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

Evaluation of Synthetic Video Data in Machine Learning Approaches for Parking Space Classification

Daniela Horn[1], Sebastian Houben[1]

Abstract— **Most modern computer vision techniques rely on large amounts of meticulously annotated data for training and evaluation. In close-to-market development, this demand is even higher since numerous common and—more important—less common situations have to be tested and must hence be covered datawise. However, gathering the necessary amount of data ready-labeled for the task at hand is a challenge of its own. Depending on the complexity of the objective and the chosen approach, the required amount of data can be vast. At the same time, the effort to capture all possible cases of a given problem grows with their variability. This makes recording new video data unfeasible, even impossible at times. In this work, we regard parking space classification as an exemplary application to target the imbalance of cost and benefit w.r.t. image data creation for machine learning approaches. We rely on a fully-fledged park deck simulation created with Unreal Engine 4 for data creation and replace all conventionally recorded and hand-labeled training data by automatically-annotated synthetic video data. We train several of-the-shelf classifiers with a common choice of feature inputs on synthetic images only and evaluate them on two real-world sequences of different outdoor car parks. We reach a classification performance that matches our previous work on this task in which all classifiers were developed solely with real-life video data.**

I. INTRODUCTION

In the age of deep learning, we frequently strive for new challenges in machine learning and computer vision, and every breakthrough is valued as a huge success. While theories and machine learning approaches are constantly evolving, new methods require more and more highly accurately annotated data for training and evaluation [1], [2]. The quantity of generated data, however, falls behind. This is particularly true when approaching new fields where application-specific data is needed or various conditions have to be covered. In fact, qualitatively good data in the necessary amount is only available for a small number of machine learning tasks. In the area of intelligent vehicles, these mainly include different detection tasks like pedestrian, traffic sign, and vehicle detection [3], [4], [5]. This circumstance limits the potential of scientific progress to very few topics. It is a stroke of luck to find readily annotated data in the necessary amount with the required labels, so creating new data sets for a certain purpose is an important, yet oftentimes undervalued, commitment to the computer vision and machine learning community. *E.g.*, in the case of parking space classification, outside locations with uncertain weather conditions, lack of electricity supply on site, and possible violations of the Data Protection Act are some of the challenges to face when it comes to video data creation. Once the footage has been recorded, days or even weeks of manual labeling follow, making the process of ground truth creation lengthy and excessively unattractive. An automated way of gathering both image data and specific ground-truth information seems a precious asset for this scenario, but could also be valuable for other computer vision and image-based machine learning challenges.

(a) Sunshine Scenario (b) Dark Clouds Scenario

(c) Fog Scenario

Fig. 1: Various weather conditions within the simulated environment. Parking space snippets of these sequences were used for classifier training.

In this paper, we deploy the previously presented simulated car park environment [6] to investigate a new method of training data creation for machine learning and image processing algorithms. This is an extension to our recently established video-based parking space classification [7], [8] and on-site routing [9]. Here, we replace earlier used training data for parking space classification completely with synthetic video data gathered from the simulated environment. Input data was taken from three different weather conditions (sunny, dark clouds and fog), of which we recorded video sequences beforehand (*cf.* Fig. 1).

Following up on our earlier work [8], we train a number of *k-Nearest-Neighbor* (kNN) classifiers and *Support Vector*

[1] University of Bochum, Institute for Neural Computation
daniela.horn@ini.rub.de

Machines (SVM) with *Difference-of-Gaussian* (DoG) features of different filter sizes and evaluate their performance on two real-world video sequences of different parking areas. Other than [8] we have decided to omit color feature inputs, as the final system for on-site parking guidance is intended to run on preinstalled surveillance cameras with grayscale input. Our main contribution is the demonstration and evaluation of the transferability from purely synthetically generated images to the task of parking space classification with real-world camera data.

The paper is organized as follows: Section II takes a closer look at related works in the field of simulation within the context of intelligent transportation. In Section III the generation of synthetic image data for training purposes is described in more detail. Section IV focuses on the experiments which were conducted to evaluate and validate our approach. A conclusion and outlook in Section V round off the paper.

