360-Degree Video Generation by Animating Vehicles in a Single 360-Degree Image

Ryota Kimura[1,*], Norihiko Kawai[1,*]

[1]Graduate School of Information Science and Technology, Osaka Institute of Technology, Kitayama 1-79-1, Hirakata, Osaka, Japan

Abstract

We propose a method for generating a video that reproduces the motion of vehicles present in a single omnidirectional image. In the proposed method, vehicles are first detected, and a background omnidirectional image is created by removing the detected vehicles. In parallel, 3D models of the detected vehicles are generated. The direction of vehicle travel is then estimated from the road region in the omnidirectional image, and an omnidirectional video is generated by animating the 3D models accordingly within a virtual space consisting of a sphere onto which the background image is mapped. Through two experiments conducted under different road geometry conditions, we confirmed that the vehicles moved while remaining within the road region.

Keywords

VR, Omnidirectional Image, Video Generation, 3D Model, Inpainting, Road Analysis, Motion Reproduction

1. Introduction

Omnidirectional images have been utilized in a wide range of contexts with the widespread use of smartphones and virtual reality (VR) devices. A prominent example of such applications is Google Street View [1], which enables users to virtually explore cities and regions around the world via the internet using omnidirectional images. However, since the presented omnidirectional images are static, this service suffers from a lack of presence and immersion. One potential solution to this limitation is to capture video at fixed locations and present dynamic imagery instead of still images; however, acquiring the vast amount of video data required to cover broad areas across the globe demands an enormous amount of time and resources. To address this challenge, this study proposes a method for generating dynamic imagery that reproduces the motion of vehicles from a single omnidirectional image.

Existing methods for video generation from a single image can be broadly classified into two categories: those targeting the motion of non-fluid objects [2, 3, 4], and those targeting the motion of fluid elements [5, 6, 7]. The former category focuses primarily on predicting and generating the motion of objects such as vehicles and pedestrians, whereas the latter is designed to reproduce the motion of fluid phenomena, such as fire and water, with high fidelity. However, since these methods are designed for perspective projection images, applying them directly to omnidirectional images represented in the equirectangular projection leads to unnatural visual artifacts when the resulting imagery is converted back to perspective projection images.

As a method employing omnidirectional images relevant to this study, Wang et al. [8] proposed an approach for generating omnidirectional video based on prompt and motion conditions. This method generates omnidirectional video that accounts for the distortion inherent to equirectangular projection. However, it is not designed for the purpose of animating existing omnidirectional images acquired from sources such as Google Street View. In contrast, a prior study [9] focusing on natural elements within existing omnidirectional images proposed a method for generating omnidirectional video that reproduces the motion of sky, water and tree regions in omnidirectional still images. In this method, landscape images are taken as input, and semantic segmentation is applied to the omnidirectional image to partition the scene into regions to be animated and regions to remain static. Motion is then

APMAR'26: The 18th Asia-Pacific Workshop on Mixed and Augmented Reality, Aug. 3-5, 2026, Taipei, Taiwan

*Corresponding author.

m1m02530114@oit.ac.jp (R. Kimura); norihiko.kawai@oit.ac.jp (N. Kawai)

0000-0002-7859-8407 (N. Kawai)

Figure 1: Example of an input omnidirectional image

introduced into the segmented sky, water, and tree regions, while all remaining regions are replaced with the original static input image, thereby producing omnidirectional videos.

In contrast, this study targets the animation of non-fluid objects, specifically vehicles. The proposed method detects vehicles within the omnidirectional image and removes the detected vehicles from the image. In parallel, it generates 3D models of the detected vehicles. Furthermore, road regions are extracted from the omnidirectional image to determine the direction of vehicle movement. Based on this information, motion is applied to the 3D models in a virtual space consisting of a sphere onto which the background image is mapped, thereby generating omnidirectional video.

