CHAPTER 10

# Simulation Tools

## Contents




*Intelligent Vehicles*
DOI: http://dx.doi.org/10.1016/B978-0-12-812800-8.00010-2

SUBCHAPTER 10.1

# Driving Simulators

**Alfonso Brazález[1], Luis Matey[1], Borja Núñez[1] and Ana Paúl[2]**
[1]Ceit-IK4, San Sebastián, Spain
[2]CTAG - Centro Tecnológico de Automoción de Galicia, Porriño, Spain

## 10.1.1 INTRODUCTION

The continuous improvement of hardware performance is a well-known fact that is allowing the development of more complex driving simulators. The immersion in the simulation scene is increased by high fidelity feedback to the driver. A high quality visual system in terms of, high image refresh rate, realistic environment, and good 3D modeling increases the immersion in the simulation session. The behavior given by the mathematical models affects the scope of the simulation and the fidelity of the simulation itself.

In this framework, the application of new methods and the need to provide more realism have generated new requirements for simulator performances. The immersive character in the simulators is obtained by the stimulation of the sensorial organs of the driver, so sensations experienced by the human being are most similar to those that the simulator user feels driving the actual vehicle. Various senses are stimulated in the simulator: the visual sense, the hearing sense, the tactile sense and the vestibular system. The motion system usually consists of a six degrees of freedom (dof) motion platform that reproduces the sensations of linear accelerations and angular velocities that the driver feels in the actual vehicle (Liu, 1983). There are simulators with more complex motion systems, even without motion systems, or with simpler motion platforms of three degrees of freedom.

Some simulators provide haptic feedback to establish the interactions between the driver and some controls of the vehicle in driving simulators. Steering wheel torque feedback, configurable joysticks, and the actuators in the accelerator, clutch, and brake pedals enhance the degree of realism in the simulation and they allow the driver to feel realistic forces and torques on his arms and legs.

Through increasing the immersion in the simulation, driving simulators will get a bigger transfer rate of results from the virtual environment to real life in several fields. Automotive simulators are an important research tool in design, development, and validation stages. Naturalistic driver studies can achieve certain objectives, but experiments are a more appropriate approach to test hypotheses (Gelau et al., 2004). Some information such as subjective, physiological, or other performance data are not collected in naturalistic studies. Although the real road and FOTs (Field Operational Tests) give more realistic surroundings, simulators are necessary in order to reduce experimental costs and risks. On real roads, experiments suffer extra noise (Carsten and Brookhuis, 2005), while in a simulator with a fully controlled scene it is possible to produce the exact desired situation. Thus, getting a realistic environment is the key requirement of automotive simulators.

Apart from the investigation field, the main application of automotive simulators is training purposes. They can be used to instruct both novice and experienced drivers. According to the GADGET project, driving training is divided into four levels: maneuvers, traffic situations, context goals, and skills (Peräaho et al., 2003). In practice, however, it was concluded that most simulators only covered the first two levels, whereas the other two levels typically are omitted from the driver training because of the limitation of simulators (Lang et al., 2007). Increasing the configurability of the HMI enables new possibilities in training simulators especially in these two highest levels.

Focusing on the field of research, automotive simulators provide a wide range of possibilities in different disciplines (Slob, 2008). On the one hand, it is possible to analyze and investigate human factors (driver behavior and HMI); on the other hand, it is possible to use them to design and validate environmental issues, like tunnels, positioning road signs, or road planning. Finally, simulators are useful tools to develop, validate, and evaluate technical and technological innovations.

Simulators present two main benefits on R&D performance (Thomke, 1998). First, their use reduces costs and time, facilitating design iterations; second, they help to achieve a more effective learning in the R&D process, because they can increase the depth and the quality of the experimental analysis. From the point of view of experiments, a simulator adds other advantages: it is possible to reproduce hazard situations without any risk for the driver and a simulator enables to keep all the vehicle and environment parameters under control.

