Learning Pothole Detection in Virtual Environment

Jung-Cheng Tsai
Department of Electronic Engineering
National Taipei University of Technology
Taipei, Taiwan
t107368009@ntut.org.tw

Kuan-Ting Lai
Department of Electronic Engineering
National Taipei University of Technology
Taipei, Taiwan
ktlai@ntut.edu.tw

Tzi-Chun Dai
Department of Electronic Engineering
National Taipei University of Technology
Taipei, Taiwan
t107360723@ntut.org.tw

Jun-Jia Su
Department of Electronic Engineering
National Taipei University of Technology
Taipei, Taiwan
t108368012@ntut.org.tw

Chao-Yu Siao
Department of Electronic Engineering
National Taipei University of Technology
Taipei, Taiwan
t108368002@ntut.org.tw

Yung-Chin Hsu
Department of Electronic Engineering
National Taipei University of Technology
Taipei, Taiwan
t108368014@ntut.org.tw

***Abstract*—Pothole detection is an important function in autonomous vehicles, which can help vehicles to avoid dangerous traps on roads, or change suspension to make passengers more comfortable. However, it is challenging to train a high quality pothole detector, mainly due to the difficulty of collecting training data. Sending automobiles with cameras to record videos of potholes is time consuming, expensive, and may lead to unexpected accidents.**

To address this issue, we leverage the recent emerging virtual-to-real learning and use latest virtual reality technology to train a pothole detector. We develop a pothole generation system that can generate holes with various shapes, sizes, and depths. The virtual pothole images are added to the training dataset, and the detector performance is evaluated on real data. Experiment results show that virtual pothole images can successfully increase the overall detection accuracy and enable users to train detectors with less real data.

Keywords*—*pothole detection, virtual-to-real learning, deep learning

I. INTRODUCTION

Autonomous vehicle is one of the most important applications in the near future. The key technologies in modern autonomous navigation system are computer vision and deep learning, which enable onboard computers to recognize numerous objects like roads, pedestrians, traffic lights, etc., and perform corresponding actions. To achieve best accuracy, we need to train specific neural-network detectors for different tasks, e.g. object detector, traffic light detector and classifier, scene segmentation, to name a few. Modern autonomous system usually contains dozens of neural networks that work concurrently.

Among many subsystems, pothole detection is an important function that have not been fully studied. Potholes are caves or hollows in a road surface, usually asphalt pavement, which are caused by the water in the underlying soil structure and traffic passing over the affected area [3]. The process of pothole formation is as follows. Water first weakens the underlying soil, and then traffic of vehicles gradually fatigues the poorly supported asphalt surface. After a period of time, the ongoing traffic action damages both asphalt and the underlying soil and creates a hole in the pavement. Small potholes cause vibration of vehicles and uncomfortable ride experience, while large potholes can lead to serious accidents. If potholes can be detected accurately, the autonomous automobiles can change the

Fig. 1. Our virtual environment for learning pothole detection, which is based on AirSim [1] and VIVID [2].

This work was supported by the National Science Council of Taiwan under contract MOST 109-2218-E-035-007

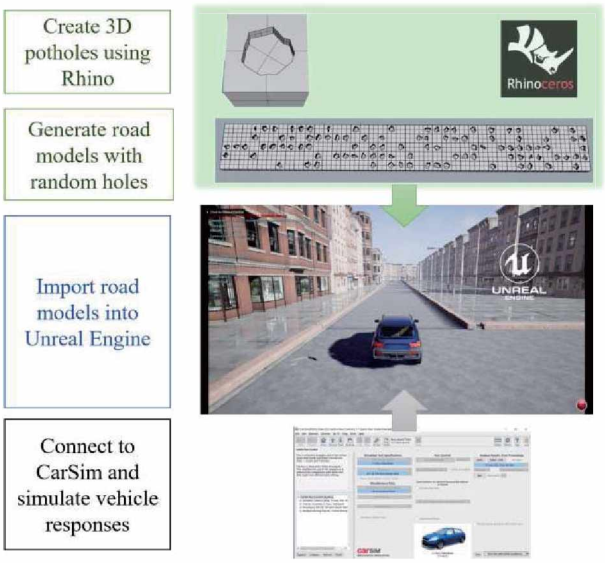

Fig. 2. Flowchart of our system. Rhino 3D is used to create 3D road models with various potholes. The models are then imported into Unreal engine, which supports plugins such as CarSim for real vehcile simulation.

suspension to absorb the vibration, or even modify the route to avoid potential accidents. We expect that pothole detection will be an indispensable part of next-generation vehicles.

