2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

The ParallelEye-CS Dataset: Constructing Artificial Scenes for Evaluating the Visual Intelligence of Intelligent Vehicles

Xuan Li, Yutong Wang, Kunfeng Wang, *Senior Member, IEEE*, Lan Yan, and Fei-Yue Wang, *Fellow, IEEE*

Abstract— **Offline training and testing are playing an essential role in design and evaluation of intelligent vehicle vision algorithms. Nevertheless, long-term inconvenience concerning traditional image datasets is that manually collecting and annotating datasets from real scenes lack testing tasks and diverse environmental conditions. For that virtual datasets can make up for these regrets. In this paper, we propose to construct artificial scenes for evaluating the visual intelligence of intelligent vehicles and generate a new virtual dataset called "ParallelEye-CS". First of all, the actual track map data is used to build 3D scene model of Chinese Flagship Intelligent Vehicle Proving Center Area, Changshu. Then, the computer graphics and virtual reality technologies are utilized to simulate the virtual testing tasks according to the Chinese Intelligent Vehicles Future Challenge (IVFC) tasks. Furthermore, the Unity3D platform is used to generate accurate ground-truth labels and change environmental conditions. As a result, we present a viable implementation method for constructing artificial scenes for traffic vision research. The experimental results show that our method is able to generate photorealistic virtual datasets with diverse testing tasks.**

I. Introduction

In recent years, self-driving technology [1] has become a hot research field. Because of the high risk, immeasurable cost and long test period of real experiments, offline training and testing have become important steps in intelligent vehicle research. The traditional approach to evaluating intelligent vehicles is to collect large amounts of annotated scene data, which can train and evaluate the models offline [2-4]. After passing the offline test, the real road tests are conducted with models loaded into vehicles' memory. Due to the good performance of deep convolutional neural network (DCNN) achieved in computer vision tasks, intelligent vehicles use DCNN to conduct visual perception. As we know, DCNN has a large demand for large-scale annotated and diverse dataset. However, it is very time-consuming to get useful real datasets. Also there are some defects in the current real datasets. For instance, KITTI [5] and Cityscapes [6] datasets are of small changes in the weather, and do not contain challenging tasks. At the same time, the collection and annotation of large-scale complex traffic scene data are difficult and prone to error, which become the reason to hinder the further development of intelligent vehicle.

In view of the limitations of real datasets, more high-tech companies (Google and Microsoft) and researchers begin to use virtual image training and evaluating the visual intelligence of intelligent vehicles. As early as 1989, Pomerleau [7] used virtual road images to train the multilayer neural network model, and they hoped to find a new intelligent navigation platform. After nearly 30 years of development, virtual data can be derived from three sources:

(1) Using 2D real scene data for generating or synthesizing 2D virtual image [4],[8]. Gaidon *et al.* [8] propose a real-to-virtual cloning method to synthesize virtual images, and change their weather conditions. Then they validate the effectiveness of virtual data in tracking model pre-training. But images generated in this way are highly dependent on real images, and look similarly to real ones, which cannot satisfy the diversity need.

(2) Using computer games for obtaining virtual images [9-14]. Chen [10] use players video images to train convolutional neural network, and evaluate it in Torcs [12]. However, Torcs can only simulate autodrome-like environment, rather than the complex city roads. Richter *et al.* [14] use the commercial game Grand Theft Auto V to generate images, and annotate them to evaluate the performance of semantic segmentation algorithm. However, the Grand Theft Auto V is not open-source, and do not support detailed annotation, environment customization.

(3) Constructing 3D virtual scene for obtaining virtual images [15-19]. Dosovitskiy *et al.* [15] use UE4 to build a complete virtual city scene, and test the performance of different algorithms in autonomous driving. Google's autonomous car subsidiary Waymo builds a virtual world "Carcraft".

Recent studies also show that images generated by GANs and selected by people have been comparable to real images in fidelity and diversity, which it a potential method to generate virtual images. Wang *et al.* train conditional GAN to generate high-resolution photo-realistic images [20].