In the automotive area, the approach to technical innovations is very wide; it covers different tasks, such as the integration of new subsystems and applications, communication related issues, or ADAS/IVIS development and testing. Simulator capabilities have a special importance in all the cases. The more configurable a simulator is, the faster and easier will be the creation of new scenarios and their implementation for experiments.

Alongside the technical success, new systems implementations must reflect the study of human factors to guarantee their safety and usability. In this field, there is no standard and general methodology to measure the validity of a certain simulator (Eskandarian et al., 2008); this concept is strongly linked to the task to be performed. While the motion platform and a high resolution of the visuals can help to enhance the realism (Kaptein et al., 1996), the main conclusion is that the validity must be evaluated depending on the driving task being studied.

Although a normalized classification does not exist yet, there are different perspectives to categorize the existing simulator types: depending on whether they have motion or not; determined by the type of vehicle (car, bus, or truck); depending on the visual type; or subject to the simulation scope.

## 10.1.2 ARCHITECTURE OF DRIVING SIMULATORS

There are many configurations and scopes of use of driving simulators, but usually the main subsystems are: visual system, simulator control station, mathematical models, vehicle cabin, audio system and the motion system if the simulator provides motion feedback. Moreover, when the simulator is coordinated, it can be used as a host system for real-time coordination of every subsystem (Fig. 10.1.1).

The Simulator Control Station is the simulator interface. From this station different exercises can be selected. Usually, it provides an edition mode for the creation of new exercises; based on the configurability

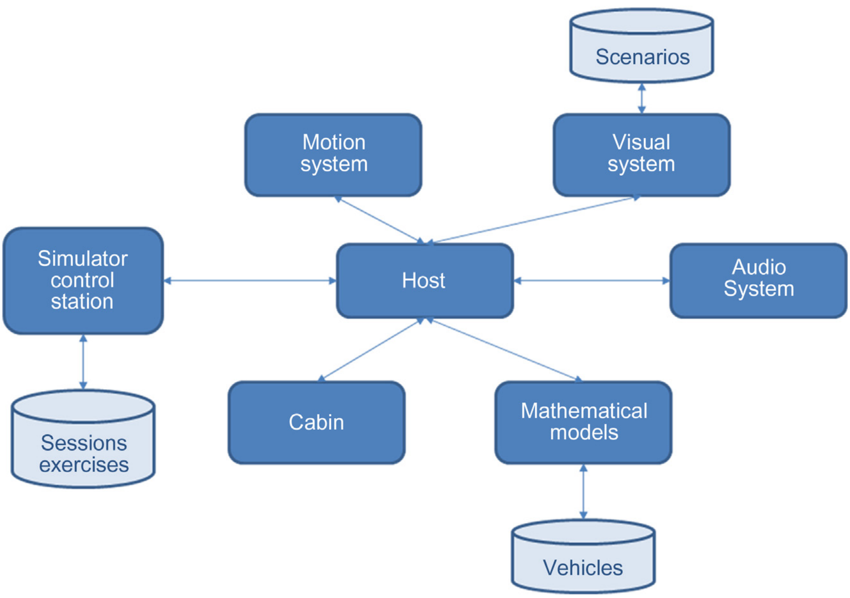



**Figure 10.1.1** Driving simulator global architecture.

of the simulator modules, it can change vehicle parameters or weather conditions and set new incidences. In the Control Station, the simulated sessions can be stored for further evaluation and analysis.

The visual system provides visual feedback to the driver. Nowadays, it is possible to simulate different lighting conditions, weather conditions, and moving elements in the scene with a high degree of fidelity. From the hardware point of view, visualization systems are more efficient with a high resolution and visualization performance. It is common to have a 360 degrees of field of view. It is important to have a rear view though the rear mirrors and for front view, it is necessary for at least 180 degrees, covering the side views. The gaming sector has improved the development of dedicated hardware for real-time image rendering, with relatively low cost systems. Moreover, from the software point of view it is possible to program directly in the Graphic Processing Unit (GPU) of the image system. It allows for a better performance for image rendering in real time reducing the traffic data on the network.

