Evaluation of Dense Differential Filter to Detect Semantic Edges for Estimating 3D Room Structure

Marin Wada[†], Kae Nakayama[†], Junya Morioka[†], and Ryusuke Miyamoto[‡]

Abstract— **The authors attempt to actualize 3D reconstruction from a single view for previewing a room in virtual space using results of semantic segmentation. The segmentation accuracy has been drastically improved by state-of-the-art method, which enables pixel-wise classification of walls, floors, ceilings, and objects in the target room with sufficient accuracy. Assuming a room can be represented as a cuboid, its parameters can be computed analytically when semantic edges of the room structure are accurately obtained. In the actual process of the estimation, lines constructing the cuboid are estimated from detected edges. These edges are derived by spatial filtering to a semantic map corresponding to an input image. To enhance the effectiveness of edge detection on a semantic map, we adopted a simple differential filter that incorporates only two active values as filter coefficients, utilizing the smallest filter size. Experimental results using synthetic datasets showed no significant difference in the accuracy of line parameter estimation when comparing our method with a typical filter, despite using half the samples during the optimization process though sample numbers for parameter estimation became smaller obviously.**

I. Introduction

Estimation of three-dimensional (3D) room structure is crucial for various virtual reality (VR) applications, such as virtual previews and virtual coworking. For Virtual previewing, obtaining the absolute sizes of a target room is essential for planning furniture layouts in the 3D virtual space. The most popular option to estimate 3D room structure is measurement using a 3D scanner such as FARO [1], which can capture 3D point clouds that include color and depth information. However, in addition to the high cost of the 3D scanner, this method requires a person experienced in the process to obtain accurate measurements. It has recently become possible to scan a 3D structure using a LiDAR implemented on iPhones, but the obtained density and accuracy of the point clouds are insufficient. To reduce the device cost and simplify 3D scanning of a room without requiring specialized skills, we attempt to estimate a 3D room structure using only an input image.

For this purpose, monocular depth estimation emerges as a promising option, as it can estimate the depth from a single image with deep learning. Recently, the accuracy of monocular depth estimation has significantly improved with large datasets [2], [3]. Some researchers have adopted monocular depth estimation to estimate the relative depth for 3D reconstruction [4], [5]. However, classifiers for monocular depth estimation struggle to accurately estimate planar surfaces, despite the fact that accurate plane estimation is essential for 3D reconstruction of rooms composed of planes such as ceilings, floors, and walls.

Considering the current trend in image-based 3D reconstruction, we decided to pursue an alternative approach: estimating 3D room structure based on the results of two-dimensional (2D) semantic segmentation of an input image [6]. In this method, the 3D structure of a room is estimated using semantic edges as a visual cue to perform parameter optimization based on the least squares method, where a cuboid is selected as a model for target room. While the assumption of a cuboid may appear too inflexible for general use, it is feasible for real estate in Japan because most Japanese rental properties are composed of cuboids. Previous research [6] showed sufficient accuracy for 3D room structure estimation. To further improve this estimation accuracy, a novel method for independent extraction of semantic edges was proposed [7].

In this research, we focused on the edge filter to extract semantic edges from a segmentation result. Our previous research adopted the Canny method [8] for edge detection because it is widely used in the field of image processing. The Canny method is well-designed for edge filtering in natural images in which many types of noise are present. In contrast to natural images, segmentation images possess only a few distinct pixel values, each representing a class label. This characteristic makes a simple filter, such as the dense differential filter (DDF), particularly well suitable for this application. To investigate the performance of DDF in semantic edge detection, we applied it to the independent extraction of semantic edges from an input image for 3D room structure estimation. For this task, SegFormer [9] with an InternImage-based backbone [10] was adopted. For quantitative evaluation of edge extraction, the estimated line parameters were compared with those obtained using a sparse differential filter (SDF), leveraging known ground truth lines corresponding to semantic edges.

The rest of this paper is organized as follows. Section II introduces related work, primarily focusing on semantic segmentation and 3D room structure estimation based on it. Section III explains the process of applying DDFs to semantic edge extraction and line parameter estimation. Section IV evaluates the performance of DDF-based semantic edge extraction, and Section V concludes this paper.