However, there are not much research about pothole detection so far. One of the reasons is that collecting pothole images is expensive and time-consuming. Moreover, it is difficult to record videos under some climate condition, such as raining and snowing. In this paper, we propose to train pothole detection using virtual environment. Fig. 1 shows one of the scenes in our simulation environment. Our system is based on the open-source project AirSim [1] and VIVID [2]. The flowchart is illustrated in Fig. 2. In order to generate potholes with depths, we use Rhino 3D [4] to create a road model containing 3D holes of various shapes and depths. The road model is then imported into Unreal engine [5] with AirSim and VIVID plugin, which can create random combination of the roads with potholes under different weather conditions. We train pothole detectors with mix of real and virtual pothole images, and evaluate the performance on real photos. Experiment results demonstrate that our environment can help to increase the accuracy of pothole detection in real world.

II. Related Work

A. *Virtual-to-real Learning*

Training models in virtual environment and deploying in real world is an emerging technique, which is called virtual-to-real learning. Specifically, virtual-to-real learning can be considered as one kind of the model-based reinforcement learning. DeepMind first introduce to combine reinforcement learning with deep learning and train agents to play Atari games. The proposed Deep Q-Networks (DQN) has achieved human-level performance and inspired the frenzy of deep reinforcement learning [6]. After that, Pan et al. [7] proposed to train autonomous vehicle in Unity, claimed that virtual and real environments are nearly identical after semantic segmentation. Sadeghi and Levine [8] proposed the domain randomization, which provides agents numerous randomized environments with various colors and textures of objects. The goal is to make AI agents think that the real world is just one of the randomized environments, and focus on learning the key features. The authors successfully train an autonomous drone in CAD 3D environment and deployed in real world. Other popular learning environments include OpenAI Gym [9], Project Malmo [10], DeepDrive [11] and Unity AI [12], to name a few. Recently many advanced simulation environments have been proposed. Proposed to use AirSim to detect poachers in Africa [13]. There are several new released environment for Drone Racing [14], Lidar simulation [15], or Robot navigation without map [16].

B. *Object Detection*

Object detection is one of the most import research topic in computer vision. During the past decades, many powerful algorithms were proposed, such as R-CNN series [17, 18, 19], YOLO series [20, 21, 22], and SSD [23]. The latest detector is EfficientDet [24], which train using Neural-network Architecture Search (NAS) technique. Among all detector technologies, YOLO is currently the most popular one due to its simplicity and speed. We choose YOLO v3 as the backbone of our pothole detection, and train the detector with mix of real and virtual photos.

III. Proposed Method

In this section, we will present the implementation details of our simulation system. First, let us review the basic structure of common roads. The cross-sectional view of an asphalt road is shown in Fig. 3. Asphalt pavement consists of ten layers. The main load bearing is the dense-graded asphalt concrete layer, which is also the main layer where the road surface collapses to form potholes. Fig. 7 shows the photos of different pothole types. The upper-left one is pavement rutting; the upper-right one is aggregate losses; the lower-left one is fatigue crack, and the lower-right one is pothole repair. If the autonomous vehicle can identify the type of a pothole, then it can choose different strategies to enhance safety. For example, it can avoid the large potholes while changing suspension for small ones. Creating potholes in a virtual environment may seem like a trivial task, but it is actually difficult. The reason is that most VR development tools start with a base plane first, then add landscape and other objects on top of the plane.

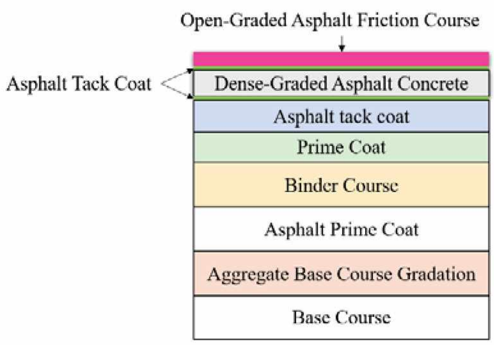

Fig. 3. Cross sections of asphalt road.

Identify applicable funding agency here. If none, delete this text box.