2. Proposed Method

The proposed method takes as input an omnidirectional image containing at least one vehicle. The omnidirectional image is then converted into a set of perspective projection images at regular angular intervals along the horizontal direction, from which vehicle regions are detected and their corresponding mask images are generated. Using these mask images, the vehicles are removed by inpainting, and the perspective images are converted back into the omnidirectional format and composited onto the original omnidirectional image. In parallel, 3D models of the detected vehicles are generated. The road region is then extracted from the omnidirectional image, and the direction of vehicle movement is determined accordingly. Finally, an omnidirectional video showing the vehicles in motion is generated by animating the 3D vehicle models in the determined directions within a virtual space consisting of a sphere onto which the omnidirectional image is mapped. Each of these processing steps is described in detail in the following sections.

2.1. Conversion to Perspective Projection Images

Given an input omnidirectional image such as that shown in Fig. 1, the image is converted into perspective projection images at regular angular intervals along the horizontal direction of the camera, as indicated by the framed regions in Fig. 2. This conversion produces multiple perspective projection images that are used in the subsequent processing steps. An example of the conversion of the region indicated by the red frame into a perspective projection image is shown in Fig. 3.

2.2. Object Detection and Mask Image Generation

Next, the segmentation model of YOLOv8 [10] is applied to extract vehicle regions from the perspective projection images, as shown in Fig. 4. This process is performed for all generated perspective projection images. Subsequently, mask images to be used in the subsequent processing steps are generated. Fig. 5 shows an example of a mask image generated from the extracted region shown in Fig. 4. During region segmentation, the segmentation model may fail to extract regions accurately, resulting in black areas

Figure 2: Example of conversion to perspective projection images

Figure 3: Example of a perspective projection image

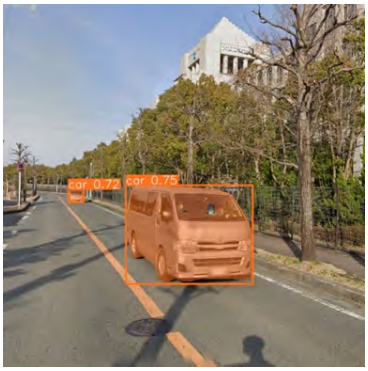


Figure 4: Example of object detection using YOLOv8

Figure 5: Example of a vehicle region mask image

remaining within the vehicle region, as shown in Fig. 5. Furthermore, shadow regions cast by vehicles on the road surface are often not included in the extracted regions by the segmentation. To address these issues, a dilation operation is applied to the generated mask images, as illustrated in Fig. 6, and the resulting dilated mask images are used for inpainting. In addition, as shown in Fig. 7, images are generated by extracting only the regions corresponding to the dilated masks from the color perspective projection images. These mask and color images are used for inpainting and 3D model generation, which are described in later sections.

2.3. Generation of an Omnidirectional Image with Vehicles Removed

To prevent the vehicles originally present in the image from remaining after they have moved, the vehicles are removed by inpainting. Here, DeepFillv2 [11] is applied to the perspective images by treating

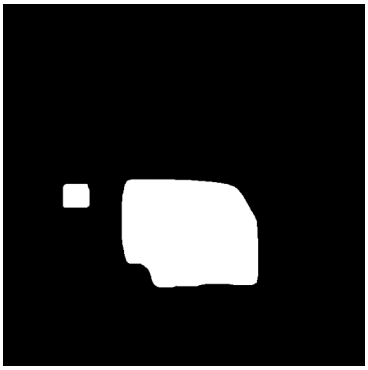


Figure 6: Example of dilation operation

Figure 7: Example of mask region extraction

Figure 8: Example of inpainting

Figure 9: Example of an omnidirectional image with vehicles removed

the dilated mask regions as missing regions, resulting in perspective images with the vehicles removed, as shown in Fig. 8. Subsequently, the perspective images are converted back into the omnidirectional format and composited onto the omnidirectional image, thereby generating an omnidirectional image with vehicles removed, as shown in Fig. 9.

2.4. 3D Model Generation

For vehicles whose area detected in Section 2.2 exceeds a certain threshold, 3D models are generated using the perspective projection image from which the vehicle region has been extracted, as shown in Fig. 7. Here, TripoAI [12] is employed for 3D model generation. TripoAI is a web-based application that enables the generation of 3D models from text or image inputs. Fig. 10 shows an example of a generated 3D model from Fig. 7.