This work was supported partly by National Key R&D Program of China under Grant 2018YFB1004800, and partly by National Natural Science Foundation of China under Grant 61533019 and Grant 71232006.

Xuan Li is with the School of Automation, Beijing Institute of Technology, Beijing 100081, China, and also with The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: lixuan0125@126.com).

Yutong Wang and Lan Yan are with The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with University of Chinese Academy of Sciences, Beijing 100049, China.

Kunfeng Wang (Corresponding author) is with The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the Innovation Center for Parallel Vision, Qingdao Academy of Intelligent Industries, Qingdao 266000, China (e-mail: kunfeng.wang@ia.ac.cn).

Fei-Yue Wang is with The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the Research Center for Computational Experiments and Parallel Systems Technology, National University of Defense Technology, Changsha 410073, China (e-mail: feiyue@gmail.com).

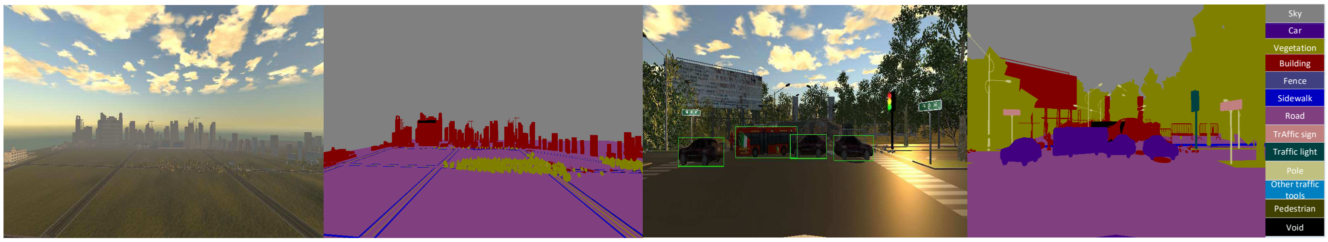

Fig. 1. The ParallelEye-CS dataset. A general view of the constructed artificial scenes (Left 1), the corresponding semantic labels (Left 2), a sample frame with detection bounding boxes (Left 3), and its semantic labels (Left 4). Best viewed with zooming.

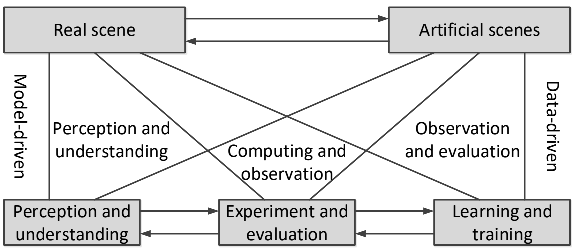

Fig. 2. Basic framework of parallel vision [21].

However, GANs lack sufficient theoretical support and are error-prone, which result in local deformation and distortion.

In order to solve these problems, this paper proposes a artificial scene for evaluating the visual intelligence of intelligent vehicles. First of all, we use map data to build the 3D scene model of Chinese Flagship Intelligent Vehicle Proving Center (IVPC) area, Changshu, Suzhou, China. Then, we use different computer graphics tools (3D max, google 3D warehouse) to create a realistic, large-scale virtual testing field, in which the fidelity and geographic information can match the real world well. Furthermore, we use Unity3D to automatically generate accurate ground-truth labels including pixel-level semantic/instance segmentation, object bounding box, tracking, depth and so on. The environmental conditions in artificial scenes can be controlled completely. In consequence, we generate a new virtual image dataset, called "ParallelEye-CS" (see Fig. 1). We will build a video and show more details for this dataset (http://openpv.cn/datasets.php). Finally, the experimental results show that our proposed method is able to generate lifelike virtual images with high diversity.

The rest of this paper is organized as follows. Section II introduces the importance of parallel vision and virtual dataset for evaluating the visual intelligence. Section III presents our method to constructing artificial test scenes and generating virtual images with ground-truth labels. Section IV reports the experimental results and analyzes the performance. Finally, the conclusions are made in section V.

