JOIV : Int. J. Inform. Visualization, 5(3) - September 2021 206-211

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Real-time Estimation of Road Surfaces using Fast Monocular Depth Estimation and Normal Vector Clustering

Chuho Yi[a], Jungwon Cho[b,*]

[a] Department of Software Convergence, Hanyang Women's University, Seoul 04763, South Korea
[b] Department of Computer Education, Jeju National University, Jeju 63243, South Korea
Corresponding author: *jwcho@jejunu.ac.kr

***Abstract*— Estimating a road surface or planes for applying AR (Augmented Reality) or an autonomous vehicle using a camera requires significant computation. Vision sensors have lower accuracy in distance measurement than other types of sensor, and have the difficulty that additional algorithms for estimating data must be included. However, using a camera has the advantage of being able to extract various information such as weather conditions, sign information, and road markings that are difficult to measure with other sensors. Various methods differing in sensor type and configuration have been applied. Many of the existing studies had generally researched by performing the depth estimation after the feature extraction. However, recent studies have suggested using deep learning to skip multiple processes and use a single DNN (Deep Neural Network). Also, a method using a limited single camera instead of a method using a plurality of sensors has been proposed. This paper presents a single-camera method that performs quickly and efficiently by employing a DNN to extract distance information using a single camera, and proposes a modified method for using a depth map to obtain real-time surface characteristics. First, a DNN is used to estimate the depth map, and then for quick operation, normal vector that can connect similar planes to depth is calculated, and a clustering method that can be connected is provided. An experiment is used to show the validity of our method, and to evaluate the calculation time.**

***Keywords*— Real-time estimation; deep neural network; road surfaces; fast monocular depth estimation; normal vector clustering.**

Manuscript received 8 Feb. 2021; revised 22 Mar. 2021; accepted 1 Apr. 2021. Date of publication 30 Sep. 2021.

I. INTRODUCTION

Many studies have used camera sensors in augmented reality, and for autonomous driving. Considerable information can be extracted from an image. However, in computer vision, various issues arise due to the large amount of data involved and poorly posed problems [1]. With the evolution of deep learning techniques and increases in computing power, extensive research has been conducted on camera sensors [2]. This paper determines the value of each pixel for estimating distance quickly, which is difficult with a single camera, using a deep neural network with a single state-of-the-art camera [3]. We modify the method used for extracting the plane to estimate the road surface in real time.

When using a camera, distance information is generally expressed in terms of depth or disparity. With an active sensor, a method that emits and receives a specific signal can be used to measure distances directly. It is possible to obtain red-green-blue plus depth data by adding an infrared sensor, such as a Kinect sensor, to the measurement system [4], [5]. Using a passive method, two cameras are calibrated; the difference in the distance to the same point between the left and right cameras can be used to obtain a short distance when the distance between the two cameras is sufficiently large. Another method is based on the binocular difference, which is a short distance [6], [7]. When using a single camera, the structure-from-motion method, which estimates the difference in three-dimensional distance while tracking multiple identical points that change over time, is also widely applied [8-10]. When using a single camera, pixel differences are typically not transformed into a distance immediately, but this can be done under certain conditions.

Zhan proposed an unsupervised learning method that models points in an image according to one real-world point as the camera moves over time [11], [12]. Godard et al. suggested a technique to estimate the depth map and ego-motion simultaneously using three loss functions to improve the monocular depth estimation. They also proposed a new matching loss function to handle hidden pixels, a simple auto-masking method for use when there is no movement relative to the camera, and a loss function to reduce depth artifacts

[13]. They proposed multi-scale appearance matching loss that runs at the input resolution. With that method, distances can be estimated based on a single camera image for autonomous vehicle applications. However, while these deep-learning methods can provide useful results, they require many parameters as the depth of the network increases, which increases the computation requirements [14]. Increasing the number of calculations in turn increases the processing time and power consumption. Poggi suggested a way to solve these problems by using a six-step method to simplify a single image, transferring the intermediate results of a small-scale network to a larger one with better performance [3]. Fig. 1 is an example of the network proposed by Poggi.