Figure 10: Example of 3D model generation using TripoAI

Figure 11: Mask image of the road region

2.5. Detection of Lane Markings on the Road

Assuming that the lower portion of the omnidirectional image in the vertical direction corresponds to the road surface, lane markings on the road are detected. Semantic segmentation [13] is applied to the omnidirectional image with vehicles removed to produce a mask image in which the road region is designated as the target area, as shown in Fig. 11. Next, as illustrated in Fig. 12, a coordinate system is defined with the omnidirectional image capture position as the origin and the zenith direction of the omnidirectional image as the Y-axis. In addition, we assume that the Z-axis direction is the approximate traveling direction of the road when the omnidirectional image is projected onto a sphere in a virtual space. Using this coordinate system, the road region is divided into partial planes along the Z-axis: a red partial plane representing the road surface in the immediate vicinity, and blue partial planes representing the road surface extended in the forward and backward directions, as shown in Fig. 12. The omnidirectional image and its mask image are projected onto these planes to generate bird's-eye view images of the road for each partial plane, as shown in Figs. 13 and 14. Subsequently, edge detection using the Canny method is applied to the generated bird's-eye view images, and the logical AND operation with the road bird's-eye view mask image is performed to detect the lane markings on the road, as shown in Fig. 15.

2.6. Estimation of Vehicle Motion

The orientation of the detected lane markings is computed and used as the motion direction of the 3D model in the virtual space. First, the Hough transform is applied to the detected edges to identify straight lines, as shown in Fig. 16. Next, a unit vector is computed for each partial plane from the detected lines. Since the three planes are connected as shown in Fig. 17, the initial position of the vehicle is tentatively set at the left edge, and the coordinates at which the vehicle reaches the boundaries between adjacent partial planes are computed by sequentially moving it along the unit vector of each

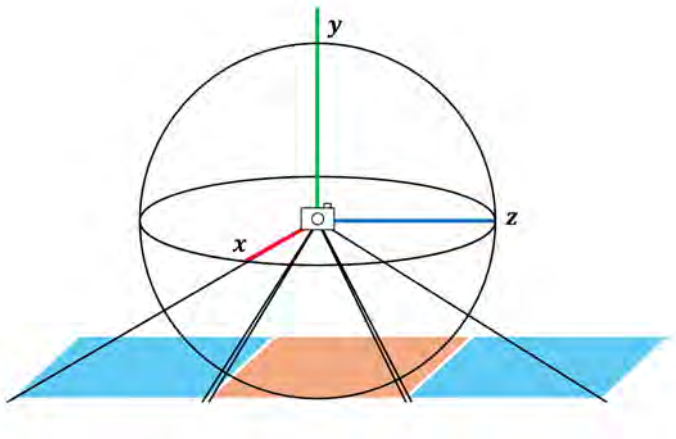


Figure 12: Generation of bird's-eye view images of the road

Figure 13: Example of a bird's-eye view image of the road

Figure 14: Example of a road bird's-eye view mask image

Figure 15: Example of lane marking extraction

partial plane, as indicated by the red dots in Fig. 17. These points are then interpolated using cubic spline interpolation to form a smooth curve.

Furthermore, the three-dimensional coordinates (X, Y, Z) of the vehicle in the virtual space are computed from the pixel coordinates (p_1, q_1) in the omnidirectional image corresponding to the two-dimensional coordinates at the lower part of the center of the bounding box of the vehicle detected in the perspective projection image in Section 2.2, using the following equations:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} T\tan\left(\frac{\pi q_1}{h}\right)\sin\left(\frac{2\pi p_1}{w}\right) \\ T \\ T\tan\left(\frac{\pi q_1}{h}\right)\cos\left(\frac{2\pi p_1}{w}\right) \end{bmatrix}, \tag{1}$$