*This work was supported by JSPS KAKENHI Grant Number JP25K01236

[†]Junya Morioka with Department of Computer Science, Graduate School of Science and Technology, Meiji University, Kawasaki, Japan mjun@cs.meiji.ac.jp

[‡]Ryusuke Miyamoto is with Department of Computer Science, School of Science and Technology, Meiji University, Kawasaki, Japan miya@cs.meiji.ac.jp

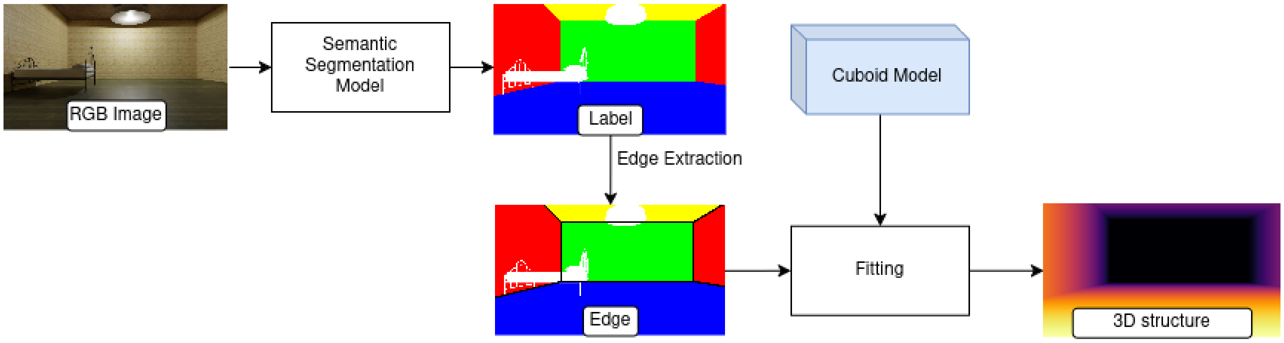


Fig. 1. Overview of our method.

-1	0	1
-1	0	1
-1	0	1

(a) Prewiit filter.

-1	0	1
-2	0	2
-1	0	1

(b) Sobel filter.

0	0	0
-1/2	0	1/2
0	0	0

(c) Differential filter.

Fig. 2. Typical edge extraction filters.

0	0	0
-1	1	0
0	0	0

(a) DDF.

0	0	0
-1	0	1
0	0	0

(b) SDF.

Fig. 3. Differential filter types.

II. Semantic Segmentation

Semantic segmentation is a task to estimate a class for each pixel in an image. By recognizing objects in a scene, semantic information can be obtained such as relationships between objects and their shapes: a cyclist is a person on a bicycle, or the shape of an orange is a sphere. The performance of semantic segmentation is generally evaluated with Intersection over Union (IoU) as the following equation:

$$IoU = \frac{TP}{TP + FP + FN}, \qquad (1)$$

where TP represents the number of pixels where the predicted region and the correct region match, FP represents the number of pixels that exist only in the predicted region, and FN represents the number of pixels that exist only in the correct region. A higher IoU indicates a greater overlap between the predicted and ground truth regions, meaning higher accuracy in semantic segmentation.

Before the emergence of deep learning (DL), Markov Random Fields (MRFs) [11], [12] and Conditional Random Fields (CRFs) [13], [14] were widely used for semantic segmentation. Clustering algorithms such as k-means [15] and mean-shift [16] were also employed. Bilateral filters [17], primarily used for smoothing, found utility in various applications. Notably, the Co-occurrence filter [18], based on the bilateral filter. demonstrated impressive accuracy in specific tasks without relying on learning, even in 2017, when DL-based semantic segmentation approaches had already achieved substantial advancements.

The advent of Convolutional Neural Networks (CNNs) has led to remarkable improvements in the accuracy of image classification tasks, such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [19], as well as object detection tasks. Based on the Integral Channel Features (ICF) [20], Informed filters [21] and Filtered Channel Features [22] were developed and demonstrated effectiveness in some object detection tasks [23]. However, the improved accuracy of methods using DL rendered these methods obsolete.

At the advent of DL, there were no methods designed for semantic segmentation because of the difficulty of this task compared to classification and detection. Fully Convolutional Networks (FCN) improved accuracy but could not sufficiently account for the structure of the entire image and the relationship.

In 2017, Pyramid Scene Parsing Network (PSPNet) [24] improved accuracy by introducing Spatial Pyramid Pooling [25], which considers global interrelationships in images. Following the remarkable success of Transformers [26] in natural language processing, they were introduced into image processing, leading to models such as the Vision Transformer (ViT) [27]. SegFormer [9], based on Transformer, was proposed for semantic segmentation. Its performance was further demonstrated on synthetic datasets generated from 3D point clouds for autonomous moving [28], [29]. Our aim is to reconstruct 3D information from an image with semantic segmentation [6] as shown in Fig. 1.