II. Parallel Vision and Virtual Dataset

Parallel vision [21-23] is an extension of the ACP (Artificial societies, Computational experiments, and Parallel execution) theory [24,25] in computer vision field. In parallel vision, artificial scenes are used to model complex real scenes, computational experiments are utilized to learn and evaluate a variety of vision models, and parallel execution is conducted to optimize the vision system online and realize perception and understanding of complex scenes. The basic framework of parallel vision [21] is shown in Fig. 2. Based on the parallel vision theory, we construct a large-scale virtual urban road network and generate a large number of photo-realistic images.

The first stage of parallel vision is to construct artificial scenes by simulating real scenes under various environmental conditions, and produce large-scale diverse datasets with precise annotations generated automatically. Generally speaking, the construction of artificial scenes can be regarded as "video game design", i.e. using computer animation technology to model artificial scenes. The main technologies used in this stage include computer graphics, virtual reality, and micro-simulation. Computer graphics and computer vision, on the whole, can be considered as a pair of forward and inverse problems. The goal of computer graphics is to synthesize image measurements given the description of world parameters according to physics-based image formation principles (forward inference), while the focus of computer vision is to map the pixel measurements to 3D scene parameters and semantics (inverse inference). Apparently their goals are opposite, but can converge to a common point: parallel vision.

From the parallel vision perspective, we design the ParallelEye-CS dataset. ParallelEye is collected in artificial scenes constructed according to urban network of Changshu area, Suzhou. Unity3D is used to control the environmental conditions in the scene. There are 12 object classes in ParallelEye-CS, reflecting the common elements of traffic scenes, including sky, buildings, cars, roads, sidewalks, vegetation, traffic signs, traffic lights, poles, pedestrians, an so on. These object classes can be automatically annotated to generate pixel-level semantics. For traffic vision research, we pay special attention to instance segmentation, with each object of interest segmented automatically. In addition,

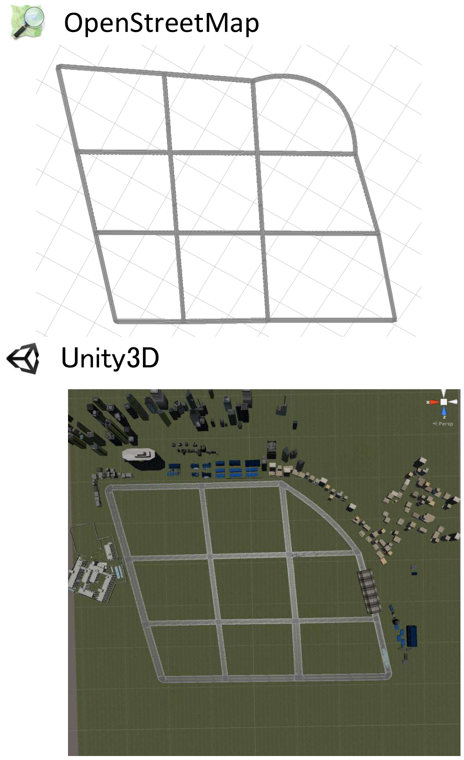

Fig. 3. Pipeline for generating the ParallelEye-CS dataset with OpenStreetMap, and Unity3D.

ParallelEye-CS provides accurate ground truth for object detection and tracking, depth, and so on.

III. Approach

The main purpose of this paper is to create the ParallelEye-CS dataset, which can assist the environmental perception of self-driving. Specifically, we construct a virtual test scene, and then get the virtual traffic images and label information. The virtual scene collected annotation information of objects around the intelligent vehicle, including cars, buses and trucks. Therefore, the virtual dataset can help evaluate the visual performance of the algorithm of intelligent vehicle. In the next section, we will introduce the main steps of constructing ParallelEye-CS.