Figure 16: Example of straight line detection

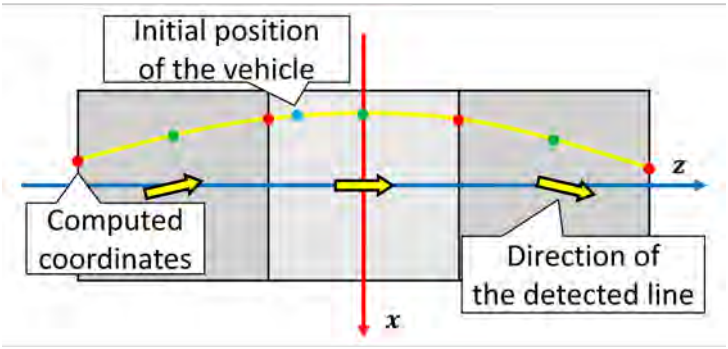


Figure 17: Estimation of motion direction on the coordinate plane

where w and h denote the width and height of the omnidirectional image, respectively, and T is a negative constant representing the relative height of the ground with respect to the omnidirectional image capture position. Subsequently, to correct for the offset between the motion trajectory and the initial position of the vehicle, the two-dimensional curve on the road plane representing the motion trajectory is converted into a three-dimensional curve satisfying $Y = T$, and this curve is translated in the direction of the X-axis so that it passes through the vehicle's position, thereby determining the final motion path of the vehicle. The orientation of the vehicle is also updated accordingly based on the direction of motion.

2.7. Light Source Position Estimation and Shadow Rendering

Assuming that the omnidirectional image is captured outdoors, the direction of parallel light rays used in the virtual space is determined. The pixel with the maximum luminance in the omnidirectional image is identified, and the centroid of the region with the greatest number of maximum-luminance pixels is computed. The direction from the three-dimensional position on the sphere's surface corresponding to the computed centroid coordinates to the origin of the sphere is calculated, and parallel light rays are configured accordingly based on this direction. Furthermore, to reproduce shadows, a transparent planar object is placed in the virtual space at $Y = T$, coinciding with the road plane, as shown in Fig. 18. The shadows of the vehicle 3D models are then rendered onto this transparent plane to achieve realistic shadow reproduction. The shading of the vehicle is also represented using the parallel light rays.

2.8. Video Generation

An omnidirectional video is finally generated by constructing a virtual space consisting of a sphere onto which the omnidirectional image with vehicles removed, generated in Section 2.3, is mapped as the background. In the virtual space, the 3D models based on the detected vehicles are animated along

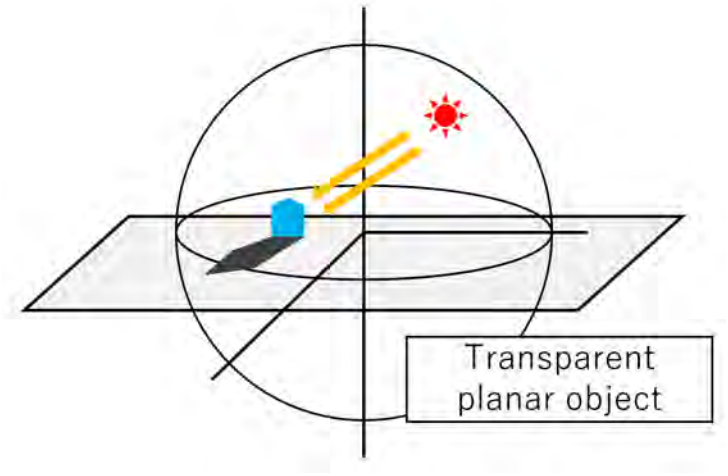


Figure 18: Shadow rendering in the virtual space

Figure 19: Input image for Experiment 2

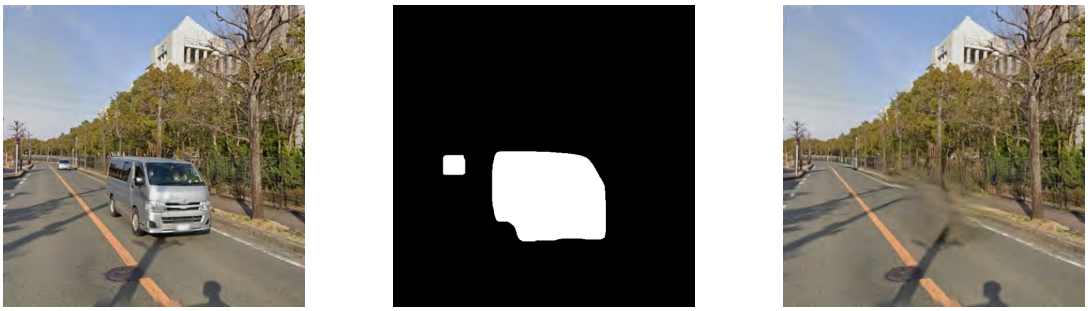
the coordinates computed in Section 2.6 to reproduce their motion, and the scene is rendered from a virtual camera placed at the center of the sphere.