II. RELATED WORK

Simulations are an essential tool in many industrial and scientific fields of research. In the context of intelligent transportation systems they are frequently used for optimization and evaluation purposes, or to visualize results. Commonly used tools for traffic simulation are *VISSIM*[2], *AnyLogic*[3], *MATSim*[4] and *SUMO*[5].

The PTV Group, developer and owner of VISSIM, is one of the most popular suppliers of simulation software for traffic analysis and visualization of given data. Using VISSIM for microscopic traffic simulation in combination with other complementary products like VISUM for macroscopic transportation planning and VISTRO for traffic impact analysis & signal optimization, PTV offers a solution for many traffic simulation demands. The PTV Group themselves use VISSIM to visualize simulations of project conducted for a number of clients. Furthermore, the PTV products are used worldwide to support researchers and companies in various traffic-related simulation tasks [10].

For the purpose of developing and testing parking guidance systems, Yuan and Liu [11] built a simulation framework including car following and vehicle generation models and employ VISSIM for the dynamic simulation of traffic. The main focus of this project, as with most simulation-related works, lies in the analysis and simple visualization of predefined data within the simulated system as proof of concept.

In order to maximize the intended output and use simulations as flexible as possible, many research groups opt for building their own simulation frameworks. Here, game engines are becoming a more and more valued tool for scientific purposes. Konrad et al. [12] use a game engine for the reproduction of traffic-related scenarios from previously recorded data. They combine information extracted from Google StreetView, OpenStreetMaps and SketchUp (also by Google) to collect map data and other static scene characteristics and conjoin the resulting information with such recorded from vehicle sensors to rebuild the scene in CityEngine. While this first step is the virtual visualization of an occurred real-world scenario, their project goal is the analysis of information for the evaluation of collision avoidance algorithms. With this purpose in mind they attempt to show the usability of game engines in a scientific context along the way.

Modeling an artificial transportation system with a Delta3D-based platform, Miao et al. [13] use the advantages of this engine's gaming background by integrating human action and interaction rather than using precalculated data. They use native concepts of game engines by adopting a multiplayer approach to create new data with a simulated environment in which the human behavior component is still intact. Their main focus is the interaction with the real world, *i.e.*, the reception of real-time information from control signals and direct multi-user interaction.

Signal control scenarios are modeled by Wang and Abbas [14] as well, however in a primarily educational context. Their simulation is used to help engineering students understand complex concepts behind traffic signal control. A study about the learning effects show the usability of simulations for the analysis and comprehension of complex data.

All of these project use simulations of visually highly simplified manners to depict or create data. Their focuses are diverse but never related to realism in visualization. The approach presented in this paper is opposed to that. A simplification of object's appearance is not desirable; instead a photo-realistic environment with detailed cars and natural behavior is required for the purpose of image training data creation for machine learning algorithms.

III. SIMULATED DATA GENERATION

Using the simulated environment described in [6], a number of video sequences with different weather and lighting conditions were recorded. From these sequences we selected training data from sunny, cloudy and foggy weather conditions to use as training data. The extraction of the final images used for training is outlined in Section III-A. While recording the relevant sequences, we automatically extracted the ground-truth data needed for training. The procedure is described in Section III-B.

A. Extraction of Training Data

Synthetic data forms the sole input data for all classifiers. Fig. 2 shows typical snippets of empty and occupied parking spaces, which were used for training. The training data set covers a total of 24.498 samples, 11.760 of them depicting empty parking spaces, 12.738 snippets showing occupied ones.

Before the training images can be extracted from the video sequences, a couple of preprocessing steps have to be executed, similar to real-world videos. Using the methods

[2] vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/

[3] www.anylogic.com/

[4] www.matsim.org/

[5] www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883

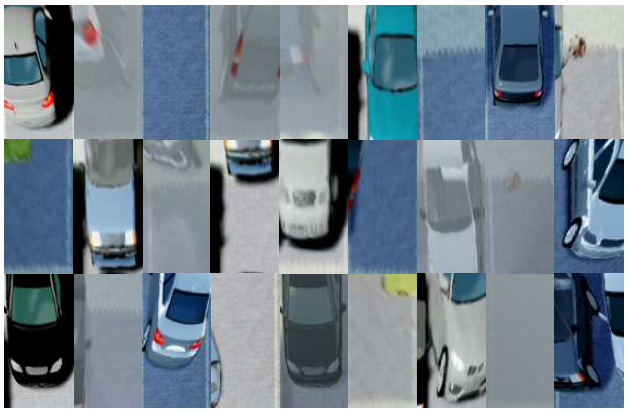


Fig. 2: Examples of training data extracted from video sequences of the simulated environment. Weather conditions include sunny, cloudy and foggy samples.

