Urban Environment Simulator for Train Data Generation Toward CV Object Recognition

Kirill Karpov[1,2], Ivan Luzianin[1], Maksim Iushchenko[1,2] and Eduard Siemens[1]
[1]Future Internet Lab Anhalt, Anhalt University of Applied Sciences, Bernburger Str. 57, 06366 Köthen, Germany
[2]Department of Transmission of Discrete Data and Metrology, Siberian State University of Telecommunications and Information Sciences, Kirova Str. 86, 630102 Novosibirsk, Russian Federation
{kirill.karpov, ivan.luzianin, maksim.iushchenko, eduard.siemens}@hs-anhalt.de

Keywords: Virtual Reality, Computer Vision, Object Detection, Public Street Environment, Simulation, Traffic, Pedestrian, Distribution, Labeling.

Abstract: Detecting moving objects in an urban environment is a challenging and widely explored problem in computer vision. This task requires huge amounts of data. Their obtaining and labeling is challenging. However the available datasets are not always fit the task. This work proposes a framework for synthesizing the train data based on 3D visualization of an urban environment using Unity 3D. Methods of mathematical statistics and distribution theory were used to build the background models of the framework. The framework, presented in the article, allows to simulate the real urban environment in an adjustable 3D virtual scene. It considers different environmental parameters amd makes possible to simulate the real behavior and physical characteristics of moving objects.

1 INTRODUCTION

There are an enormous amount of publically available datasets for the training of computer vision models. Nevertheless, there is a lack of data considering the specific parameters of scene or equipment, e.g. camera location, scene parameters, weather conditions, time of day, etc. Although, these parameters are critical for the dimensional based method developed in the previous works [1, 2]. The intrinsic and extrinsic parameters of the camera are significant for this algorithm.

Manual data collecting in real streets is time-consuming. There are streets with low traffic, hence the observation area is empty the majority of time. To provide data variety many different streets must be investigated. Moreover, of some rarely occuring events like car accidencs, appearance of jogging persons or disabled people must be gathered, it might take a while. Also, a real environment is pretty inconvenient for the development and debugging of object tracing algorithms [1]. In addition to the mentioned challenges, data labeling may take even longer time than data collection.

There are some approaches to synthesize the datasets by rendering the moving object on photo-background. However, these methods do not consider the effects which are important for background subtraction affecting the quality of the detection algorithms. Such possible effects are e.g. dynamic shadows, unexpected light reflections, or flares. Taking these effects into account makes the data more realistic.

A solution, proposed in this paper meets the above-mentioned requirements and allows the simulation of necessary conditions in 3D virtual reality. For this, Unity 3D engine provides a comprehensive toolset to recreate any kind of urban environment.

To make produced datasets representative in terms of a variety of moving objects the model creates pedestrians with different behavior and physiology based on data-driven statistical models. These models consider both, a variety in physiological features of real humans and their behavior changes depending on the environmental conditions and a time of day.

Statistical models are used for considering the changes in the behavior of vehicles during the day. It is also possible to vary the speed of the vehicle and produce ones with different colors.

The above considerations allow producing data that are very close to the real-world street conditions.

The remainder of this paper is structured as follows. Already existing methods are discussed in section 2. Section 3 describes the mathematical model for generating traffic data. The proposed virtual environment simulatior is described in section 4. In section 5, results of this research along as future work suggestions are presented.

2 RELATED WORK

The idea of train data generation for object detection purposes may be addressed in different ways. In the work [3] a framework based on the Generative Adversarial Network (GAN) with multiple discriminators is proposed. It aims to synthesize realistic pedestrians on a given picture and learn the background context simultaneously. The framework includes the following components: generator G and two discriminators (D_b for background context learning and D_p for pedestrian classifying). The generator G replaces pedestrians on ground truth pictures with bound boxes, filled out with random noise and generates new pedestrians within that noise region. The discriminator D_b, learns to differentiate between real and synthesized pairs and forces the generator G to learn the background information like road, light condition in noise boxes. In the meanwhile, discriminator D_p learns to judge whether the pedestrian is a synthetic or real one. It leads to a smooth connection between the background and the synthetic pedestrian. After training, the generator can learn to generate photo-realistic pedestrians in the noise box regions and the locations of noise boxes are taken as the ground truth for detectors. Adding the Spatial Pyramid Pooling (SPP) layer in the discriminator D_p enables generation of pedestrians of different sizes.