Scene elements are critical for having a better feeling of immersion. The surrounding traffic provides additional elements for interaction. Many incidences can be simulated from other vehicles: a vehicle stopped in the shoulder, a vehicle at low speed, a vehicle passing through a red light, etc. A realistic microscopic traffic simulation increases the driving simulator capabilities. Also the pedestrian simulation leads to higher capabilities of the simulator, because it allows for the interaction with other elements of the scene. Various incidences related to pedestrians can be simulated, for example people crossing streets or passengers in a bus simulator.

The mathematical models are a key module for the performance of the simulator. It affects the simulation scope. Depending on the needs of the simulator, a high accuracy mathematical model could be required. Moreover, if the simulator should be used for the validation of new subsystems, where some hardware must be integrated as Hardware In the Loop (HIL), a well-defined interface should be provided as it was simulated. Complete vehicle models should reproduce the effect of the tires with a detailed mathematical model of the contact between road and tire, using different approaches of the Pacejka Magic Formula. For the vehicle dynamics, a multibody simulation must be performed, including the effect of the suspension with its geometry, parameters, and nonlinearities. The complete powertrain should be modeled; engine and transmission elements need to be simulated and any other system, such as e.g., ASR. Additional subsystems such as ABS or ESP can be simulated for a high fidelity simulator.

Not all driving simulators are equipped with a motion system. This tries to provide some motion feedback to the driver. The strategy of this system is to translate vehicle linear accelerations and angular velocities into driver moving sensations. Typically washout filter algorithms inherited from flight simulators were used, but there are some limitations due to large longitudinal and lateral accelerations in the case of road vehicles. It is necessary to develop appropriated algorithms in order to ensure the driver feels realistic motion feedback (Ares et al., 2001), and some additional degrees of freedom have been added to typical 6 dof motion platforms. Many driving simulators mount the 6 dof motion platform over a large excursion translation system in lateral and longitudinal. Additionally the motion platform can be rotated along his the vertical axis, reaching a 9 dof motion system.

It is important to have a real cabin, or at least real controls, for a better immersion of the driver in the simulation scene. This could avoid the initial gap of the driver for trusting in the simulator. A period of adaptation for the driver with several driving sessions is always needed, prior to obtaining any valid results of the simulation. A real vehicle, with a complete vehicle control set reduces the training sessions required for the adaptation. Once the driver feels he or she is actually driving, the results of the simulator can be transferred to real situations. In the cabin real vehicle communications can be installed for the integration of different subsystems. Usually CAN networks must be used for the connection of different ECUs or the integration of the instrument cluster or any other interface.

The audio system is not a minor feedback for the driver. If this system is switched off, the situation is like a deaf person is driving. Many inputs are received from hearing the engine, and sometimes the perception of gear changing is performed from the engine sound. Additionally, some malfunctions are identified from the specific sound that it generates.

### 10.1.3 APPLICATIONS

The use of driving simulators has evolved from the first applications related to driver training or analysis of the effect of different substances such as alcohol or drugs, to more complex research studies focused on human factors and driver behavior during primary and secondary driving tasks. Nowadays, driving simulators are used not only in research studies, but also in the different stages of the design, development, and validation of in-vehicle systems, as well as in the design of infrastructure elements (Paul et al., 2009).

Regarding human factors, the studies in the driving simulators have allowed examination of the human–machine interface (HMI), i.e., the communication between the user and the vehicle or its technologies covering the following aspects:

- Workload analysis.
- Usability.
- Distraction due to secondary tasks.
- Reaction times (e.g., to avoid a collision when offering a warning).
- The effects of driver information on driving performance.
- The location of HMI elements on the dashboard.
- Selection of communication channel (acoustic, visual, haptic).