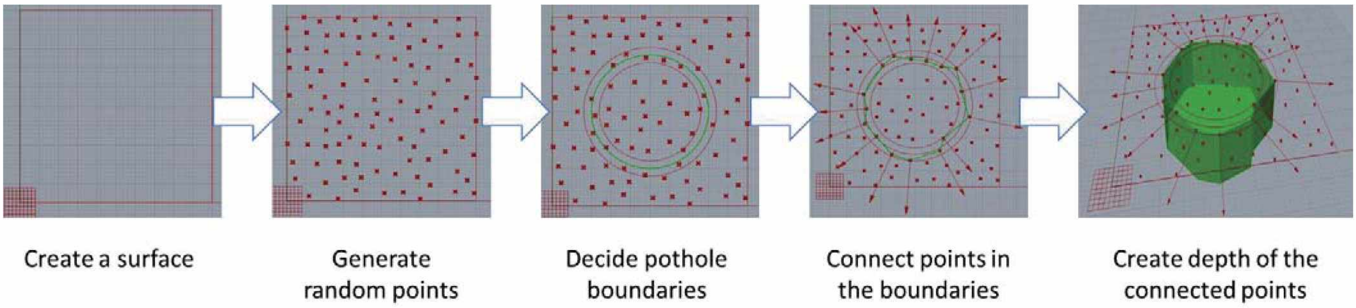

Fig. 4. Process of creating a 3D pothole model using Rhinoceros.

As a result, it is easy to add bumps, but hard to make holes on the base plane. Most games avoid this issue by using pothole textures to pretend holes. However, a texture does not have depth information, which is important for advanced pothole detection. Therefore, we utilize professional 3D modeling software, Rhinoceros 3D, to create pothole models. The process is shown in Fig. 4. The first step is to create a surface region. The random points are then generated inside the region. Next, we generate two pothole circle boundaries, and connect the random points inside the circle area. Finally, the depth of the hole is created based on the connected points. The visual script used to generate Rhino 3D is shown in Fig. 5.

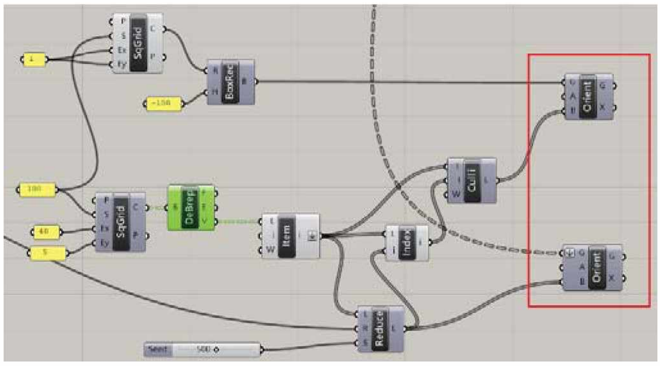

Fig. 5. Visual script used to generate the potholes in Rhino 3D.

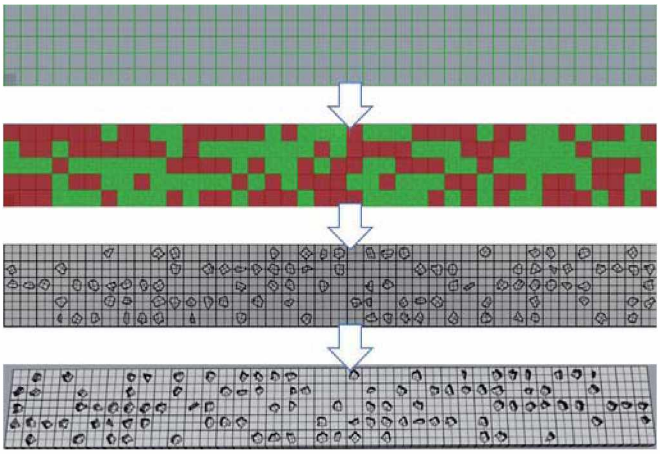

Fig. 6. Process of generating the 3D road model. In this case, we create a road grid with 5×200 units, and randomly select the pothole locations.

Fig. 7. Different types of potholes. The upper-left one is pavement rutting; the upper-right one is aggregate losses; the lower-left one is fatigue crack, and the lower-right one is pothole repair. Photo courtesy of [26].

Once the 3D pothole models are ready, the next step is to generate the road model. The process is shown in Fig. 6. We first create a road grid, which is 5×200 in this case, and randomly select pothole locations. Finally, the 3D pothole models are put in the locations, and the 3D road model can be imported into Unreal engine. Fig. 8 demonstrates the vanilla 3D road model in Unreal. To make it photo-realistic, we add texture and other materials, as shown in the first page of our paper (Fig. 1).