The articles [4, 5] propose an efficient discriminative learning method that generates a spatially varying pedestrian appearance model that takes into account the perspective geometry of the scene. The method considers the surveillance setting where the following information is available: (1) intrinsic and extrinsic parameters of the static camera and (2) the geometrical layout of the scene, i.e., semantic labels for all the regions in the scene where a pedestrian could possibly appear ("pedestrian region") and semantic labels for obstacles in the scene where a pedestrian could either be occluded or physically cannot be present. The area labeling performs manually. This obtained information is leveraged along with synthesized 3D pedestrian models to generate realistic simulations of the appearance of pedestrians for every location of the "pedestrian region". All artificial pedestrians are being rendered with respect to the camera parameters and the geometrical layout of the entire scene e.g., obstacles and occlusions. Consequently, these data are used to learn a smooth spatially-varying scene-specific discriminative appearance model for pedestrian detection.

The publication [6] proposes to replace or complement the real training data with augmented data, i.e. photo-realistic images comprised of virtual agents rendered onto a real image background. A sequence of real recorded images acquired from low-resolution vehicle-based cameras is used to reconstruct the surrounding 3D scene. Virtual pedestrians are being put in non-occluded positions and then animated in the reconstructed scene. Illumination is added to the scene to match the environment, and also simple geometry to cast and receive shadows. The bounding box for each virtual pedestrian is automatically generated using the alpha-mask obtained from rendering.

3 TRAFFIC GENERATION MODEL

The aim of a traffic generation model is to define a number of moving objects to be created by the simulator at each certain time period.

The straightforward approach is to generate data randomly [5]. However, this solution has one significant disadvantage: in this case, it is not possible to adjust the number of moving objects according to the real traffic situation in the particular street. In contrast, a data-driven approach, where the number of objects is generated by the predefined data-based function, enables simulating the traffic distribution for any particular street.

To obtain a data-based function one can either build a regression model, e.g. by using of neural networks [7] or use a probabilistic approach [8]. In the second case, the quantity of moving objects is to be considered as a random variable varying in the 24-hours time domain. The function itself is a probability density function (PDF) of a known distribution. This approach allows to directly find the expected number of objects from the histogram depending on the total number of objects while the regression model is a polynomial function of the n-th degree. The data preparation and modeling are described in the following subsections.

3.1 Data Preparation

The following moving objects are considered in the article: cars, motorcycles, trucks, buses, bicycles and pedestrians.

Three different data sources were considered to be the input in the modeling process.

- Traffic Counts - Hourly Classification Counts 2017 [9] - the dataset contains a set of records collected from different observation stations located in different roads of USA. Each record specifies vehicle count passed near a certain station on a certain date and within a certain hour. All vehicles are divided into several aggregated classes such as motorcycles, passenger cars, pickups and panel vans, buses, single-unit trucks, multi-unit tracks. From this dataset, only the data on car traffic was derived and used in further research.
- Bike Counts (Eco Counter) [10] - the dataset provides bicycle and pedestrian counts that were monitored at a number of locations in Edmonton, Canada. The data recording was carried out at 15-minute intervals. The part of the dataset related to bicycle traffic was derived to use in further research. Each record contains a vehicle count for two vehicle types (light or hard vehicles depending on their length). The part of the dataset related to bicycle traffic was derived to use in further research.
- Pedestrian Counting System – Monthly [11] - the dataset contains hourly pedestrian counts since 2009 from pedestrian sensor devices located across Melbourne, Australia. The data were aggregated from a variety of sources. The dataset was used in pedestrian traffic research.

Before modeling, the data were cleaned and averaged. During data cleaning, the duplicates and uncompleted rows were removed. Since the required model needs to describe the average behavior of the moving objects, the outliers in data were also removed. Although the second dataset includes both pedestrian and bicycle data, the last ones have many outliers and incomplete rows. For this reason, the third dataset was used for modeling pedestrian behavior.