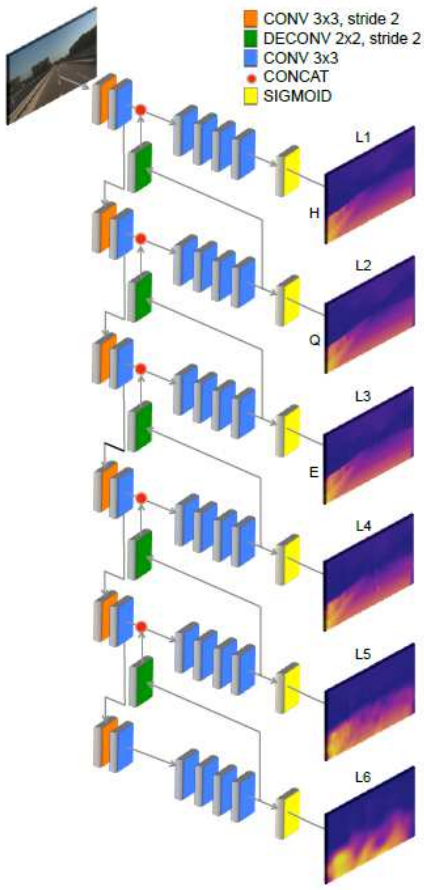

Fig. 1 Depth map estimation using a single camera in real time [3].

The first step when using the proposed network is to pass through the convolution layer in two-stride increments, which causes the image to shrink by a factor of two. The next step involves passing through five convolution layers, which results in a smaller image. This requires fewer parameters than most networks. For example, the Visual Geometry Group model [15] requires 31 million parameters, while the proposed network requires only 1,000,000, i.e., about 6.12% of the parameters. Consequently, the calculation time is about 20.3% shorter.

Here, we show how to estimate road surfaces using a depth map and our novel methods. Then, we present a modified method for estimating the road surface quickly. The position of each pixel in the image obtained using the depth map can be mapped to the world coordinate system; the mapped points form a point cloud. We also propose a modified clustering method. By estimating the existing plane, the road surface is known to be a certain distance from an extrinsic parameter (the center of the car), and an area having certain values in the point cloud is taken to be the road surface. To connect the planes, the normal vector is obtained. While these methods are somewhat basic, they require very little computation and are suitable for estimating road surfaces in real time. We confirm the usefulness of the proposed method experimentally and evaluate the calculation time.

II. Materials and Method

Fig. 2 is an image from the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) public dataset for autonomous vehicle research. This dataset comprises data obtained using multiple cameras, light detection and ranging (LIDAR), the Global Positioning System (GPS), and in-car sensors [16], [17]. It is widely used because it contains a large amount of manually labeled data. Here, we select some KITTI images to demonstrate the usefulness of the proposed method.

Fig. 2 KITTI dataset image.

Fig. 3 Results obtained using the fast monocular depth estimation method.

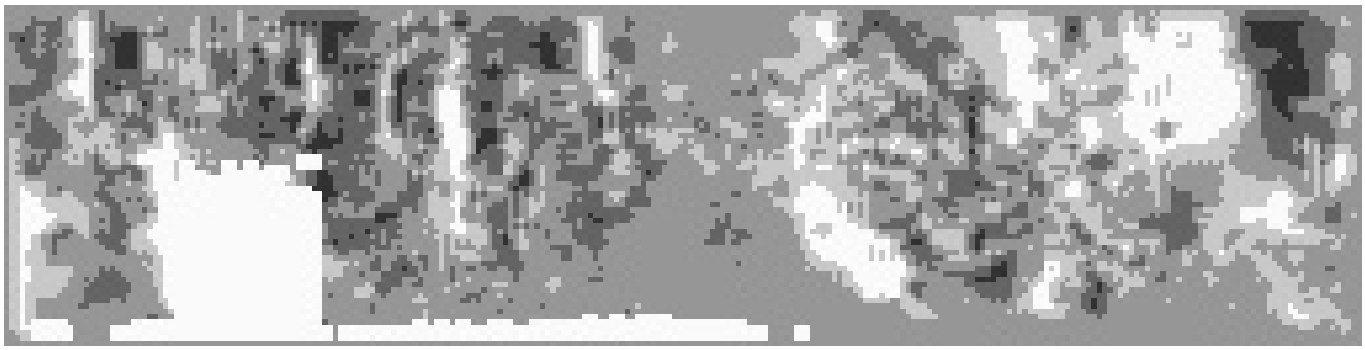

Fig. 4 Image of the normal vector value of the *x*-axis on the left–right axis in the camera world coordinate system.