After importing the potholes and roads, we need to simulate the vehicles. There are several Unreal plugins that can perform this task. The most popular one is AirSim, which is open-source software developed by Microsoft. Originally created for drone simulation, AirSim has been widely adopted and support car simulation now. Additionally, AirSim provides weather condition simulation including raining and snowing. Therefore, we also create a pothole model with water for raining condition.

However, AirSim uses the default car model of Unreal, which was developed for games and is too simple for professional simulation. The alternative solution is CarSim [25], which is a commercial software that can simulate a large number of vehicle parameters. We have connected our simulation system with CarSim, as shown in Fig. 9.

Fig. 8. Our vanilla 3D road model in Unreal.

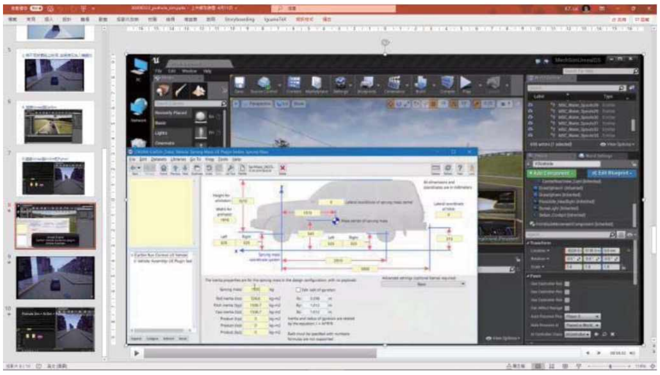

Fig. 9. Connecting CarSim to our pothole simulation environment.

IV. EXPERIMENTS

We recorded video on roads in Taiwan and created a pothole dataset for evaluating our method. Several selected photos are shown in Fig. 10. There are around 17,000 labelled potholes in the dataset. For training data, we select 15,084 real images, and generate 5,417 virtual potholes with water and 5,607 potholes without water. In term of testing, we evaluate the detector performance on 1,745 real images. TABLE I. lists the number of training and test images.

TABLE I. TRAINING AND TEST DATA

Image Type	Number of Images
Training Data	
Real photos	15084
Virtual potholes	5607
Virtual potholes with water	5417
Test Data	
Real Photos	1745

Fig. 10. Several selected photos from the Taiwanese pothole dataset.

We use the mean Average Precision (mAP) as our metric. The detection is considered successful if the Intersection over Union (IoU) is larger than 0.5. YOLO v3 is employed as our neural network model.

For training pothole detector, we combine real photos with virtual images and observe the performance change. Specifically, we gradually increase the number of real images in the training dataset to see the effects of virtual images. The results are shown in TABLE II. As can be seen, adding virtual images can improve the recognition rate, and reduce the requirement of real images.

One thing to note is that using both virtual potholes with and without water as training data may lead to lower accuracy. We conjecture that pothole with water should be considered as a new class for the model to learn how to classify it precisely. We leave it as future work.

TABLE II. EXPERIMENT RESULTS OF TRAINING POTHOLE DETECTORS

Number of Real Images for Training	**Mean Average Precision (mAP)**		
	Real Images only	***Real + Virtual Images with water (5417)***	***Real + Virtual Images without water (5607)***
1571	**77.12%**	75.53%	74.29%
3414	**86.38%**	84.99%	85.33%
4712	**91.95%**	88.83%	90.18%
6282	90.46%	**92.83%**	91.27%
7852	**91.83%**	91.56%	91.31%
9423	**91.93%**	90.57%	91.12%
10993	92.61%	**93.68%**	91.29%
12564	92.74%	93.43%	**93.70%**
14134	92.79%	92.24%	**92.93%**
15705	92.65%	93.45%	**93.62%**

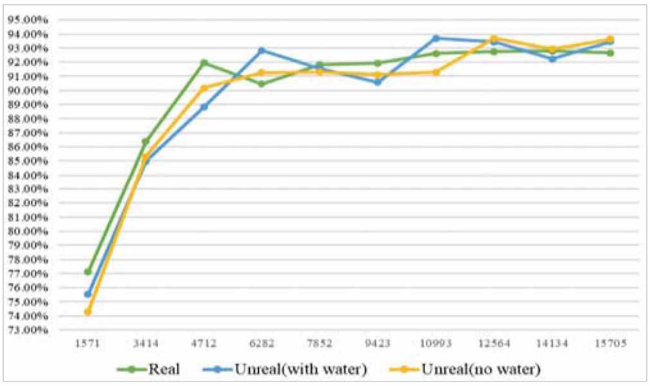

Fig. 11. Comparison of pothole detection accuracy using real photos only, real + Unreal-generated images with water, and real + Unreal-generated images without water.