After cleaning the data were averaged by the standard hour periods. After that, the whole observation history for each hour was also averaged. Finally, for each single observation point, the quantities of objects were obtained.

3.2 Data Modeling

Independently from the type of a moving object, the traffic is assumed to be normally distributed with the following PDF.

$$\phi_{\mu,\sigma^2}(X) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \quad (1)$$

It is usually expected to see more moving objects in the streets in the morning and in the evening, than in the afternoon and at night, therefore one can assume the distribution to be bimodal [8].

To prove the above assumptions the traffic intensity histograms for averaged data were built. All the data were proven to be normally distributed. The assumption of bimodality was not confirmed because among the data there are cases where the distribution has a different number of peaks as shown in Figure 1.

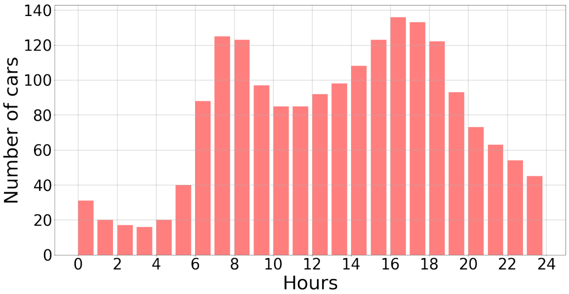

Figure 1: Averaged 24-hours traffic intensity histogram for individual cars.

The assumption of bimodality was not confirmed because among the data there are cases where the distribution have different number of modes. The above picture also illustrates, that the distribution is not purely bimodal, it has three local maxima. Therefore the general distribution is assumed to be multimodal. The equation 2 describes the distribution of traffic intensity I_t.

$$I_t = \sum_i A_i e^{-\frac{(t-t_i)^2}{s_i}}, \quad (2)$$

where i is a number of peaks on the histogram, t_i is a time where the i-th peak appears and A_i is a value of traffic intensity at the time t_i. This equation is an extension of that given in [8].

The above considerations allow using a Gaussian mixture model to estimate the parameters of PDF for the traffic intensity distribution. To do this the individual traffic intensities were recalculated into probabilities of observing the certain number of cars by the following formula:

$$P(I_t) = \frac{I_t}{\sum_i I_t} \quad (3)$$

The resulting PDF was obtained as a sum of PDFs for each $P(I_t)$ with an equal variance. The plot of individual PDFs for the cars is presented in Figure 2.

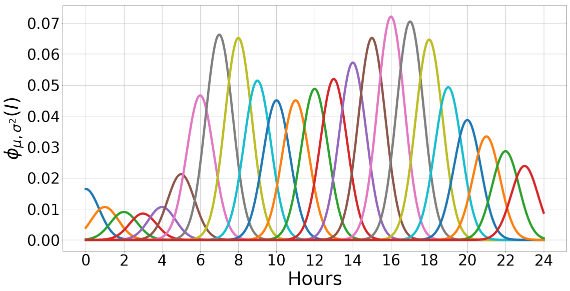

Figure 2: Plot of individual PDFs for the cars traffic intensities.

The resulting PDF for the car traffic intensity is presented in Figure 3. One can observe the first peak between 00:00 and 01:00 that will be lost when modeling with equation proposed in the paper [8].

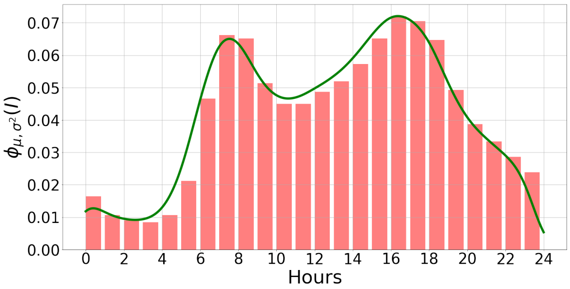

Figure 3: Histogram and resulting PDF curve for cars traffic intensities.

The resulting PDF for the traffic intensity of motorcycles is presented in Figure 4. The motorcycles are usually driving in the evening, that generally corresponds to real situation.