III. How to Estimate Parameters of a Line Composing a Cuboid

A. *Filtering for edge extraction from a semantic map*

In natural images, edges are typically extracted by spatial filters such as Prewitt, differential, and Sobel filters having 3 × 3 coefficients. Except for the differential filter, most filters attempt to suppress influence of noises by adding the averaging operation in filter coefficients. The most significant feature of these filters is rows or columns with zero as shown in Fig. 2. Though such filters work well for edge extraction on natural images, it may not appropriate when an input image is a semantic map composed of binary values because difference between just neighboring pixels should be measured. To extract semantic edges from just neighboring pixels, we decide to adopt a filter as shown in Fig. 3 (a),

-1	0	1
-1	0	1
-1	0	1

(a) Filter.

0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	1	1	1

(b) Input.

0	0	1	1	1	1
0	1	1	1	1	0
1	1	1	1	0	0

(c) Output.

Fig. 4. Edge extraction with Prewitt filter.

0	0	0
-1	0	1
0	0	0

(a) Filter.

0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	1	1	1

(b) Input.

0	0	0	1	1	0
0	0	1	1	0	0
0	1	1	0	0	0

(c) Output.

Fig. 5. Edge extraction with sparse differential filter.

0	0	0
-1	1	0
0	0	0

(a) Filter.

0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	1	1	1

(b) Input.

0	0	0	0	1	0
0	0	0	1	0	0
0	0	1	0	0	0

(c) Output.

Fig. 6. Edge extraction with dense differential filter.

where active values are densely arranged. This filter will be referred to as the dense differential filter (DDF) in the rest of this paper. In addition, the sparse differential filter (SDF) as shown in Fig. 3 (b) whose active coefficients are twice the size of the typical differential filter is defined. Edges extracted by these spatial filters are used to estimate line parameters, which enables 3D structure estimation of a cuboid representing a target room.

B. Parameter estimation of a line

Straight lines constructing the cuboid are estimated with Least Squares Method (LSM), using pixel coordinates in the semantic edge. The line is represented as the following equation:

$$x\cos(\theta) + y\sin(\theta) + c = 0, \tag{2}$$

where $(\cos(\theta), \sin(\theta))$ is a normal vector of the line and c is a constant. When Eq. (2) is denoted as the model function $S(\theta)$, the non-linear LSM should be solved because $\sin(\theta)$ and $\cos(\theta)$ are non-linear. Our problem is represented as the following equation:

$$\min_{\theta,c} S(\theta, c) = \min_{\theta,c} \sum_{i=1}^{N} (x_i \cos(\theta) + y_i \sin(\theta) + c)^2, \tag{3}$$

where (x_i, y_i) is a coordinate of the i-th pixel in N extracted pixels.

The Levenberg-Marquardt method, which exhibits characteristics intermediate between gradient descent and Gauss-Newton methods, is adopted to find the optimal solutions of θ and c when $S(\theta, c)$ is minimum. (θ, c) is updated as follows:

$$(\theta, c)_{k+1} = (\theta, c)_k + \boldsymbol{\delta}. \tag{4}$$

$\boldsymbol{\delta}$ is calculated by the following equation:

$$(J^T J + \lambda I)\boldsymbol{\delta} \quad = \quad J^T \mathbf{r}, \tag{5}$$

where

$$J_{ij} \quad = \quad \frac{\partial r_i}{\partial \theta_j}, \tag{6}$$

$$\mathbf{r}(\theta, c) \quad = \quad \begin{pmatrix} r_1(\theta, c) \\ \vdots \\ r_n(\theta, c) \end{pmatrix} = \begin{pmatrix} z_1 - f(x_1; \theta, c) \\ \vdots \\ z_n - f(x_n; \theta, c) \end{pmatrix}, \tag{7}$$

where $f(x_i; \theta, c)$ is a predicted value. J is a Jacobian matrix and λ is a damping coefficient.

C. Refinement with parallel movement

When typical filters for edge detection are applied to the semantic map, the width of the extracted edge is at least two pixels, as shown in Figs. 4 and 5. On the other hand, the width is one pixel if the DDF is applied, as shown in Fig. 6 All pixels detected as the semantic edge are employed as sample points for parameter estimation of the straight line. When the width of the semantic edge is two pixels, as shown in Fig. 7 (b), the estimated line is at the center of the two-pixel semantic edge. This is matched to the side of the cuboid, shown as a blue line in Fig. 7 (a).