(1)To transform the actual track into the corresponding virtual track information, and add the basic model (including roads, lanes, steering signs, sidewalks, an so on);

(2)To introduce the main target objects (cars, buses, trucks) into virtual track, and add movement rules for these objects;

(3)To simulate the virtual scene test tasks according to "Chinese Intelligent Vehicles Future Challenge" IVFC tasks (U turn, avoid collisions with pedestrians and vehicles, intelligent vehicle start-up, and so on);

(4)To generate accurate and efficient annotation information, automatically.

(5)To improve the diversity of artificial scenes

A. Correspondence of Artificial and Real Scenes

IVFC will be held at the Chinese Flagship Intelligent Vehicle Proving Center (IVPC), Changshu, Suzhou, China, which is a county-level city located in the lower reaches of the Yangtze River in Jiangsu Province. The testing region for self-driving vehicles consists of eight roads. In fact, intelligent vehicles need GPS information to determine the route. According to the actual situation, the vehicle will plan the route. First, we make full use of OpenStreetMap (OSM) data, realizing location mapping between virtual and real world, which can help the virtual scene more realistic. Then, we put the virtual path information into CityEngine, which can be used for rapid modeling. According to the real road information, we use CityEngine to add basic models for the virtual road, including lanes, streets and sidewalks and buildings.

B. Introducing Foreground Objects into Virtual City

There are some differences between the virtual scene and real scene. In order to solve this problem, we adopt the following two ways to introduce new models, and thus making the virtual scene more realistic. First and foremost, the scene makes full use of the sharing model. For example, Google 3D Warehouse provides a huge and rich 3D model (static and dynamic objects). Second, using 3Dmax software, we construct buildings, road signs, street lighting, buildings for virtual scene. We import the virtual scene and other elements into the Unity3D development platform (see Fig. 3). This can help us build efficient and realistic virtual scene. Because the virtual scene contains a large number of custom elements, we can simulate the track and the surrounding environment more efficiently and realistically.

C. Design of Virtual Testing Task

The virtual test scene including three video sequences (01,02,03). In sequence 1: at the starting point, intelligent vehicle starts moving when the signal light is green, Then, If other working vehicles stay on the road, the intelligent vehicle will change lanes to avoid collision; In sequence 2, Intelligent vehicle runs at a constant speed, and avoids collision with a truck at the intersection; In sequence 3, Self-driving vehicle drives in harsh conditions (flood, smoke, cross-country, tunnel and so on); Intelligent vehicle avoid collision with a pedestrian at the intersection. Next, we will introduce the specific design of each virtual test task. Using the C# scripts, we control those models to move according to the traffic rules in virtual world. Also, with the help of the collision system in Unity3D, the vehicles and pedestrians are adjusted as needed. In ParallelEye-CS_01, the departure time of the intelligent vehicle is related to the change of traffic lights. The duration of red light, yellow light and green light is 10 seconds, 5 seconds and 10 seconds, respectively. When the light turns green, the intelligent vehicle starts in straight driving. When encounters a working vehicle, it changes lane to bear off the vehicle. In ParallelEye-CS_02, the collision detection system on the intelligent vehicle senses the truck in advance, and issues instruction to slow to a stop. After the truck passes through, until the collision detection system cannot sense the truck, the intelligent vehicle restarts and makes a u turn. In ParallelEye-CS_03, the intelligent vehicle encounters challenging tasks, such as water (pro or simple), heavy fog (particle system) and rocky ground, which make the collision detection system sense the distraction respectively. Then the intelligent vehicle stops in front of the distraction and confirms to pass. When it enters the tunnel, the intelligent vehicle slows down and passes through the tunnel. Finally, a pedestrian appears at the pedestrian crossing, and the collision detection system senses the pedestrian and the vehicle stops to wait. Until the pedestrian passes through, the intelligent vehicle restarts.