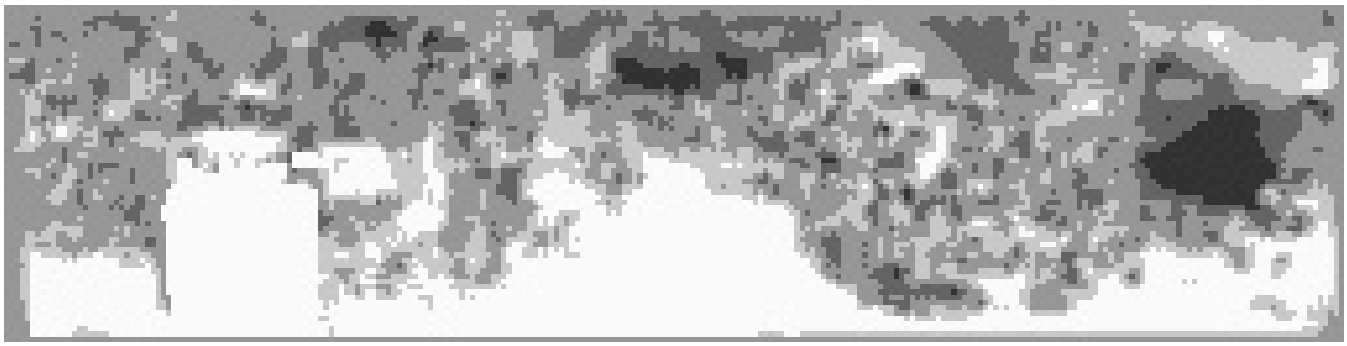

Fig. 5 Image of the normal vector value of the *y*-axis in the vertical axis in the camera world coordinate system.

Fig. 6 Image of the normal vector value of the *z*-axis on the front–rear axis in the camera world coordinate system.

Fig. 3 shows the results obtained on GitHub using PyD-Net code by Poggi, who proposed the fast monocular depth estimate method [18]. In the depth map, bright pixels indicate near distances, while dark areas are far from the camera. Compared to Fig. 2, the road surface and pillars on the left appear bright, while an empty area that can be assumed to be at infinity is black. In PyD-Net, the depth estimation and calculation time are closely linked to the image resolution and performance of the hardware used [3].

We propose a method for estimating the road surface in real time using depth estimates. We modified Holz’ s method to

enable real-time performance [19]. First, if the image in Fig. 3 is I, the value of each pixel can be expressed as $I_C = [I_u \quad I_v \quad I_d]^T$. Here, u means the left and right of the image and v means the top and bottom. To obtain the road surface, these values are changed to the world coordinate system via the camera's intrinsic parameter K. The associated expression is $I_W = [I_X \quad I_Y \quad I_Z]^T$ when using the real-world value of each image point. The value of K can be found in the KITTI dataset. The transformed equation is expressed as equation (1), where K^{-1} is the inverse matrix of the internal parameters of the camera [20], [21].

$$I_W = K^{-1} I_C \tag{1}$$

Then, with respect to the normal vector value thus obtained, and assuming that the plane or road surface we want to obtain has a large area, points are selected at regular intervals s. Here, we used $I'_W = [I'_X \quad I'_Y \quad I'_Z]^T$ and intervals of five pixels to significantly reduce the number of calculations. Then, to obtain the values in the image to the left and right of each image point, we determine three vectors: one on the x-axis, a second on the y-axis for the up and down values, and a third on the z-axis for in front of and behind values. In turn, these are used to obtain the unit normal vector $n_W = [n_X \quad n_Y \quad n_Z]^T$ [22]. Then, the values are quantized to cluster similar normal vector values. Assuming that the quantized value of a certain size is q, the normal vector can be expressed as:

$$n_W^q = \left[\frac{n_X}{q} \quad \frac{n_Y}{q} \quad \frac{n_Z}{q}\right]^T \tag{2}$$

After quantization, the value n_W^q of each normal vector is the same as in Fig. 4 to 6.

In Fig. 6, the road surface to be detected has a constant z-axis (–) value, obtained based on position information for the automobile provided by the camera. Using these values, the value of n_W^q is selected on a histogram, and a continuous area with a certain minimum size is determined to be the road surface. The formula for the histogram is as follows, where $\vec{n}$ represents the normal vector derived from equation (2).

$$H(\vec{n}) = \sum_{i=1}^{N} \vec{n} \tag{3}$$

The clustering results that enable these quick and simple operations are the same as in Fig. 7 and 8, and can be obtained in 20 ms or less.