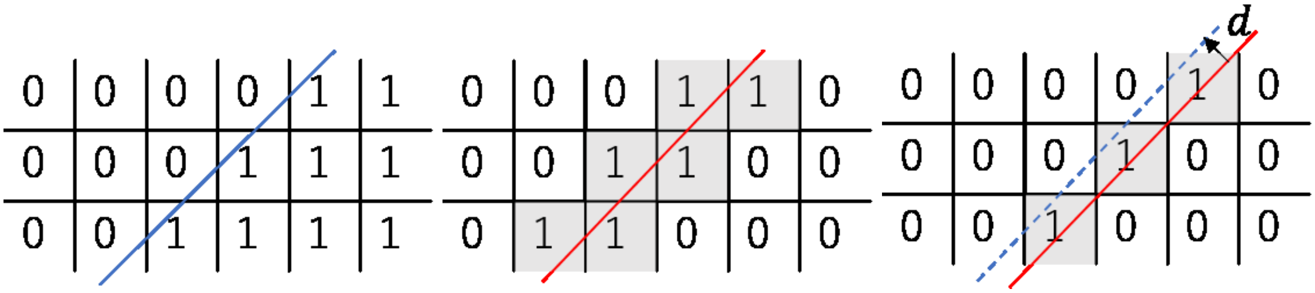
In contrast, the equation estimated from the semantic edge extracted with SDF is not the same as the side of the cuboid as shown in Fig 7 (b), the estimated line lies to the right of the line constructing the cuboid. This shift can be represented as signed distance d:

$$d \quad = \quad \mathrm{sign}(\cos(\phi)\sin(\phi)) \cdot \frac{\sin(\theta)}{2}, \tag{8}$$

$$\phi \quad = \quad \arctan 2(\cos(\theta), \sin(\theta)), \tag{9}$$

where ϕ is an angle between the x-axis and the straight line when the normal vector of the line is $(\cos(\theta), \sin(\theta))$. Therefore, the line constructing the cuboid is formulated as the following equation:

$$x\cos(\theta) + y\sin(\theta) + c + d = 0 \tag{10}$$



(a) Input. (b) Sparse differential filter. (c) Dense differential filter.

Fig. 7. Refinement of estimated line.

TABLE I
COLOR OF THE CLASS ON SEMANTIC SEGMENTATION.

Class	Color
ceiling	🟥
floor	🟦
left wall	🟨
right wall	▨
front wall	🟩
object	

IV. EVALUATION

SDF was evaluated for estimating edges of a cuboid from a single view with synthetic images. The synthetic images were created from a 3D model that replicates a cuboid room with randomly arranged furniture and a single ceiling light. The 3D model was assigned class labels for semantic segmentation: floor, ceiling, front wall, left wall, right wall, and object, as shown in Table. I. To introduce variations, four kinds of datasets were generated by varying the maximum camera roll angle during rendering: $\pm 0°, \pm 15°, \pm 30°$, and $\pm 45°$.

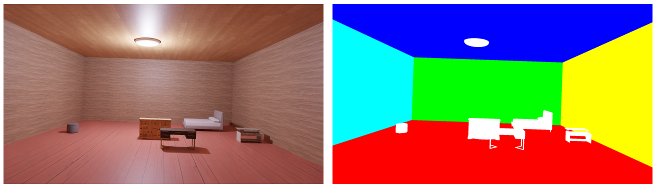
Figure. 8 is example images of datasets. Eight semantic edges are extracted as shown in Fig. 9. Line 0 (blue) is between the ceiling and the left wall, line 1 (yellow green) is between the ceiling and the front wall, line 2 (red) is between the ceiling and the right wall, line 3 (sky blue) is between the left and the front wall, line 4 (pink) is between the right and the front wall, line 5 (yellow) is between the left wall and the floor, line 6 (green) is between the front wall and the floor, and line 7 (purple) is between the right wall and the floor.