D. Generation of Ground-Truth Annotations

The off-line testing of intelligent vehicle depends on a large number of accurate annotation data. ParallelEye-CS can provides diverse annotation information, namely object detection, tracking, instance segmentation, depth and normal (see Fig. 4). Traditionally, the images were annotated by hand. However, the manual annotation is time-consuming and prone to error. Instead of manual annotation, this paper uses Unity3D to automatically generate accurate ground-truth labels. We generate object detection ground truth based on these rules: 1) when target model appears in camera view, the two dimensional bounding box of the model will be drawn 2) In the screen, all target models smaller than 15 pixels will be ignored. 3) when occlusion occurs and the occlusion rate is higher than a threshold (80 percent), we do not draw bounding boxes for the occluded object. In order to record the ground truth, we use a green rectangular box to draw the detection ground truth for each model. We also assign rectangular frames of different colors to record the tracking ground truth for each object instance. Semantic segmentation ground truth can be directly generated by using unlit shaders on the materials of the models, and each category corresponds to output of a color. Instance segmentation ground truth uses the same method, the only difference is that different models of the same category are expressed in different colors. The modified shaders output a color which is not affected by the lighting and shading conditions. Depth ground truth is generated using built-in depth buffer information to get depth data for screen coordinates. The depth ranges from 0 to 1 with a nonlinear distribution, with 1 representing infinitely distant.Normal maps are a special kind of texture that allow us to show surface detail.

E. Diversity of Artificial Scenes

Traditionally, video image datasets are collected in the real world or retrieved from the Internet. It is impossible to control the environmental conditions and repeat the scene layout under different environments. Thus it is difficult to isolate the effects of environmental conditions on the performance of computer vision algorithms. By contrast, it is easy to control the environmental conditions in artificial scenes. In order to build a more robust virtual scene, which can enhance the evaluating of the adaptability and diversity of visual algorithms. We set up different collection conditions for each test task. The specific settings are as follows: we designed different environmental conditions (cloudy, sunny and fog) and illumination (from sunrise to sunset) conditions for the artificial scene (see Fig. 5). In addition, we set the speed, direction and track for the cars. Finally, we are able to configure the camera angle flexibly. (i.e. 0, 15, 30 degree with respect to the moving direction). These subtle changes simulate different environmental conditions in the virtual world, and would otherwise need the expensive process of re-acquiring and re-labeling images of the real world. The advantage of this setting is that it can increase diversity of the ParallelEye-CS dataset. Fig. 5 illustrates the diversity of artificial scenes in terms of illumination and weather conditions.

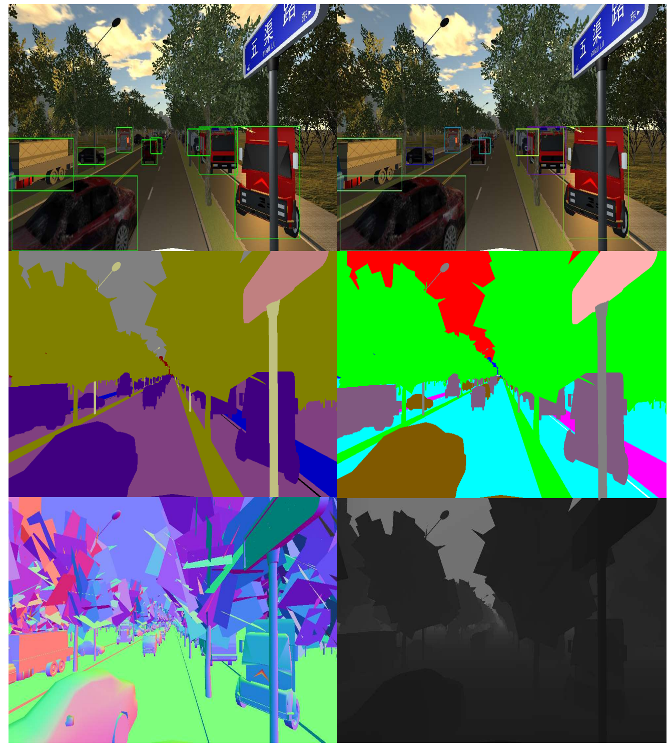

Fig. 4. Examples of ground-truth annotations generated automatically by Unity3D. Top: object detection (left) and object tracking (right). Middle: semantic segmentation (left) and pixel-level instance segmentation (right). Bottom: optical flow (left) and depth (right).