3. Experiments and Discussion

Experiments on video generation from a single omnidirectional image were conducted. Experiment 1 uses the omnidirectional image shown in Fig. 1, represented in the equirectangular projection used in Google Street View, while Experiment 2 uses the image shown in Fig. 19. Furthermore, Experiment 1 features a straight road, whereas Experiment 2 features a gently curving road, such that the two experiments differ in road geometry.

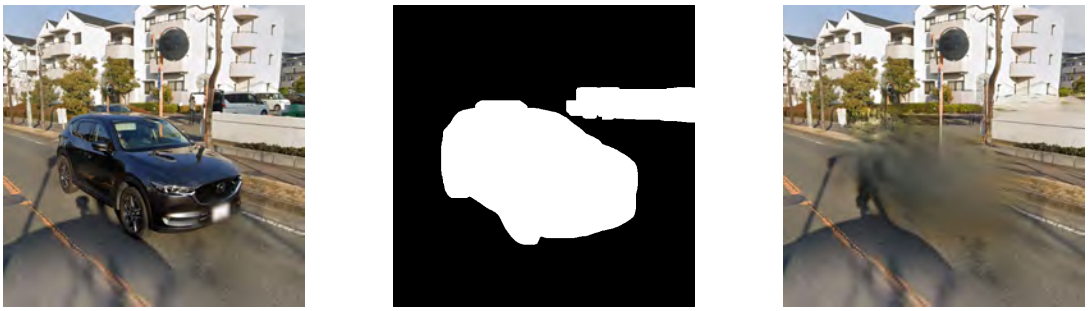
The perspective projection images containing the detected vehicles, the corresponding mask images, and the inpainting results for Experiments 1 and 2 are shown in Figs. 20 and 21, respectively. Furthermore, the omnidirectional images with vehicles removed, used as background images, are shown in Figs. 22 and 23, respectively. From the results, we confirmed that the vehicle regions were appropriately detected in both Experiments 1 and 2, and that background images with vehicles removed were generated by applying inpainting to the corresponding regions.

Furthermore, the bird's-eye view images used for movement trajectory estimation and the corresponding detected line results for Experiments 1 and 2 are shown in Figs. 24 and 25, respectively. The detected lines are represented by red lines overlaid on the bird's-eye view images. Since Experiment 1 features a straight road, the variation in slope across the partial planes is small. In contrast, since Experiment 2 features a gently curving road, the slopes of the detected lines change gradually, confirming that the slopes of the detected lane markings vary in accordance with differences in road geometry.



(a) Input image (b) Vehicle mask (c) Result of inpainting

Figure 20: Vehicle detection results for Experiment 1



(a) Input image (b) Vehicle mask (c) Result of inpainting

Figure 21: Vehicle detection results for Experiment 2

Figure 22: Resulting omnidirectional image for Experiment 1

Figure 23: Resulting omnidirectional image for Experiment 2

(a) Back view (b) Middle view (c) Front view

Figure 24: Movement trajectory estimation results for Experiment 1

(a) Back view (b) Middle view (c) Front view

Figure 25: Movement trajectory estimation results for Experiment 2

Subsequently, frames 1, 120, 240, and 360 of the omnidirectional videos represented in the equirectangular projection generated in Experiments 1 and 2 are shown in Figs. 26 and 27, respectively. These experimental results demonstrate that the motion of vehicles moving along the road can be reproduced by animating the 3D models in accordance with the detected lane markings.

To verify the effectiveness of the proposed trajectory estimation method, we compared the result of Experiment 2 with the result obtained by applying the movement trajectory estimated in Experiment 1 to the vehicles in Experiment 2. Fig. 28 shows a comparison of perspective projection views from the center. The figure shows that the vehicle using the trajectory estimated in Experiment 1 deviates onto the sidewalk, whereas the vehicle in Experiment 2 remains within the road region. These results confirm that the motion generated by the proposed method is appropriate.