described in [8], first an intrinsic calibration is applied to correct the camera distortion of the recording camera, before the parking spaces are labeled once by hand in the then rectified images. For the latter, it is sufficient to mark a complete row with four corner points and divide the chosen region by the number of parking spaces in it. This approach is relatively time effective while sufficiently exact at the same time.

Although the preparation of the simulated training data does not differ from real-world footage, working with simulated environments has a huge advantage, namely the automatic extraction of ground-truth data. In the given scenario, the required ground truth comprises of a unique parking space ID and a binary classification of the occupancy status in each frame of a given video sequence. While we used this advantage for the evaluation of classifiers in [6], the extracted occupancy information now aids in the automatic sorting of data snippets into two classes (available/occupied) for training. The image regions marking each single parking space are cut out automatically and matched with their respective labels. The resulting images can now be used as input data for training.

B. Automatic Generation of Ground-Truth Data

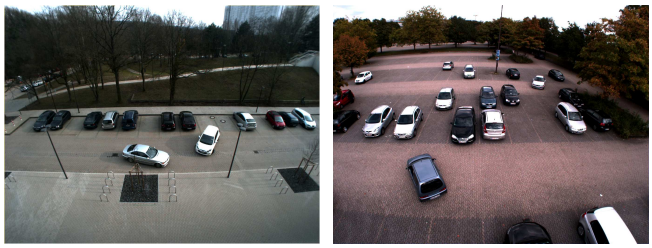
When it comes to the training and evaluation of image processing algorithms, annotated data is a key element. However, the process of annotating video material is time-consuming and hardly feasible. For parking space classification, *e.g.*, sensible annotations might include the location and current occupancy status of each parking space that should be observed. This information, of course, has to be gathered for every single frame in order to use the data for training or further assessment purposes. There exist public data sets for the given task on the Internet, but using them naturally comprises a number of potential problems as well, as the given image data might only partially fit the task or the annotations are either not available or marking the wrong aspect.

In the simulated environment, it is possible to gather these and other ground-truth data automatically at will. So, in combination with the image data, ground-truth data was extracted from the car park environment. In the given case we are interested in each parking space's occupancy status, therefore each parking space object in the environment was equipped with a trigger volume, which registers changes of occupancy status and enters the information to a log file together with a system time stamp. This change detection operated both from "available" to "occupied" and vice versa.

Every video sequence that is recorded out of the simulated environment also automatically creates a sequence log file, stating the system's time stamp for every single recorded frame. The information of both parking space and sequence log files is afterwards automatically matched to gain ground-truth data with a framewise precision. The resulting information is then used to cut out the image snippets required for training and sort them into the two classes for the decision process without further ado.

IV. EXPERIMENTS

In this section, we evaluate our approach of replacing input images for classifier training completely with synthetic data. As described in Section III, we use training samples from three different weather scenarios created with the simulated car park environment presented in [6]. In Section IV-A we introduce the two sequences which were used for validation. The conducted classifier training is explained in more detail in Section IV-B. The results of our experiments are presented and discussed in Section IV-C.



(a) Sequence A (b) Sequence B

Fig. 3: Video sequences used for the evaluation of the different classifiers.

A. Validation Sequences

Two real-world video sequences were chosen in order to evaluate the classifiers (*cf.* Fig. 3). Sequence A shows a single parking row at the side of a street. The video camera was placed in a building opposite the parking area. The sequence was recorded with 5 fps and is 11:18 min long. 15 parking spaces are evaluated. The difficulty with this sequence is the mirroring effects of the closed window through which the scene was recorded.

Sequence B overlooks a parking area with several rows. The camera was placed on a tripod about 11 m off the ground to cover a maximum of parking spaces at the same time. This

sequence was recorded with 5 fps, as well, and lasts 3:25 min. For evaluation, three rows with a total of 36 parking spaces have been chosen. The rows are enumerated from row 1 to row 3 corresponding with their distance to the camera, row 1 being the closest. A major problem in this outdoor environment is wind, which results in partially extreme camera movement. This is especially prominent in row 3 as each parking space comprises of a relatively small number of pixels.

