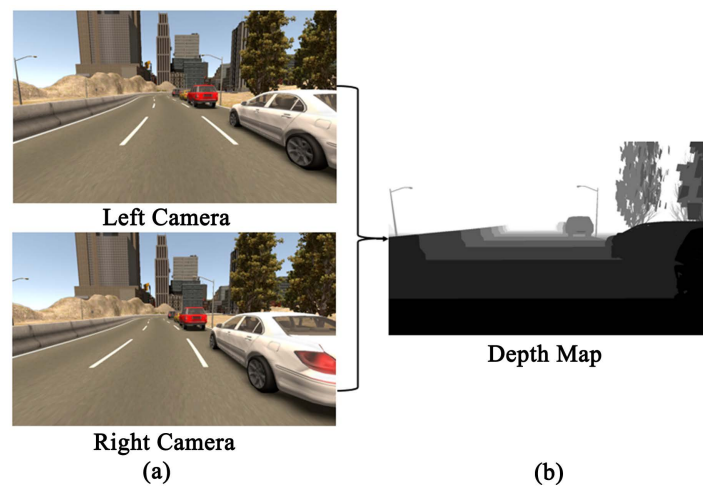


Figure 2. (a) A road image and (b) Depth map of the road image.

Figure 3 shows the process for generating vanishing line. In this figure, the effective straight line component can be extracted by using three steps of filtering process. Eventually the vanishing point can be found based on this component. The process for extracting valid straight line is given as follows. First, a new image has to be created by cropping the road part at the low area of the image. The aim of this step is to eliminate the other linear components that could cause an error while locating the vanishing point, and eventually use the linear component that appeared in the only road lane. Second, Gaussian Blur [31] is used to reduce noises since they may interfere with linear component. Third, Canny Edge [32] is used to find the contour of the road lane corresponding to the edge where the gray scale intensity changes. Fourth, Hough Lines [33] is used to detect the straight line component of the lane. Hence, the vanishing point becomes the intersection point where the extension line of the linear components in the image is most encountered. Eventually, the vanishing lines centering on the vanishing point are generated over the entire image.

3.3. Object Localization Based on Vanishing Line

3.3.1. Setting Bounding Box for Object of Various Sizes

Generally, objects of various sizes can be located over the whole image since the sizes of objects are different according to the type of objects. In conventional sliding window method hence, it is necessary to set the size of the reference bounding-box, and thus objects can be scanned with bounding boxes of various sizes. This method consumes a lot of time and resources, however, since it requires a plurality of scanning processes for the entire image [1]-[9].

In order to solve these drawbacks, it is necessary to promptly fix the size of reference bounding-box. In our proposed scheme, the proper size of bounding-box can be easily determined by scanning the image according to the depth

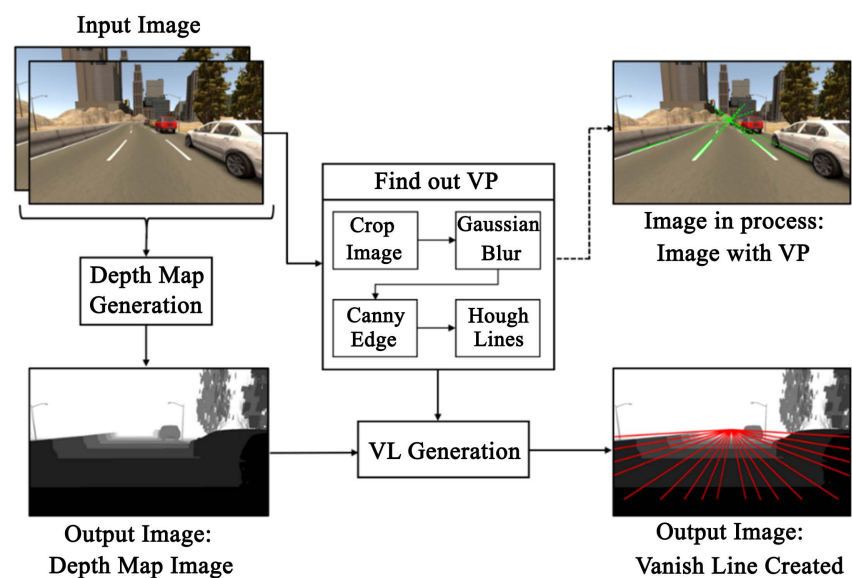


Figure 3. Image pre-processing for generating vanishing lines in VL0L.