III. Results and Discussion

Our experiment used the KITTI dataset, which is suitable because it contains experimental data obtained under multiple conditions, and some labeling data specifically related to automobiles. The computer used had an Intel i7-8700 CPU, 16 GB of memory, and a GTX 1070Ti graphics card. However, in the experiments and evaluations, the graphics processing unit (GPU) was not used for real-time operations, and will not be used in future embedded systems. The development environment was based on Python (ver. 3.7), and the library was implemented using TensorFlow (ver. 1.8).

The calculation involved two steps. Fast monocular depth estimation (step 1) took an average of 254.73 ms, while the road surface clustering process (step 2) took an average of 17.80 ms, for a total of 272.53 ms. Therefore, it was possible to operate at a speed of about 3.66 Hz.

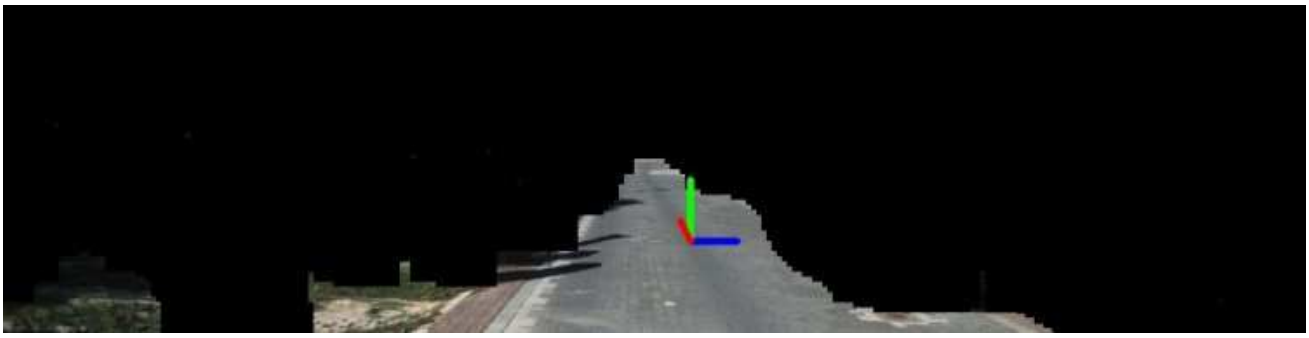

Fig. 7 Road surface estimated using the proposed method. The x- (blue), y- (green), and z-axes (red) are shown.

Fig. 8 The complete image, including the estimated road surface.

Fig. 9 Input image with strong edge components behind obstacles, such as buildings and trees along the road.

Fig. 10 The proposed method estimates the road surface by looking behind obstacles, such as trees along the road.

Fig. 11 Input image of a road in which speed and cross walk signs are strong edge components.

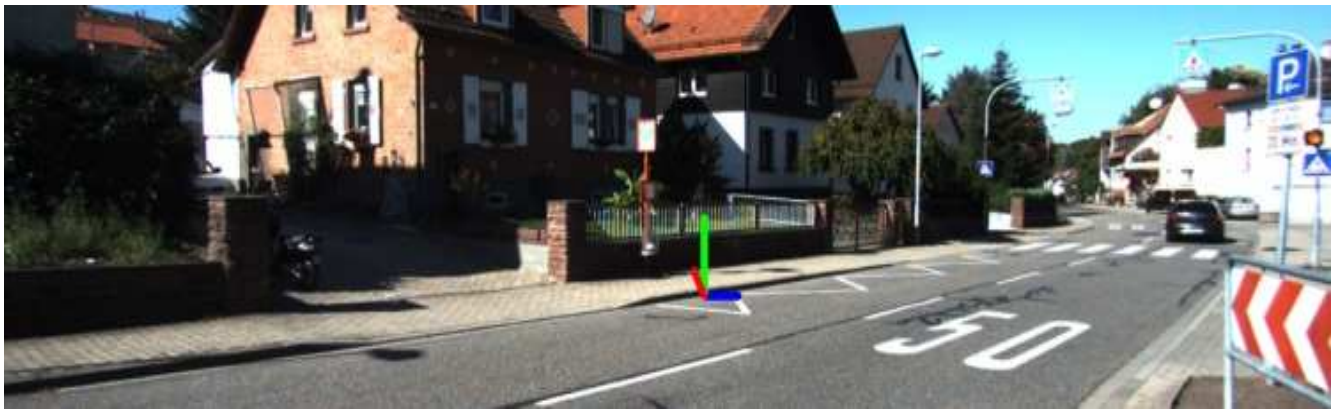

Fig. 12 The proposed method obtained a good result despite the road signs.

Fig. 9 to 12 show images of the two environments examined in the experiment.