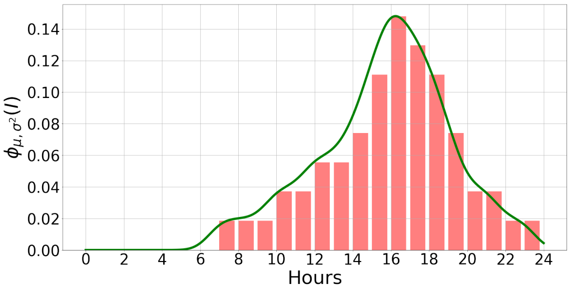

Figure 4: Histogram and resulting PDF curve for motorcycles traffic intensities.

The resulting PDF for the traffic intensity of trucks is presented in Figure 5. One can observe the highest peak at 08:00. After 11:00 the curve is closer to the uniform distribution than for cars.

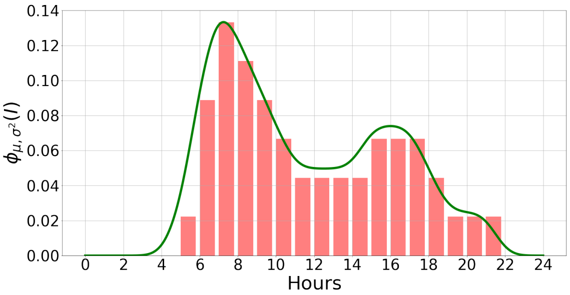

Figure 5: Histogram and resulting PDF curve for trucks traffic intensities.

The resulting PDF for the traffic intensity of buses is presented in Figure 6. One can observe only two segments on the histogram where the probability of observation is non-zero. The traffic of the buses is lower than that of other vehicles and that the buses use the schedule, while other objects move more randomly.

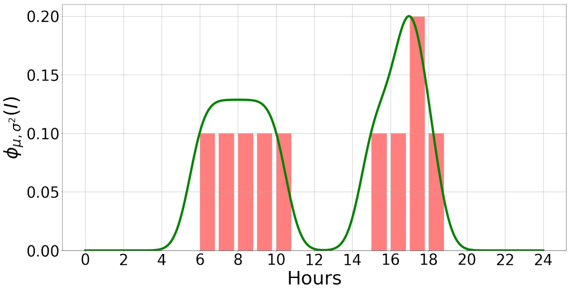

Figure 6: Histogram and resulting PDF curve for buses traffic intensities.

The resulting PDF for the traffic intensity of cars is presented in Figure 7. One can observe more extreme slopes in the PDF curve than in previous cases. It means, that cyclists are moving more randomly than other moving objects.

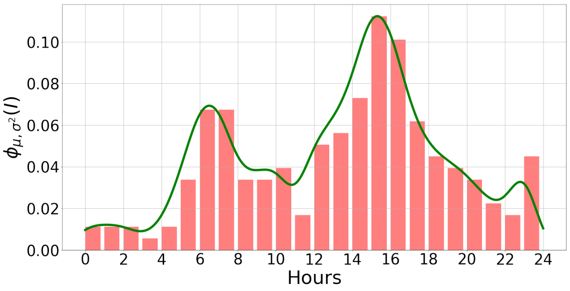

Figure 7: Histogram and resulting PDF curve for bicycles traffic intensities.

The resulting PDF for the traffic intensity of pedestrians is presented in Figure 8. One can observe multiple peaks on the histogram, therefore it might be concluded, that the pedestrian traffic behavior is more complex than that for vehicles.

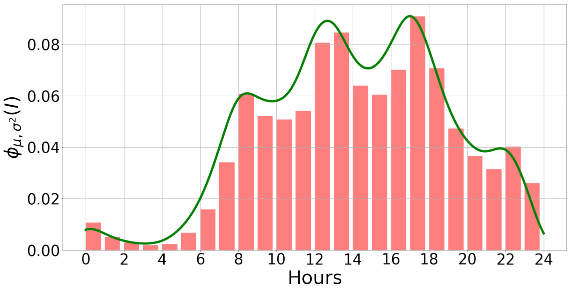

Figure 8: Histogram and resulting PDF curve for pedestrians traffic intensities.

To use the obtained model, one needs to define the total number of objects. Then the model will calculate the number of objects of a certain type at the given time point based on the PDF.