type of object. That is, the candidate regions for object detection to be performed over the whole image are noticeably reduced. In addition, the objects of various sizes can be extracted since it is possible to obtain bounding-boxes of various sizes by scanning the whole image only once (Table 1).

The size of the object can be predicted by scanning along the vanishing line on the road image and thus the reference bounding-box can be created, therefore, since pedestrians and cars are important factors in road images. For example, a small bounding-box can be extracted efficiently if an object of human size is encountered, and a large bounding-box can be created if an object of vehicle size is encountered. The range of object sizes is quite various since it is actually set according to the length of the front and the side for an object. This range is also determined in inverse proportion to the depth value, which can be obtained from the experiment for the SYNTHIA dataset [26].

3.3.2. Localization Methods Depending on Distance

In vanishing line-based object localization method (VLLOL), the pixels of an image are first scanned along a vanishing line. Second, the position of the object is fixed by the point where the depth of the pixel in the image changes. The edge image is excluded from the detection range since an object near the edge image may be erroneous due to the characteristics of the road image. An object that is cut across the edge image is referred to as a nearby object. To solve these errors, two different algorithms are used to detect both nearby objects and far objects.

The basic detection methods in Algorithm 1 and Algorithm 2 are as follows. First, we try to find the x-axis length of the detected object (bounding box). Second, the x-axis length is compared with the range given in Table 1. Third, if it falls within the range, this determines that the object exists and then localization is performed.

<p>Input: <code>img[]</code> for depth map image, <code>VLlist[]</code> for coordinate points constituting vanish line Output: <code>Objlist[]</code> for pairs of the center coordinate points of detected object its tag</p>
<pre> 1: Construct VLlist of coordinate points constituting the vanishing line from the depth image. 2: for <code>i = 0</code> to <code>length(VLlist[])</code> do 3: if the current pixel in depth image is similar to the previous one then 4: the current pixel is stored as the starting point of the object 5: the previous pixel is set as the current pixel 6: else if the current pixel in depth image is different from the start point pixel then 7: the current pixel is stored as the end point of the object 8: if the end point of object is not zero then 9: get the x-axis width (end point – starting point) length of the object 10: end if 11: if x-axis width length of detected object falls within human size range then 12: put the center coordinate (x, y) of human object with 'human' tag into <code>Objlist</code> 13: else if x-axis width length of detected object falls within car size range then 14: put the center coordinate (x, y) of car object with 'car' tag into <code>Objlist</code> 15: end if 16: end if 17: end for 18: return <code>Objlist</code> </pre>

Algorithm 1. Algorithm for detecting far objects in the road image.


```

Input: imag[ ] for depth map image, VLlist[ ] for coordinate points constituting vanish line
Output: Objlist[ ] for pairs of the center coordinate points of detected object its tag

1: Construct the VLlist of coordinate points constituting the vanishing line from the depth image.
2: if the depth of edge object is 0, 1 or 2 then
3:   for i = 0 to length(VLlist[]) do
4:     if the depth is suddenly brightened at that point then
5:       put center coordinate (x, y) of nearby object with 'nearby' tag into Objlist
6:     end if
7:   end for
8: end if
9: return Objlist

```

Algorithm 2. Algorithm for detecting nearby objects in the road image.

Table 1. Detection range according to object depth on SYNTHIA dataset.