Fig. 9 shows the results obtained when the road surface has many complex edge components, including the shadows of buildings and trees. While camera sensors encounter multiple problems, Fig. 10 shows that the proposed method can estimate the road surface without difficulty.

Fig. 12 shows that the proposed method could estimate the road surface even when there were various markings on the road surface, such as speed limits, cross walks, and lane demarcations.

IV. Conclusion

Our proposed method involves two steps. First, it uses a single camera and deep learning to produce a depth map, with PyD-Net code and a deep neural network with a pyramid form employed for rapid calculation. Since PyD-Net has significantly fewer parameters than a convolutional neural network, high-speed calculation is possible. Then, using the obtained depth map, a normal vector is obtained at each point in the image; these are then quantized and clustered for real-time operation, which is much faster. Our method requires very few calculations regardless of the environmental conditions and changes in the image. The method enables real-time estimation with no drop in performance. We plan to conduct additional experiments to identify other applications using a single camera, such as augmented reality applications.

Acknowledgment

This work was supported by the research grant of Jeju National University in 2021

References

[1] B. Jähne and H. Horst, *Computer vision and applications*, pp. 111-151, 2000.

[2] J. Brownlee, *Deep learning for computer vision: image classification, object detection, and face recognition in python*, 2019.

[3] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, Towards real-time unsupervised monocular depth estimation on cpu, *In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5848-5854, Oct. 2018.

[4] K. Y. Lin and H. M. Hang, Depth map enhancement on rgb-d video captured by kinect v2, *In 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1530-1535, Nov. 2018.

[5] Y. W. Kuan, N. O. Ee, and L. S. Wei, Comparative study of intel R200, Kinect v2, and primesense RGB-D sensors performance outdoors, *IEEE Sensors Journal, 19(19)*, pp. 8741-8750, 2019.

[6] D. Pohl, S. Dorodnicov, and M. Achtelik, Depth map improvements for stereo-based depth cameras on drones, *In 2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 341-348, Sep. 2019.

[7] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou, Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios, *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 899-908, 2019.

[8] O. Ozyesil, V. Voroninski, R. Basri, and A. Singer, A survey of structure from motion, *arXiv preprint arXiv:1701.08493*, 2017.

[9] M. R. U. Saputra, A. Markham, and N. Trigoni, Visual SLAM and structure from motion in dynamic environments: A survey, *ACM Computing Surveys (CSUR), 51(2)*, pp. 1-36, 2018.

[10] A. Shalaby, M. Elmogy, and A. A. El-Fetouh, Algorithms and applications of structure from motion (SFM): *A survey, Algorithms, 6(06)*, 2017.

[11] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid, Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction, *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 340-349, 2018.

[12] J. W. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M. M. Cheng, and I. Reid, Unsupervised scale-consistent depth and ego-motion learning from monocular video, *arXiv preprint arXiv:1908.10553*, 2019.

[13] C. Godard, O. M. Aodha, M. Firman, and G. J. Brostow, Digging into self-supervised monocular depth estimation, *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3828-3838, 2019.

[14] A. Ardakani, C. Condo, M. Ahmadi, and W. J. Gross, An architecture to accelerate convolution in deep neural networks, *IEEE Transactions on Circuits and Systems I: Regular Papers, 65(4)*, 1349-1362, 2017.

[15] S. Karen and A. Zisserman, Deep Convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*, 2014.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, Vision meets robotics: The kitti dataset, *The International Journal of Robotics Research, 32(11)*, pp. 1231-1237, 2013.

[17] A. Geiger, P. Lenz, and R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, *In 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354-3361, June 2012.

[18] M. Poggi, PyDnet, 2018. [online]. Available: https://github.com/mattpoggi/pydnet

[19] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, Real-time plane segmentation using RGB-D cameras, *In Robot Soccer World Cup*, pp. 306-317, July 2011.

[20] J. Oh, K. S. Kim, M. Park, and S. Kim, A comparative study on camera-radar calibration methods, *In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1057-1062, Nov. 2018.

[21] Z. Zhang, R. Zhao, E. Liu, K. Yan, and Y. Ma, A single-image linear calibration method for camera, *Measurement, 130*, pp. 298-305, 2018.

[22] O. Bouafif, B. Khomutenko, and M. Daoudi, Monocular 3D Head Reconstruction via Prediction and Integration of Normal Vector Field, *In 15th International Conference on Computer Vision, Theory and Applications*, Feb. 2020.