IV. Experiments

Based on the method proposed above, we construct a artificial track scene. We collect data called "The ParallelEye-CS Dataset" from the artificial track by configuring camera parameters and changing environmental conditions. The dataset is used as visual perception of city roads, which assist intelligent vehicles in training object detection, semantic segmentation, object tracking algorithms. ParallelEye-CS includes ParallelEye-CS_01, ParallelEye-CS_02 and ParallelEye-CS_03, in the simulation of testing tasks of self-driving vehicles. The construction of ParallelEye-CS also proves the repeatability of artificial

Fig. 5. Illustration of the diversity of artificial scenes. Virtual images with illumination at 6:00 am (upper left) and 12:00 pm (lower right) in a sunny day. Virtual images with weather of fog (lower left) and rain (upper right).

Fig. 6. Continuous images captured by the intelligent vehicle in ParallelEye-CS_01. Some examples of virtual testing tasks including intelligent vehicle starts (upper left) and avoids collision (lower). Best viewed with zooming.

scene collection, configurability of camera parameters and the variability of environmental conditions.

A. ParallelEye-CS_01

In this experiment, ParallelEye-CS_01 mainly simulates the startup of a unmanned vehicle, straight line driving and avoiding obstacles in Yunshen Road. The camera height is set to be 2 meters above the ground. There are 118 vehicles, 116 of them parked by the roadside (17 buses, 18 trucks and 83 cars) and a working vehicle. The first part of the ParallelEye-CS dataset is shown in Fig. 6.

B. ParallelEye-CS_02

In this experiment, ParallelEye-CS_02 mainly imitates the detection and avoidance of trucks and driving U-turn in Wuqu Road. The camera height is set to be 2 meters above the ground. There are 71 vehicles (24 buses, 27 trucks and 20 cars) and a moving truck. The second part of the ParallelEye-CS dataset is shown in Fig. 7.

C. ParallelEye-CS_03

In this experiment, ParallelEye-CS_03 mainly simulates the encounter with special road conditions (water, fog, rough terrain and tunnel) and avoidance of pedestrians. This road is designed as a reference for real testing tasks. As shown in table I, the advantage of ParallelEye-CS over other artificial

Fig. 7. Continuous images captured by the intelligent vehicle in ParallelEye-CS_02. Some examples of virtual testing tasks including intelligent vehicle avoids collision with truck (upper middle and right) and u turn (lower). Best viewed with zooming.

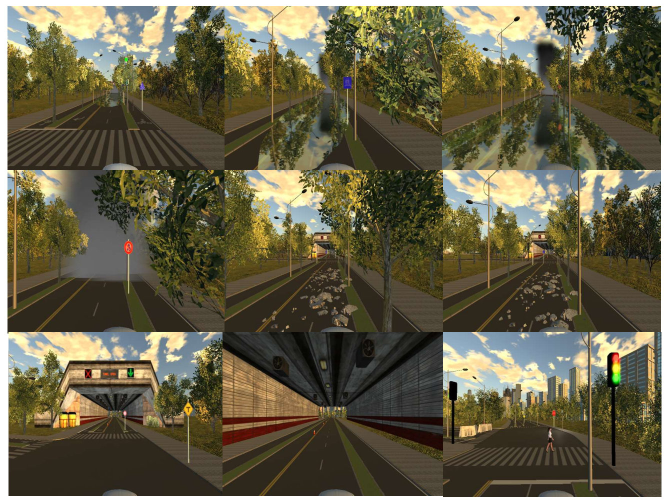

Fig. 8. Continuous images captured by the intelligent vehicle in ParallelEye-CS_03.Some examples of virtual testing task including intelligent vehicle passes through flood (upper middle and right), smoke, cross-country (middle right), tunnel (lower left). Best viewed with zooming.

scenes like KITTI and SYNTHIA is that it can not only construct artificial scenes with the guidance of real scenes, but also evaluate the visual intelligence of intelligent vehicles.The third part of the ParallelEye-CS dataset is shown in Fig. 8.