Depth	Detection range			
	0	1	2	3
Car (unit: # of pixels)	min = 56 max = N/A	min = 56 max = N/A	min = 41 max = N/A	min = 41 max = N/A
Pedestrian (unit: # of pixels)	min = 25 max = 55	min = 25 max = 55	min = 25 max = 40	min = 15 max = 40

The sliding window has been used for many years as an object localization method for locating an object in an image. This method utilizes an image pyramid to scan the objects in an image at various scales and different locations. As shown in **Table 2**, the search range becomes the entire image in sliding window and the time complexity becomes $O(n^2)$. In the proposed VLOL scheme, however, the search range is reduced to the pixels on the vanishing lines in an image, and so the time complexity becomes $O(n)$.

4. Experimental Results and Discussion

SYNTHIA dataset is a virtual image set which is designed for representing road traffic conditions [26]. In our experiment, object detection based on VLOL method is performed, where depth map and RGB image from SYNTHIA Dataset are used. A total of 297 images were used in the experiment, and the total number of 434 bounding boxes are generated from these images. A total of 280 bounding boxes of them were created exactly at the position where the object is actually located. Similarly, a total of 20 bounding boxes of them were not created properly at the position where the object is actually located, and also a total of 134 bounding boxes of them were created wrongly at the position where the object is not actually located (see **Table 3**).

A total of 300 of the 434 bounding boxes have actual objects, but the actual objects are detected correctly in only 280 of the 300 bounding boxes. Therefore, the object detection accuracy becomes 93.3%. In our experiment, the total run duration time for 279 images is 48.89 seconds, and hence the average run duration time per image becomes 0.16 seconds. Hence, the average run duration time

Table 2. Time complexity and candidate region on the proposed method.

	Sliding Window	VLOL
time complexity	$O(n^2)$	$O(n)$
candidate region	whole image pixels	pixels on vanishing line

Table 3. Experimental results using VLOL method.

Total 297 SYNTHIA Images (SYNTHIA-SEQS-05-SPRING/Stereo_Left/Omni_B)						
Total Boxes	True Positive	False Negative	False Positive	Accuracy	Run Duration	Average Run Duration per Image
434	280	20	134	93.3%	48.89	0.16

per image can be much reduced in general scenario since the proposed method requires the time complexity of $O(n)$ while the existing sliding window method requires the time complexity $O(n^2)$ for detecting all objects in the entire image.

Experimental results show that a large number of objects can be detected by using the proposed VLOL method. That is, the total number of undetected objects is only 20 when 297 images are deployed in actual experiment. When analyzing the reasons why these 20 objects were not detected, we can conclude that this mistake was not due to an error in VLOL method but an inaccuracy in the vanishing point creation process. In other word, if vanishing point is not created accurately for the image, the object is not included in the area of this vanishing line. If more precise vanishing point creation process is developed and hence all the objects are included in the area of the generated vanishing line, therefore, a very high rate of object detection can be obtained.

5. Conclusions

In this paper, we proposed an efficient localization method called Vanishing Line based Object Localization (VLOL), which uses a vanishing line, a characteristic of urban road image. In this method, the number of pixels included in searching process is significantly reduced since computational resources can be concentrated on a specific area on the image. It is possible to detect effectively objects from the outline of the image, moreover, since the different detection approach is deployed according to the close object or the long distance object. This method does not require the iterative scanning process, therefore, since the localization range can be adjusted precisely by scanning only a single image as considering the distance of the object. As a result, the total range and times for searching can be significantly reduced by considering together the distance and position of the object.