B. Classifier Training

Partly following up on earlier results (*cf.* [8]), we chose to train kNNs and SVMs for comparison. Although we achieved promising results with color feature input, we decided to train merely with DoG features of different filter sizes. This is due to the fact that the system is intended to be used with preinstalled surveillance cameras, which normally only capture grayscale images. We trained both kNN and SVM with DoG filter sizes of 3×3, 5×5, 7×7, 9×9, 11×11, 13×13, and 15×15.

The kNN classifier performs a clustering on the training set in order to reduce the number of examples that have to be stored. Regarding the kNN method, the parameters (*cf.* [8]) were kept at $k = 5$ cluster centers and 50 prototypes per class. For the SVM, training parameters were slightly altered as follows: We chose the influence of the kernel functions $\gamma \in [10, 10\,000]$ and the regularization parameter $C \in [10, 10\,000]$ for training, and kept the radial basis function kernel. We maintained 3-fold cross-validation for SVM model selection as described.

As a temporal smoothing strategy, we adapt the following filtering parameters: For Sequence A, we chose a constant learning rate of $\alpha = 0.8$ and a confidence threshold of 0.2. For Sequence B we kept the learning rate at $\alpha = 0.8$, however, adapted the threshold for each parking row. Row 1 was evaluated with a threshold of 0.6, row 2 was given a threshold of 0.5 and row 3 a threshold of 0.15.

The threshold values were chosen depending on the distance of the respective parking row to the camera, with smaller thresholds for bigger distances. This was necessary as the classifiers, trained on simpler image data as the one given to them for validation, had a lower confidence for parking spaces classified as "occupied". The lower the given threshold, the less confidence is required to set a parking space to "occupied". As parking rows that were further away from the camera had less pixels per parking space and thus less information available for the decision, the threshold was set lower than for the parking spaces in front. The threshold for Sequence A was estimated according to the parking row's distance to the camera, as well, lying close to row 3 in Sequence B.

C. Results and Discussion

First, we tested the performance of different feature/classifier combinations on Sequence A. The results are shown in Table I. The values denote percentages of accuracy. The accuracy for each parking space was calculated as the ratio of correctly classified frames with respect to one parking space to the total number of frames. The average performance of individual results amounts to the row accuracy. Each frame was evaluated. Accuracies highlighted in gray show the best performances for each classifier type. It is clear to see that the SVM outperforms the corresponding kNN classifier in all cases. For both classifiers too small filter sizes were unfavorable. While SVMs manage a peak accuracy of 88.46 %, the best kNN classifier results at 84.51 %, with almost 4 % less accuracy.

TABLE I: Classification results of Sequence A for kNN and SVM classifiers for different feature inputs

	classification rate [%]	
DoG kernel size	**kNN**	**SVM**
3 × 3	71.68	77.41
5 × 5	68.21	78.59
7 × 7	74.72	83.85
9 × 9	77.92	88.46
11 × 11	77.90	88.40
13 × 13	70.87	81.71
15 × 15	84.51	85.15

In a direct comparison with [8], the classifiers trained on real-world image data show slightly better results. Here, the SVM gained a peak performance of 94.13 % on Sequence A using a DoG filter size of 9×9. The best performing kNN resulted in 93.58 % accuracy, using the biggest filter size tested, *i.e.* 17×17.

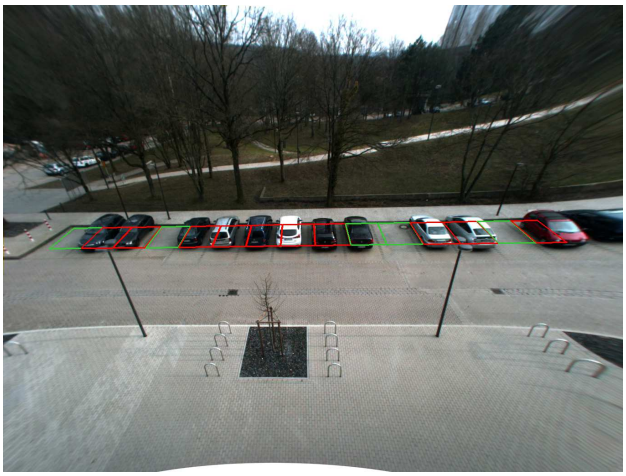


Fig. 4: Visualized classification of Sequence A. Green boxes are detected as available, red boxes identify occupied parking spaces. Overall 15 parking spaces were monitored in this sequence.