In the experiments, with image resolution of 1000*500 pixels for ParallelEye-CS, the pipeline for artificial scene construction and ground truth generation runs at 7-10 fps (frames per second) on a workstation computer.

V. Concluding Remarks

In this article, we propose ParallelEye-CS dataset with various label information. The artificial scene matches the real world well in terms of fidelity and geographic information. The advantages of artificial scene are: (1) artificial scenes can guide the build and setting of real scenes, which used to evaluate the visual intelligence of intelligent vehicles. (2) label information can be generated automatically in artificial scenes, including semantic/instance segmentation, object bounding box, object tracking, normal, and depth. (3) images

TABLE I

A comparison of ParallelEye-CS and other datasets

Real scenes	CHANGSHU	KITTI	N/A
Virtual scene	ParallelEye-CS	Virtual KITTI	SYNTHIA
Having real scene guidance	yes	yes	no
Real scene influencing virtual scene	yes	no	no
Having real and virtual interaction	yes	no	no

under different conditions can be generated repeatedly by varying the camera parameters and environmental conditions.

In future work, we will study the use of GANs [26-30] to generate realistic artificial images and conduct domain adaptation. We also plan to investigate the methods for constructing artificial scenes [31-35] in a more intelligent, effective, and efficient way.

References

[1] J. T. Schwartz and M. Sharir, “A survey of motion planning and related geometric algorithms,” *Artificial Intelligence*, vol. 37, no. 1-3, 1988, pp. 157-169.

[2] Y. Zhou and O. Tuzel, “Voxelnet: End-to-end learning for point cloud based 3d object detection,” *arXiv preprint arXiv:1711.06396*. 2017.

[3] M. Bojarski, D. Del Testa, D. Dworakowski, “End to end learning for self-driving carsv,” *arXiv preprint arXiv:1604.07316*, 2016.

[4] E. Santana and G. Hotz, “Learning a driving simulator,” *arXiv preprint arXiv:1608.01230*, 2016.

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, 2013, pp. 1234-1237.

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, “The cityscapes dataset for semantic urban scene understanding,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213-3223.

[7] D. A. Pomerleau, “Alvinn: An autonomous land vehicle in a neural network,” *Proceedings of Advances in Neural Information Processing Systems*, 1989, pp. 305-313.

[8] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, “Virtual worlds as proxy for multi-object tracking analysis,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3234-3243.

[9] T. P. Lillicrap, J. J. Hunt, A. Pritzel, “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971* 2015.

[10] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, “Deepdriving: Learning affordance for direct perception in autonomous driving,” *Proceedings of the IEEE Conference on International Conference on Computer Vision*, 2015, pp. 2722-2730.

[11] J. Koutnk, G. Cuccu, J. Schmidhuber, and F. Gomez, “Evolving large-scale neural networks for vision-based reinforcement learning,” *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, ACM, 2013, pp. 1061-1068.

[12] B. Wymann, E. Espie, C. Guionneau, C. Dimitrakakis, R. Coulom, “Torcs, the open racing car simulator,” *Software available at http://torcs. sourceforge. net*, vol. 4, , 2000.

[13] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, “Playing for data: Ground truth from computer games,” *Proceedings of the European Conference on Computer Vision*, 2016, pp.102-118.

[14] S. R. Richter, Z. Hayder, and V. Koltun, “Playing for benchmarks,” *Proceedings of the IEEE Conference on International Conference on Computer Vision*, 2017, pp.322-339.

[15] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” *arXiv preprint arXiv:1711.03938*, 2017.

[16] D. Biedermann, M. Ochs, and R. Mester, “Evaluating visual adas components on the congrats dataset” *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2016, pp. 986-991.