For the future work, the proposed **Algorithm 1** and **Algorithm 2** need to be improved by deploying more precise vanishing point creation process and vanishing line generation method. Final step for the experiment is to use real road data set instead of the virtual road data set known as SYNTHIA. Eventually this

method can be used for object localization on the real road image in autonomous vehicle systems.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Rowley, H.A., Baluja, S. and Kanade, T. (1996) Human Face Detection in Visual Scenes. *Neural Information Processing Systems*, **8**, 875-881.
- [2] Dalal, N. and Triggs, B. (2005) Histograms of Oriented Gradients for Human Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1**, 886-893.
- [3] Bosch, A., Zisserman, A. and Munoz, X. (2007) Representing Shape with a Spatial Pyramid Kernel. *ACM International Conference on Image and Video Retrieval*, Amsterdam, 9-11 July 2007, 401-408. <https://doi.org/10.1145/1282280.1282340>
- [4] Lampert, C., Blaschko, M. and Hofmann, T. (2008) Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, 24-26 June 2008, 1-8. <https://doi.org/10.1109/CVPR.2008.4587586>
- [5] An, S., Peursum, P., Liu, W. and Venkatesh, S. (2009) Efficient Algorithms for Subwindow Search in Object Detection and Localization. *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, 20-25 June 2009, 264-271.
- [6] Lampert, C.H., Blaschko, M.B. and Hofmann, T. (2009) Efficient Subwindow Search: A Branch and Bound Framework for Object Localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**, 2129-2142. <https://doi.org/10.1109/TPAMI.2009.144>
- [7] Harzallah, H., Jurie, F. and Schmid, C. (2009) Combining Efficient Object Localization and Image Classification. *IEEE International Conference on Computer Vision*, Kyoto, 27 September-4 October 2009, 237-244. <https://doi.org/10.1109/ICCV.2009.5459257>
- [8] Zhang, Z., Cao, Y., Salvi, D., Oliver, K., Waggoner, J. and Wang, S. (2010) Free-Shape Subwindow Search for Object Localization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, 13-18 June 2010, 1086-1093. <https://doi.org/10.1109/CVPR.2010.5540095>
- [9] Ibrahim, M.S., Badr, A.A., Abdallah, M.R. and Eissa, I.F. (2012) Bounding Box Object Localization Based On Image Superpixelization. *Procedia Computer Science*, **13**, 108-119. <https://doi.org/10.1016/j.procs.2012.09.119>
- [10] Park, K., Lee, J., Stanačević, M., Hong, S. and Cho, W.-D. (2008) Iterative Object Localization Algorithm Using Visual Images with a Reference Coordinate. *EURASIP Journal on Image and Video Processing*, **2008**, Article ID: 256896. <https://doi.org/10.1155/2008/256896>
- [11] Everingham, M., Van Gool, L., Williams, C.K., Winn, J. and Zisserman, A. (2010) The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, **88**, 303-338. <https://doi.org/10.1007/s11263-009-0275-4>
- [12] Dai, Q. and Hoiem, D. (2012) Learning to Localize Detected Objects. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2012)*, Providence, 16-21 June 2012, 3322-3329.