To evaluate the effectiveness in practical use, each inferenced semantic map of datasets was also prepared. SegFormer with InternImage [10] backbone was adopted for semantic segmentation. The performance of classifiers is shown in Table. II. SDF was adopted to compare the performance with DDF. These filters were evaluated with differences of θ and c:

$$\Delta\theta = |\theta_{gt} - \theta_{pred}| \tag{11}$$

$$\Delta c = |c_{gt} - c_{pred}| \tag{12}$$

, where ground truth edge is $xcos(\theta_{gt}) + ysin(\theta_{gt}) + c_{gt} = 0$ and predicted edge is $xcos(\theta_{pred}) + ysin(\theta_{pred}) + c_{pred} = 0$.



(a) Input image. (b) Semantic map.

Fig. 8. Example dataset images.

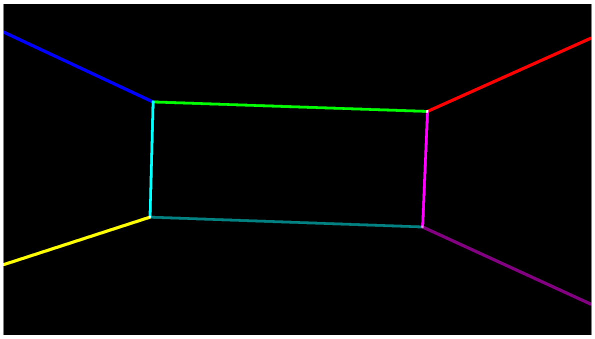


Fig. 9. Semantic edges.

The averages of $\Delta\theta$ and Δc are shown in Tables. III-X. For filtering the ground truth of semantic map, the DDF exhibited an angle error $\Delta\theta$ that was at least -0.03 and at most $+0.01$ larger than that of SDF. In addition, the difference in position error Δc between SDF and DDF ranged between -0.2 and $+0.8$. When the inferenced semantic maps were filtered, the $\Delta\theta$ of DDF was less than SDF in all cases, while the Δc of DDF was greater than SDF in the $\pm 0°$ and $15°$ datasets. Therefore, there was little difference between the performance of DDF and SDF, although DDF appeared to be slightly more robust for angle error than SDF.

The edge extracted with DDF had a width of one pixel, whereas the edge obtained with SDF was two pixels wide. As a result, DDF provided half the sample pixels for line parameter estimation compared to SDF. This indicates that DDF allows for more efficient straight-line estimation than SDF. Moreover, the dense arrangement of 1/-1 values in DDF allows less required memory capacity and faster memory access. Accordingly, DDF is determined to be more suitable for edge detection from the semantic map.

TABLE II

IoUs and mIoU with SegFormer.

dataset	floor	front	left	object	right	ceiling	mIoU
±0°	0.9948	0.9941	0.9981	0.9736	0.9981	0.9975	0.9927
±15°	0.9951	0.9941	0.9979	0.9743	0.9978	0.9981	0.9928
±30°	0.9956	0.9935	0.9978	0.9739	0.9977	0.9983	0.9928
±45°	0.9955	0.9937	0.9977	0.9722	0.9977	0.9983	0.9924

V. Conclusion

Considering the trend of image-based 3D reconstruction, our aim is to accurately estimate 3D room structure from an input image. In our approach, 2D semantic segmentation is applied to obtain components of a room in the 2D image plane; pixel-wise class labels are then obtained as a semantic map. To avoid unnecessary operations in the estimation procedure, this paper investigated the utility of dense differential filter (DDF), which employs only two active values within its spatial filter.

To evaluate the effectiveness of DDF, experiments were conducted with the inferenced semantic maps and ground truth of semantic maps. SDF was also employed for a performance comparison. Consequently, the performance of line parameter estimation showed no substantial difference between SDF and DDF. When DDF was applied, the number of sample pixels for straight line estimation was half that obtained with SDF. Therefore, DDF reduced the computation complexity of parameter estimation, indicating that DDF is more effective than SDF for practical use. In future work, we will verify the effectiveness of DDF using real-world images and realize 3D structure reconstruction for real-world rooms.

References

[1] "FARO — 3D measurement, imaging and realization technology," https://www.faro.com, [Accessed September 21, 2023].

[2] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth Anything: Unleashing the power of large-scale unlabeled data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024.

[3] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth Anything V2," in *Proc. Advances in Neural Information Processing Systems*, 2024.

[4] A. Bochkovskiy, A. Delaunoy, H. Germain, M. Santos, Y. Zhou, S. Richter, and V. Koltun, "Depth Pro: Sharp monocular metric depth in less than a second," in *Proc. Int. Conf. Learn. Represent.*, 2025.