Although the simulated classifier results are worse than could be expected from traditionally trained classifiers, *i.e.* with real-world image data, they are solid enough to confirm that both classifier types are able to classify empty and occupied parking spaces in real-world footage up to a certain

TABLE II: Classification results of kNN classifier on Sequence B for different feature inputs

DoG kernel size	classification rate [%] row 1	row 2	row 3	average
3 × 3	89.15	86.38	77.07	84.20
5 × 5	88.89	87.61	86.84	87.78
7 × 7	86.84	92.54	88.00	89.12
9 × 9	87.78	93.88	88.12	89.92
11 × 11	89.04	94.03	88.02	90.36
13 × 13	91.22	96.68	88.51	92.13
15 × 15	97.79	96.66	94.37	96.27

TABLE III: Classification results of SVM classifier on Sequence B for different feature inputs

DoG kernel size	classification rate [%] row 1	row 2	row 3	average
3 × 3	97.43	99.05	78.13	91.54
5 × 5	95.96	95.73	88.36	93.35
7 × 7	96.27	96.81	87.89	93.66
9 × 9	97.62	96.41	88.58	94.20
11 × 11	97.49	94.95	86.80	93.08
13 × 13	97.97	95.68	83.65	92.43
15 × 15	92.12	98.87	94.45	95.15

extend. The results were visualized directly in the video during the classification process, as can be seen in Fig. 4. Green boxes are classified as empty parking spaces, red boxes as occupied ones.

Sequence B showed overall better classification results for all rows. This holds for all feature/classifier combinations. While row 3 has the weakest classification rate, due to its distance to the camera, the front two rows are well classified. The kNN classifiers (listed in Table II) show a clear tendency towards bigger DoG filter sizes, resulting in a peak performance of 97.79 % accuracy in row 1 and a top average accuracy of 96.27 %. The latter even outperforms the otherwise slightly better results of the SVM classifiers (*cf.* Table III) with a best average classification rate of 95.15 %. However, the SVM's peak performance was gained for row 2 with the smallest feature input, resulting in 99.05 %. Taking a closer look into the row performance, a clear drop of accuracy can be seen for row 3. This holds for all classifiers except one, and can be explained by its distance to the camera. As depicted in Fig. 5, this row can classify the highest number of parking spaces, but at the same time has less pixels per parking space for classification, resulting in a less confident evaluation of an occupancy status. Still the results for this row are well acceptable.

Comparing these results once again to those achieved in [8] with real-world data training, the real-world SVM has a slightly better average classification rate on Sequence B with 96.43 % compared to 95.15 % achieved by synthetic data training. However, the direct comparison of best-performing kNN classifiers shows a gap of nearly 6 % regarding the average 90.25 % for real-world training data to 96.27 % accuracy. In this case, the simulated data classifier clearly outperforms its real-world pendant, hereby strengthening the presented approach. Again, the classification results were visualized during the classification process. Fig. 5 depicts the image overlay.

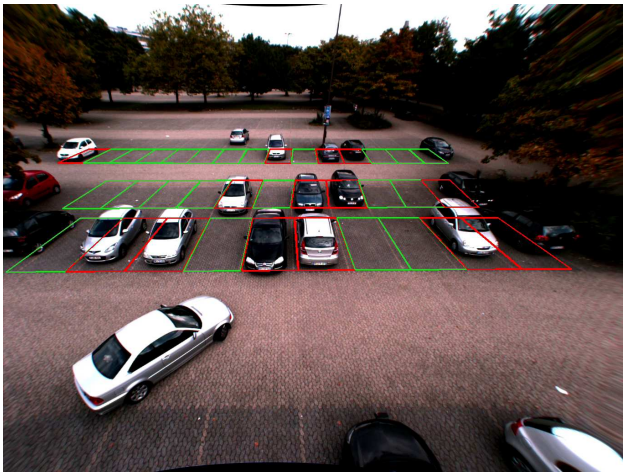


Fig. 5: Visualized classification of Sequence B. Green boxes are detected as available, red boxes identify occupied parking spaces. There were a total of 36 parking spaces monitored in three rows. Row 1, closest to the camera, holds 10 observed parking spaces, row 2, in the middle, 11 parking spaces, and row 3, which is farthest away from the camera position, can handle 15 parking spaces.