4 VIRTUAL ENVIRONMENT SIMULATOR

This section provides a detailed description of the testing infrastructure and software, which is used during the experiments.

The urban environment simulator's structure is shown in Figure 9, it consists of the following components:

1) Population Dataset - the dataset which contains the anthropometric parameters such as gender, height, width, weight, etc.
2) Generator of Anthropometric parameters - this component generates anthropometric parameters from the statistical model based on Population Dataset and passes the command to MakeHuman.
3) MakeHuman v1.2.0 (with Mass Produce plugin) - an open-source 3D computer graphics middleware designed for the prototyping of photorealistic humanoids. The example of generated models is shown in Figure 11.
4) Traffic History Dataset - a dataset which contains the data about the events that the object appears in the observation area [9, 10, 11].
5) Event Generator (EG) - a statistical model which generates the events that the object generated on step 3) will appear in the unity scene according to density data on step 4.
6) Unity 3D v2019.4.20f1 LTS - cross-platform 3d engine developed by Unity Technologies. It is used to generate the simulation scene of urban environments, animation of the moving object generated on step 3.The example of 3D scene is shown in Figure 12.
7) Labeled Dataset - the dataset obtained from 3D scene from step 6.

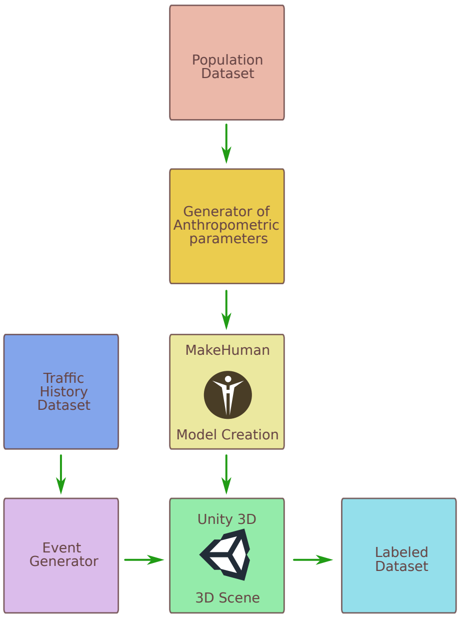

Figure 9: The components of the urban environment simulator.

4.1 Virtual Objects Generation

Pedestrian models are generated based on the anthropometrical studies of human diversity [12, 13]. The obtained statistical parameters which are shown in Table 1 are used to build a statistical model of the population which considers the height and weight of the individuals.

Table 1: Statistical parameters from population data.

Male	
μ_{height}	175.5 cm
σ_{height}	5.9 cm
μ_{weight}	74.49 kg
σ_{weight}	11.11 kg
Female	
μ_{height}	166 cm
σ_{height}	5.5 cm
μ_{weight}	60.3 kg
σ_{weight}	5.7 kg

From the statistical parameters, it is possible to make a generator of sets of anthropometrical parameters for the models. The generator model can be described as bivariate normal distribution, which density function is shown in Figure 10.

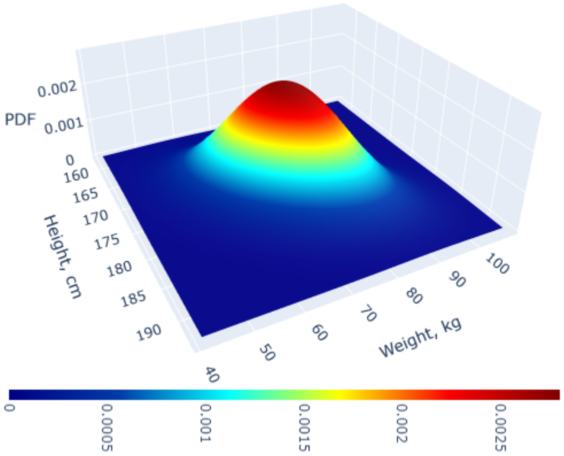

Figure 10: Bivariate normal distribution for height and weight for men.

The generated parameters are passed into MakeHuman software to produce 3D models. The models are shown in Figure 11.