[5] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen, "Metric3D v2: A Versatile Monocular Geometric Foundation Model for Zero-Shot Metric Depth and Surface Normal Estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 10 579–10 596, 2024.

[6] J. Morioka and R. Miyamoto, "3D Structure Estimation of Room Environment Using Semantic Segmentation and Model Fitting," in *IEEE International Workshop on Metrology for Living Environment*, 2024, pp. 448–453.

[7] R. Miyamoto, K. Nakayama, J. Morioka, and M. Adachi, "Semantic edge extraction by logical operations for room structure estimation using a segmentation result," in *Proc. XR Salento*, 2025 (accepted).

[8] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[9] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Advances in Neural Information Processing Systems*, 2021.

[10] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, X. Wang, and Y. Qiao, "Internimage: Exploring large-scale vision foundation models with deformable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2023, pp. 14 408–14 419.

[11] Z. Kato and T.-C. Pong, "A markov random field image segmentation model for color textured images," *Image and Vision Computing*, vol. 24, no. 10, pp. 1103–1114, 2006.

[12] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 1, pp. 25–39, 1983.

[13] X. He, R. Zemel, and M. Carreira-Perpinan, "Multiscale conditional random fields for image labeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 27 Jun.–2 Jul. 2004, pp. II–695–II–702.

[14] A. McCallum, "Efficiently inducing features of conditional random fields," in *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 403–410.

[15] T.-W. Chen, Y.-L. Chen, and S.-Y. Chien, "Fast image segmentation based on k-means clustering with histograms in hsv color space," in *2008 IEEE 10th Workshop on Multimedia Signal Processing*. IEEE, 2008, pp. 322–325.

[16] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, 1995.

[17] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 1998, pp. 839–846.

[18] R. J. Jevnisek and S. Avidan, "Co-occurrence filter," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3816–3824.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[20] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proc. Brit. Mach. Vis. Conf.*, 2009, p. 10.

[21] S. Zhang, C. Bauckhage, and A. B. Cremers, "Informed haar-like features improve pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 947–954.

[22] S. Zhang, R. Benenson, B. Schiele *et al.*, "Filtered channel features for pedestrian detection." in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1, no. 2, 2015, p. 4.

[23] T. Oki, R. Miyamoto, H. Yomo, and S. Hara, "Detection accuracy of soccer players in aerial images captured from several viewpoints," *Journal of Functional Morphology and Kinesiology*, vol. 4, no. 1, p. 9, 2019.

[24] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition." in *Proc. European Conf. on Computer Vision*, ser. Lecture Notes in Computer Science, D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8691. Springer, 2014, pp. 346–361.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 6000–6010.

[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 45–67.

[28] M. Wada, H. Sudo, M. Adachi, Y. Ueda, and R. Miyamoto, "Does a dense point cloud for training data generation improve segmentation accuracy?" in *IEEE Int. Conf. Big Data*, 2023, pp. 4356–4361.

TABLE III

±0° Dataset.

(a) DDF.

Line	$\Delta\theta$	Δc
line 0	0.010	0.264
line 1	0.000	1.029
line 2	0.017	1.623
line 3	0.000	0.503
line 4	0.000	0.489
line 5	0.102	1.608
line 6	0.000	0.981
line 7	0.069	2.577
all	0.025	1.134

(b) SDF.

Line	$\Delta\theta$	Δc
line 0	0.008	0.039
line 1	0.000	0.241
line 2	0.017	0.489
line 3	0.000	0.256
line 4	0.001	0.232
line 5	0.192	0.732
line 6	0.006	0.351
line 7	0.148	5.127
all	0.047	0.933

TABLE IV

±15° Dataset.

(a) DDF.

Line	$\Delta\theta$	Δc
line 0	0.022	0.277
line 1	0.001	0.641
line 2	0.016	1.531
line 3	0.004	0.578
line 4	0.004	0.580
line 5	0.304	4.983
line 6	0.011	0.755
line 7	0.056	4.198
all	0.052	1.693

(b) SDF.

Line	$\Delta\theta$	Δc
line 0	0.033	2.174
line 1	0.001	0.020
line 2	0.025	3.108
line 3	0.005	0.049
line 4	0.005	0.054
line 5	0.122	0.420
line 6	0.019	3.522
line 7	0.130	5.791
all	0.042	1.892

TABLE V

±30° Dataset.

(a) DDF.