[17] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, “The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3234-3243.

[18] J. Skinner, S. Garg, N. Sunderhauf, P. Corke, “High-fidelity simulation for evaluating robotic vision performance,” *Proceedings of IEEE Conference on Intelligent Robots and Systems*, 2016, pp. 2737-2744.

[19] Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza, “Benefit of large field-of-view cameras for visual odometry,” *Proceedings of IEEE Conference on Robotics and Automation*, 2016, pp. 801-808.

[20] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, “High-resolution image synthesis and semantic manipulation with conditional gans,” *arXiv preprint arXiv:1711.11585*, 2017.

[21] K. Wang, C. Gou, N. Zheng, J. M. Rehg, and F.-Y. Wang, “Parallel vision for perception and understanding of complex scenes: Methods, framework, and perspectives,” *Artificial Intelligence Review*, vol. 48, no. 3, 2017, pp. 299-329.

[22] K. Wang, C. Gou, and F.-Y. Wang, Parallel vision: An ACP-based approach to intelligent vision computing,” *Acta Automatica Sinica*, vol. 30, no. 7, 2017, pp. 577-587.

[23] K. Wang, Y. Lu, Y. Wang, Z. Xiong, and F.-Y. Wang, “Parallel imaging: a new theoretical framework for image generation,” *Pattern Recognition and Artificial Intelligence*, vol. 30, no. 7, 2017, pp. 577-587.

[24] F.-Y. Wang, “Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, 2010, pp. 630-638.

[25] F.-Y. Wang, “Parallel control: a method for data-driven and computational control,” *Acta Automatica Sinica*, vol. 39, no. 4, 2013, pp. 293-302.

[26] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, “Generative adversarial networks: introduction and outlook,” *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, 2017, pp. 588-598.

[27] Li X, Wang K, Tian Y, et al., “The ParallelEye dataset: Constructing large-scale artificial scenes for traffic vision research,” *arXiv preprint arXiv:1712.08394*, 2017.

[28] Tian Y, Li X, Wang K, et al., “Training and Testing Object Detectors With Virtual Images,” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 2, 2018, pp. 539-546.

[29] Wenbo Zheng, Kunfeng Wang, and Fei-Yue Wang, “Background Subtraction Algorithm Based on Bayesian Generative Adversarial Networks,” *Acta Automatica Sinica*, in press.

[30] Li Li, Yi-Lun Lin, Nan-Ning Zheng, Fei-Yue Wang, Yuehu Liu, Dongpu Cao, Kunfeng Wang, and Wu-Ling Huang, “Artificial intelligence test: a case study of intelligent vehicles,” *Artificial Intelligence Review*, April 2018, online available.

[31] Kunfeng Wang, Chao Gou, and Fei-Yue Wang, “M4CD: A Robust Change Detection Method for Intelligent Visual Surveillance,” *IEEE Access*, vol. 6, 2018, pp. 15505-15520.

[32] Yating Liu, Kunfeng Wang, and Fei-Yue Wang, “Tracklet Association-based Visual Object Tracking: The State of the Art and Beyond,” *Acta Automatica Sinica*, vol. 43, no. 11, 2017, pp. 1869-1885.

[33] Wenwen Zhang, Kunfeng Wang, Hua Qu, Jihong Zhao, and Fei-Yue Wang, “Scene-Specific Pedestrian Detection Based on Parallel Vision,” *arXiv preprint arXiv:1712.08745*, 2017

[34] Hui Zhang, Kunfeng Wang, and Fei-Yue Wang, “Advances and Perspectives on Applications of Deep Learning in Visual Object Detection,” *Acta Automatica Sinica*, vol. 43 no.11, 2017, pp. 1289-1305.

[35] Chao Gou, Yue Wu, Kang Wang, Kunfeng Wang, Fei-Yue Wang, Qiang Ji, “A Joint Cascaded Framework for Simultaneous Eye Detection and Eye State Estimation,” *Pattern Recognition*, vol. 67, 2017, pp. 23-31.