- [13] Tai, A., Kittler, J., Petrou, M. and Winder, T. (1992) Vanishing Point Detection. *Proceedings of the British Machine Vision Conference*, Leeds, 22-24 September 1992, 109-118. <https://doi.org/10.5244/C.6.12>
- [14] Hwang, S.J. and Grauman, K. (2012) Reading between the Lines: Object Localization Using Implicit Cues from Image Tags. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**, 1145-1158. <https://doi.org/10.1109/TPAMI.2011.190>
- [15] Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T. and Smeulders, A.W.M. (2013) Selective Search for Object Recognition. *International Journal of Computer Vision*, **104**, 154-171. <https://doi.org/10.1007/s11263-013-0620-5>
- [16] Hariyono, J., Hoang, V.-D. and Jo, K.-H. (2014) Moving Object Localization Using Optical Flow for Pedestrian Detection from a Moving Vehicle. *The Scientific World Journal*, **2014**, Article ID: 196415. <https://doi.org/10.1155/2014/196415>
- [17] Mnih, V., Heess, N., Graves, A. and Kavukcuoglu, K. (2014) Recurrent Models of Visual Attention. *The 27th International Conference on Neural Information Processing Systems*, Montreal, 8-13 December 2014, 2204-2212.
- [18] Tompson, J., Goroshin, R., Jain, A., LeCun, Y. and Bregler, C. (2015) Efficient Object Localization Using Convolutional Networks. *The IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 648-656. <https://doi.org/10.1109/CVPR.2015.7298664>
- [19] Collins, R.T. and Weiss, R. (1990) Vanishing Point Calculation as a Statistical Inference on the Unit Sphere. *Proceedings of ICCV3*, Osaka, 4-7 December 1990, 400-403.
- [20] Kong, H., Audibert, J.-Y. and Ponce, J. (2009) Vanishing Point Detection for Road Detection. *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, 20-25 June 2009, 96-103.
- [21] Ding, W. and Li, Y. (2015) Efficient Vanishing Point Detection Method in Complex Urban Road Environments. *IET Computer Vision*, **9**, 549-558. <https://doi.org/10.1049/iet-cvi.2014.0187>
- [22] Lezama, J., Randall, G. and von Gioi, R.G. (2017) Vanishing Point Detection in Urban Scenes Using Point Alignments. *Image Processing on Line*, **7**, 131-164. <https://doi.org/10.5201/ipol.2017.148>
- [23] Moon, Y., Geem, Z. and Han, G. (2018) Vanishing Point Detection for Self-Driving Car Using Harmony Search Algorithm. *Swarm and Evolutionary Computation*, **41**, 111-119. <https://doi.org/10.1016/j.swevo.2018.02.007>
- [24] Kim, J. (2019) Efficient Vanishing Point Detection for Driving Assistance Based on Visual Saliency Map and Image Segmentation from a Vehicle Black-Box Camera. *Symmetry*, **11**, 1492. <https://doi.org/10.3390/sym11121492>
- [25] Caprile, B. and Torre, V. (1990) Using Vanishing Points for Camera Calibration. *International Journal of Computer Vision*, **4**, 127-139. <https://doi.org/10.1007/BF00127813>
- [26] German, R., Laura, S., Materzynska, J., Vazquez, D. and Lopez, A.M. (2016) The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 3234-3243.
- [27] Yeh, T., Lee, J. and Darrell, T. (2009) Fast Concurrent Object Localization and Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, 20-25 June 2009, 280-287. <https://doi.org/10.1109/CVPR.2009.5206805>
- [28] Chen, X., Jia, R., Ren, H. and Zhang, Y. (2010) A New Vanishing Point Detection

-
- Algorithm Based on Hough Transform. *3rd International Joint Conference on Computational Science and Optimization*, Huangshan, 28-31 May 2010, 440-443. <https://doi.org/10.1109/CSO.2010.163>
- [29] Alexe, B., Deselaers, T. and Ferrari, V. (2012) Measuring the Objectness of Image Windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**, 2189-2202. <https://doi.org/10.1109/TPAMI.2012.28>
- [30] Kauff, P., Atzpadin, N., Fehn, C., Müller, M., Schreer, O., Smolic, A. and Tanger, R. (2007) Depth Map Creation and Image-Based Rendering for Advanced 3DTV Services Providing Interoperability and Scalability. *Signal Processing: Image Communication*, **22**, 217-234. <https://doi.org/10.1016/j.image.2006.11.013>
- [31] Getreuer, P. (2013) A Survey of Gaussian Convolution Algorithms. *Image Processing on Line*, **3**, 286-310. <https://doi.org/10.5201/ipol.2013.87>
- [32] Zhou, P., Ye, W. and Wang, Q. (2011) An Improved Canny Algorithm for Edge Detection. *Journal of Computational Information Systems*, **7**, 1516-1523.
- [33] Sneha, K. and Sheela, T. (2013) 2D to 3D Conversion for Images Using Hough Transform. *International Conference on Information Systems and Computing*, Milano, 15-18 December 2013, 482-486.