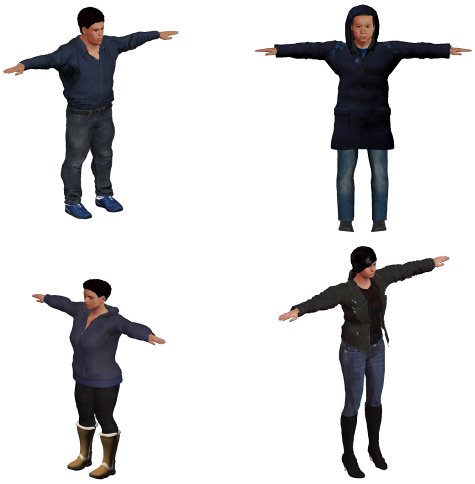

Figure 11: Sample of pedestrians generated by MakeHuman.

The produced models will appear on the Unity 3D scene according to the scheduler from the event generator. The example of 3D scene is shown in Figure 12. The event generator is responsible for notification about the time and type of the object which will appear on the scene. It provides notification about several types of objects: pedestrians (male or female), cars, trucks, and cyclists. For vehicles, EG generates the parameters of color and type of vehicle (truck, bus, sedan, van, sports car, etc).

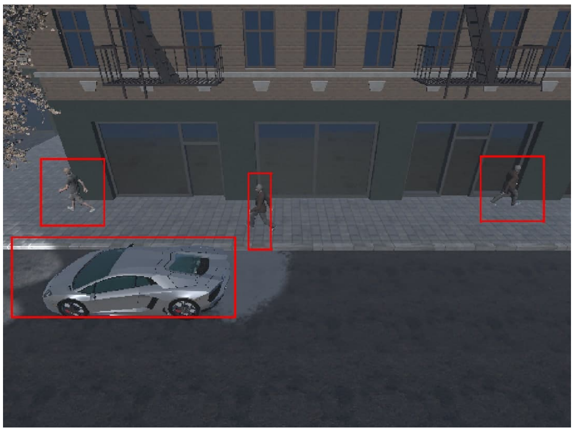

Figure 12: Daytime scene example.

Unity 3D application is responsible for urban environment visualization, illumination changes according to the time of day, animation of the objects, weather conditions on the scene, and generation of labeled screenshots with intrinsic camera parameters (FOV, height, and width of the matrix, IR filters, IR light 13, etc).

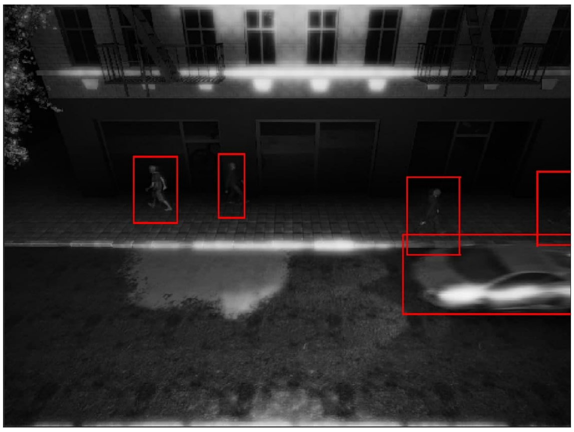

Figure 13: Nighttime scene with IR camera simulation.

5 CONCLUSIONS

The framework, presented in the article, allows to simulate the real urban environment in an adjustable 3D virtual scene. It makes it possible to simulate the real behavior and physical characteristics of both pedestrians and vehicles.

The models in the background of simulation are based on real data, which makes it possible to generate training data of high variety. To build them, the study of traffic behavior was carried out during the research. It was shown that traffic behavior generally has multimodal Gaussian distribution independently on the type of moving object. Obtained PDFs allow using only the total number of objects per day to generate the objects at the scene. They also make the simulation close to the real situations and allow simulating custom events such as car accidents or public feasts based on user data.

Since the simulator is able to increase and decrease the speed of time, it is possible to simulate behavior during a large time period (e.g. month) faster than it is to be done in a real life. The simulator is also able to install different cameras at any instant point of the environment and adjust their parameters, which makes the data generation cheap and also allows to produce the data for the computer vision algorithms of many different types including dimensional-based ones. The parameters of illumination are also adjustable, which makes it possible to simulate night scenes or IR cameras.