V. CONCLUSION

In this paper we proposed a new approach to generate synthetic image data for training purposes in machine learning and image processing algorithms, focusing on the task of parking space classification. By using a simulated car park environment in Unreal Engine 4 for image data and ground truth extraction, we have presented a time-saving alternative to traditional video recording and manual labeling. We evaluated our approach of replacing real images with synthetic data by training a number of kNN and SVM classifiers with various DoG features and tested them on two different real-world scenarios. The system gained a classification accuracy of up to 99.05 %. A detailed side-by-side comparison of the results on all sequences, taking all parking deck configurations and classifier parameters into account, revealed that replacing real-world with synthetic data slightly decreases performance, but the effort for data acquisition and labeling is significantly reduced. All in all, we have shown that the task of parking space classification can be considered solved without the use of real video data.

Although the potential usability of synthetic video data for classifier training has been shown in this paper, there is still

room for improvement. The simulated car park environment that was used for training purposes requires further optimization, aiming for even more realistic looks and vehicle behavior. More diverse data should be acquired from other simulated car park environments to make classifiers more robust against different natural lighting and weather conditions. Also an extended evaluation of grayscale-compatible input features other than DoG features might result in classifiers which are more suitable for real-world use. As data creation is no longer problematic, machine learning approaches with the requirement for big amounts of input data, such as deep neural networks, can be examined for this and other tasks as well.

REFERENCES

[1] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[3] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *PAMI*, vol. 34, 2012.

[4] C. Caraffi, T. Vojir, J. Trefny, J. Sochman, and J. Matas, "A System for Real-time Detection and Tracking of Vehicles from a Single Car-mounted Camera," in *ITS Conference*, 2012, pp. 975–982.

[5] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark," in *Proceedings of the International Joint Conference on Neural Networks*, no. 1288, 2013.

[6] M. Tschentscher, B. Pruß, and D. Horn, "A simulated car-park environment for the evaluation of video-based on-site parking guidance systems," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, vol. 28, 2017, pp. 1571–1576.

[7] M. Tschentscher, M. Neuhausen, C. Koch, M. König, J. Salmen, and M. Schlipsing, "Comparing image features and machine learning algorithms for real-time parking-space classifiaction," in *Proceedings of the ASCE International Workshop on Computing in Civil Engineering*, 2013, pp. 363–370.

[8] M. Tschentscher, C. Koch, M. König, J. Salmen, and M. Schlipsing, "Scalable real-time parking lot classification: An evaluation of image features and supervised learning algorithms," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2015, pp. 1–8.

[9] D. Horn and M. Brüggenthies, "Video-based parking space detection: Localisation of vehicles and development of an infrastructure for a routeing system," in *Proceedings of the Forum Bauinformatik*, 2015, pp. 175–182.

[10] PTV GROUP, "What keeps traffic flowing?" Brochure, 2017. [Online]. Available: http://vision-traffic.ptvgroup.com/fileadmin/files_ptvvision/Downloads_N/0_General/2_Products/2_PTV_Vissim/BRO_PTV_Vissim_EN.pdf

[11] Y. Yuan and K. Liu, "The 3d-simulation implementation of parking guidance system," in *International Conference on Audio, Language and Image Processing*, 2014, pp. 635–639.

[12] S. G. Konrad, M. L. Moreyra, and F. R. Masson, "The use of game engines and virtual models to build a simulator for intelligent transportation systems," in *2015 XVI Workshop on Information Processing and Control (RPIC)*, 2015, pp. 1–6.

[13] Q. Miao, F. Zhu, Y. Lv, C. Cheng, C. Chen, and X. Qiu, "A game-engine-based platform for modeling and computing artificial transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 343–353, 2011.

[14] Q. Wang and M. Abbas, "Using game engines for designing traffic control educational games," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 185–189.