Line	$\Delta\theta$	Δc
line 0	0.017	2.937
line 1	0.001	0.661
line 2	0.016	1.326
line 3	0.004	0.679
line 4	0.005	0.696
line 5	0.110	13.087
line 6	0.009	0.746
line 7	0.129	3.908
all	0.036	3.005

(b) SDF.

Line	$\Delta\theta$	Δc
line 0	0.020	2.608
line 1	0.001	0.017
line 2	0.011	2.900
line 3	0.004	0.039
line 4	0.007	0.067
line 5	0.048	8.011
line 6	0.035	0.523
line 7	0.118	3.306
all	0.030	2.184

TABLE VI

±45° Dataset.

(a) DDF.

Line	$\Delta\theta$	Δc
line 0	0.020	0.503
line 1	0.001	0.776
line 2	0.027	4.243
line 3	0.004	0.726
line 4	0.003	0.726
line 5	0.169	5.202
line 6	0.013	0.958
line 7	0.144	10.015
all	0.048	2.894

(b) SDF.

Line	$\Delta\theta$	Δc
line 0	0.019	0.051
line 1	0.001	0.020
line 2	0.023	0.451
line 3	0.004	0.038
line 4	0.004	0.043
line 5	0.243	5.301
line 6	0.018	0.279
line 7	0.348	17.242
all	0.082	2.928

TABLE VII

±0° Inferenced Dataset.

(a) DDF.

Line	$\Delta\theta$	Δc
line 0	0.027	0.342
line 1	0.026	1.904
line 2	0.118	4.558
line 3	0.052	1.181
line 4	0.063	1.098
line 5	0.174	2.353
line 6	0.060	1.787
line 7	0.318	9.822
all	0.105	2.881

(b) SDF.

Line	$\Delta\theta$	Δc
line 0	0.029	0.284
line 1	0.026	1.294
line 2	0.111	3.248
line 3	0.056	0.811
line 4	0.131	1.308
line 5	0.397	1.911
line 6	0.056	1.231
line 7	0.247	7.648
all	0.132	2.217

TABLE VIII

±15° Inferenced Dataset.

(a) DDF.

Line	$\Delta\theta$	Δc
line 0	0.082	0.709
line 1	0.008	1.090
line 2	0.177	6.830
line 3	0.058	1.239
line 4	0.080	1.542
line 5	0.374	11.272
line 6	0.195	6.883
line 7	0.191	7.166
all	0.146	4.591

(b) SDF.

Line	$\Delta\theta$	Δc
line 0	0.344	5.134
line 1	0.008	0.522
line 2	0.176	4.795
line 3	0.208	1.822
line 4	0.082	1.043
line 5	0.434	4.468
line 6	0.183	9.790
line 7	0.254	8.816
all	0.211	4.549

TABLE IX

±30° Inferenced Dataset.

(a) DDF.

Line	$\Delta\theta$	Δc
line 0	0.129	3.855
line 1	0.007	1.045
line 2	0.268	8.954
line 3	0.220	2.235
line 4	0.167	2.976
line 5	0.217	8.159
line 6	0.365	8.048
line 7	0.477	14.578
all	0.231	6.231

(b) SDF.

Line	$\Delta\theta$	Δc
line 0	0.609	8.028
line 1	0.014	0.569
line 2	0.237	8.209
line 3	0.217	1.671
line 4	0.162	2.302
line 5	0.198	4.050
line 6	0.383	11.369
line 7	0.478	14.688
all	0.287	6.361

TABLE X

±45° Inferenced Dataset.

(a) DDF.

Line	$\Delta\theta$	Δc
line 0	0.072	0.887
line 1	0.009	1.183
line 2	0.087	2.501
line 3	0.152	2.079
line 4	0.246	4.069
line 5	0.853	12.442
line 6	0.866	16.492
line 7	0.822	34.402
all	0.388	9.257

(b) SDF.

Line	$\Delta\theta$	Δc
line 0	0.072	0.508
line 1	0.009	0.501
line 2	0.129	3.277
line 3	0.156	1.514
line 4	0.435	6.670
line 5	0.835	12.929
line 6	1.072	19.055
line 7	0.788	31.650
all	0.437	9.513

[29] M. Wada, Y. Ueda, M. Adachi, and R. Miyamoto, "Area-wise augmentation on segmentation datasets from 3d scanned data used for visual navigation," in *Proc. CoDIT*, 2024, pp. 1499–1504.