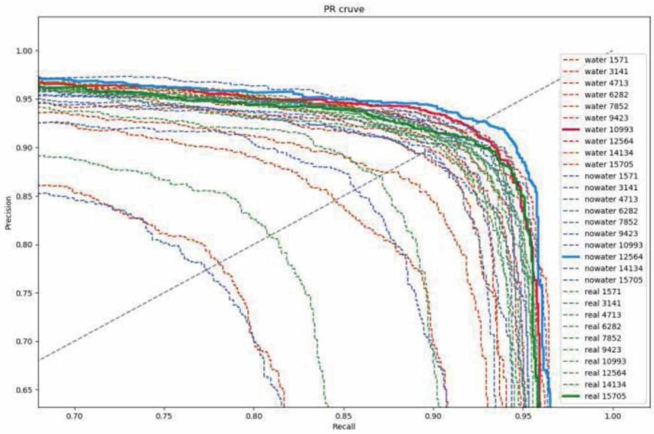

Fig. 12. Precison-recall curves of different real-virtual image combination. The solid line represents the best performance in each group.

V. CONCLUSION

In this paper, we developed a new virtual environment for training pothole detection. Our system integrates many modern VR and simulation techniques, including 3D modeling, VR simulation, car simulation, and deep learning interface. We conducted many experiments on real pothole dataset, and demonstrated that virtual images can indeed increase the accuracy of a real pothole detector. For future work, we will try deep reinforcement learning with CarSim, and train AI agent to learn to automatically adjust car suspension system under different road and weather conditions.

REFERENCES

[1] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in Field and service robotics. Springer, 2018, pp. 621–635.

[2] K.-T. Lai, C.-C. Lin, C.-Y. Kang, M.-E. Liao, and M.-S. Chen, "Vivid: Virtual environment for visual deep learning," in Proceedings of the 26th ACM international conference on Multimedia, 2018, pp. 1356–1359.

[3] Wikipedia, "Pothole," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Pothole

[4] Robert McNeel & Associates, "Rhinoceros," 2020. [Online]. Available: https://www.rhino3d.com

[5] Epic Games, "Unreal engine," 2020. [Online]. Available: http://www.unrealengine.com

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.

[7] X. Pan, Y. You, Z. Wang, and C. Lu, "Virtual to real reinforcement learning for autonomous driving," arXiv preprint arXiv:1704.03952, 2017.

[8] F. Sadeghi and S. Levine, "(CAD)2RL: Real single-image flight without a single real image," arXiv preprint arXiv:1611.04201, 2016.

[9] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI GYM," arXiv preprint arXiv:1606.01540, 2016.

[10] Microsoft, "Projectmalmo," 2015 .[Online]. Available: http://www.microsoft.com/en-us/research/project/project-malmo/

[11] DeepDrive, "Deepdrive," 2018. [Online]. Available: http://deepdrive.io

[12] Unity, "Unity machine learning," 2017. [Online]. Available: http://unity3d.com/machine-learning

[13] E. Bondi, D. Dey, A. Kapoor, J. Piavis, S. Shah, F. Fang, B. Dilkina, R. Hannaford, A. Iyer, L. Joppa et al., "Airsim-w: A simulation environment for wildlife conservation with uavs," in Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies, 2018, pp. 1–12.

[14] R. Madaan, N. Gyde, S. Vemprala, M. Brown, K. Nagami, T. Taubner, E. Cristofalo, D. Scaramuzza, M. Schwager, and A. Kapoor, "Airsim drone racing lab," arXiv preprint arXiv:2003.05654, 2020.

[15] S. Manivasagam, S. Wang, K. Wong, W. Zeng, M. Sazanovich, S. Tan, B. Yang, W.-C. Ma, and R. Urtasun, "Lidarsim: Realistic lidar simulation by leveraging the real world," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11 167–11 176.

[16] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 31–36.

[17] R. Girshick, "Fast R-CNN," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91–99.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[21] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in European conference on computer vision. Springer, 2016, pp. 21–37.

[23] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[24] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10 781–10 790.

[25] Mechanical Simulation, "CarSim," 2020. [Online]. Available: https://www.carsim.com

[26] 陳建旭、劉韋廷、廖敏志、王慶雄、蔡益智、林和志, "分析鋪面坑洞產生原因與建議維護方法," 臺灣公路工程 第 37 卷第 10 期, 民國 100 年 10 月