Furthermore, perspective projection images viewed from the center at frames 120 and 240 of the generation result of Experiment 1 are shown in Fig. 29. From these images, we confirmed that the direction of the shadows cast by the vehicles in the video is consistent with the direction of the shadows of the vegetation appearing in the omnidirectional image, thereby suggesting that the shadow rendering in the proposed method is appropriate.

4. Conclusion

In this study, we proposed a method for generating omnidirectional videos in which vehicles are animated from a single omnidirectional image. In the proposed method, a background image is generated by detecting and removing the vehicles in the omnidirectional image, and is used to construct a virtual space. In parallel, 3D models of the detected vehicles are generated. The direction of travel is estimated using lane markings on the road, and the vehicles are animated along the estimated trajectory within the virtual space and rendered to generate an omnidirectional video in which the vehicles move.

Through experiments conducted under both straight and curved road conditions, we confirmed that the vehicles moved without deviating from the road region, demonstrating the effectiveness of the

(a) Frame 1

(b) Frame 120

(c) Frame 240

(d) Frame 360

Figure 26: Generation results of Experiment 1

(a) Frame 1

(b) Frame 120

(c) Frame 240

(d) Frame 360

Figure 27: Generation results of Experiment 2

(a) Generation results of Experiment 2

(b) Generation result with movement trajectory of Experiment 1 applied

Figure 28: Generation results comparing movement trajectories

(a) Frame 120

(b) Frame 240

Figure 29: Perspective projection images of the generation result of Experiment 1

proposed method. However, since the proposed method relies on lane markings on the road, it is difficult for the method to be applied to roads without them. In future work, we plan to estimate plausible motion trajectories even on roads without clear lane markings and generate more natural videos by extending the target objects beyond vehicles to include other moving objects, such as pedestrians.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Claude in order to: Grammar and spelling check, Paraphrase and reword. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Google, Google Street View, 2026. URL: https://www.google.com/streetview/, accessed: 2026-06-16.

[2] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, S. Levine, Stochastic variational video prediction, in: Proceedings of the International Conference on Learning Representations (ICLR), 2018.

[3] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, N. Sebe, First order motion model for image animation, Advances in Neural Information Processing Systems 32 (2019).

[4] L. Zhao, X. Peng, Y. Tian, M. Kapadia, D. Metaxas, Learning to forecast and refine residual motion for image-to-video generation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 387–403.

[5] Y. Endo, Y. Kanamori, S. Kuriyama, Animating landscape: Self-supervised learning of decoupled motion and appearance for single-image video synthesis, ACM Transactions on Graphics 38 (2019) 175:1–175:19.

[6] A. Holynski, B. L. Curless, S. M. Seitz, R. Szeliski, Animating pictures with eulerian motion fields, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 5810–5819.

[7] M. Okabe, K. Anjyor, R. Onai, Creating fluid animation from a single image using video database, Computer Graphics Forum 30 (2011) 1973–1982.

[8] Q. Wang, W. Li, C. Mou, X. Cheng, J. Zhang, 360DVD: Controllable panorama video generation with 360-degree video diffusion model, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 6913–6923.

[9] N. Kawai, M. Kakuho, 3D-aware motion generation of natural objects in a single omnidirectional image for virtual sightseeing, IEICE Transactions on Information and Systems E109.D (2026) 853–862. doi:10.1587/transinf.2025HCP0005.

[10] Ultralytics, YOLOv8 documentation, https://docs.ultralytics.com/ja/models/yolov8/, 2024. Accessed: 2024-09-09.

[11] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T. S. Huang, Free-form image inpainting with gated convolution,

in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 4471–4480.

[12] TripoAI, TripoAI: 3D model generation platform, https://www.tripo3d.ai/, 2025. Accessed: 2025-02-04.

[13] J. Lambert, Z. Liu, O. Sener, J. Hays, V. Koltun, MSeg: A composite dataset for multi-domain semantic segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 2879–2888.