The study of the model shows that virtual traffic simulation corresponds to the input data. This indicates that the simulation is consistent with the defined behavior of the objects.

The proposed framework could be instantly used for train data generation, however, there is room for improvement. In future the framework can be improved in the following ways:

Scene creation is a laborious and time-consuming task. Since the automatic scene generation should be considered. For example, a scene generation algorithm may take photos or videos as input data.

In addition to the above mentioned, it is possible to recreate the context of the scene. The scene context defines where the objects may be spawned and how they could behave.

It makes it possible to consider the traffic behavior as a periodic Gaussian process. The theoretical background of such a process is given in [14]. The statistical model is used for calculating the number of objects to be created on the scene at every instant time period. The statistics were obtained by averaging the real observations during the long time period. Finally, the average 24-hours histogram was built. However, under real-world the traffic behavior is cyclic, i.e. it is expected to find the same pattern at the same time each day. The traffic intensity variation limits can be investigated and then the model can be extended using the periodic Gaussian process.

REFERENCES

[1] I. Matveev, K. Karpov, A. Yurchenko, and E. Siemens, “The object tracking algorithm using dimensional based detection for public street environment,” Eurasian Physical Technical Journal, vol. 17, pp. 123–127, Dec. 2020.

[2] I. Matveev, K. Karpov, I. Chmielewski, E. Siemens, and A. Yurchenko, “Fast object detection using dimensional based features for public street environments,” Smart Cities, vol. 3, no. 1, pp. 93–111, 2020.

[3] X. Ouyang, Y. Cheng, Y. Jiang, C.-L. Li, and P. Zhou, “Pedestrian-synthesis-gan: Generating pedestrian data in real scene and beyond,” arXiv preprint arXiv:1804.02047, 2018.

[4] W. Zhang, K. Wang, H. Qu, J. Zhao, and F.-Y. Wang, “Scene-specific pedestrian detection based on parallel vision,” arXiv preprint arXiv:1712.08745, 2017.

[5] H. Hattori, V. N. Boddeti, K. Kitani, and T. Kanade, “Learning scene-specific pedestrian detectors without real data,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA: IEEE, Jun. 2015, pp. 3819–3827.

[6] J. Nilsson, P. Andersson, I. Y. Gu, and J. Fredriksson, “Pedestrian detection using augmented training data,” in 2014 22nd International Conference on Pattern Recognition. IEEE, 2014, pp. 4548–4553.

[7] F. Moretti, S. Pizzuti, S. Panzieri, and M. Annunziato, “Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling,” Neurocomputing, vol. 167, pp. 3–7, 2015.

[8] L. Bartuška, V. Biba, and R. Kampf, “Modeling of daily traffic volumes on urban roads,” 2016.

[9] Metropolitan Washington Council of Governments. Traffic Counts - Hourly Classification Counts 2017. Accessed Mar. 20, 2021. [Online]. Available: https://rtdc-mwcog.opendata.arcgis.com/datasets/fae4f4ebf99c45088adbfba504efd650

[10] Bike Counts (Eco Counter). City of Edmonton. Accessed Mar. 20, 2021. [Online]. Available: https://data.edmonton.ca/Monitoring-and-Data-Collection/Bike-Counts-Eco-Counter-/tq23-qn4m

[11] City of Melbourne Open Data Team. Pedestrian Counting System - Monthly (counts per hour). Accessed Mar. 20, 2021. [Online]. Available: https://data.melbourne.vic.gov.au/Transport/Pedestrian-Counting-System-Monthly-counts-per-hour/b2ak-trbp

[12] S. Buchmueller and U. Weidmann, “Parameters of pedestrians, pedestrian traffic and walking facilities,” IVT Schriftenreihe, vol. 132, 2006.

[13] E. Brolin, “Anthropometric diversity and consideration of human capabilities,” p. 101.

[14] N. HajiGhassemi and M. Deisenroth, “Analytic long-term forecasting with periodic gaussian processes,” in Artificial Intelligence and Statistics. PMLR, 2014, pp. 303–311.