The evaluation of In-Vehicle Systems (IVIS) and Advanced Driver Assistance Systems (ADAS) have been broadly applied in several research and development projects, showing that driving simulator studies are, together with tests on test tracks and tests in real life, valid tools (Engen et al., 2009). In fact, dynamic driving simulators play an important role in the initial phases of development, since driver behavior and driving performance (lateral control, longitudinal control, interaction with other vehicles, etc.) provide early valuable results in a virtual environment.

Several European Projects have introduced experimentation with driving simulators to analyze new in-vehicle functionalities. For example, in the AIDE project, funded by the Sixth Framework Programme, the effect of the combination of warnings coming from different ADAS functions were tested (Paul et al., 2008). Thus, four ADAS were selected for this study, Frontal Collision Warning (FCW), Lane Departure Warning (LDW), Curve Speed Warning (CSW), and Blind Spot Detection (BSD), in order to analyze user reaction and possible conflicts when simultaneous ADAS warnings were presented (Fig. 10.1.2).

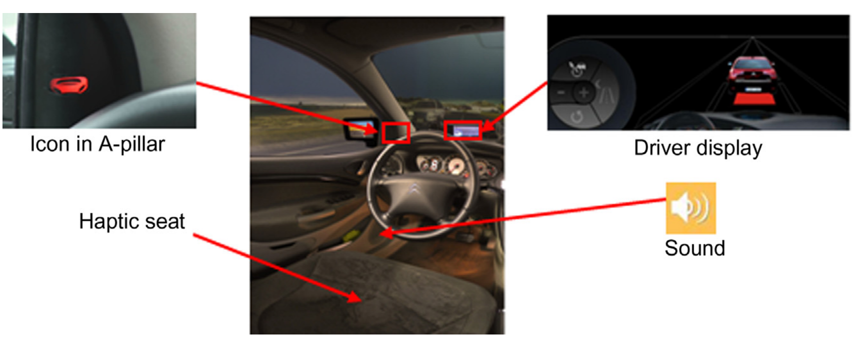



**Figure 10.1.2** Example of warnings provided by different ADAS functionalities.

In this study the warning conditions were manipulated to create different levels of theoretical mental workload. Two strategies were considered (independent variable) in a conflict situation: simultaneous activation of warning signals for the driver and prioritization of warnings. Such critical situations can only be reproduced in high performance driving simulators, with the required levels of safety and repeatability.

Another example of the application of driving simulators to the design of new ADAS is the INTERACTIVE Project (2013), launched under the Seventh Framework Programme, where information, warning, and intervention strategies were developed, according to aspects such as: layer of driving task, level of assistance and automation, situation awareness, mental workload, sequence of interaction, etc. (Fig. 10.1.3).


**Figure 10.1.3** Selection of ADAS validated in driving simulator (INTERACTIVE Project).

Apart from the evaluation of the HMI and warning strategies conducted in driving simulators, the vehicle HIL simulations have proven an added value in several phases of the development process of ADAS, such as sensor verification, rapid control prototyping, model validation, functional level validation, fine-tuning of control algorithms, production sign-off tests, and preparation of test drives (Gietelink et al., 2006). This is possible thanks to the representative environment obtained, where test scenarios can be varied very easily in accurate and reproducible conditions.

Moreover, in the development of ADAS, different driving simulation platforms may be used with the combination of HIL and driver-in-the-loop (DIL), in order to create special testing scenarios that include high speed traffic flow, low-frictional load, etc., with high controllability and repeatability (Jianqiang et al., 2010). This allows to speed up the development process and to reduce the associated development costs.

Finally, HIL simulation can be used to support the development, testing, and verification of many functions and algorithms related to autonomous driving, by extending conventional HIL simulation to vehicle interaction with other vehicles in traffic and with a simulated surrounding environment sensed by simulated sensors. (Deng et al., 2008).

In this sense, there is no doubt that driving simulation is an essential and powerful tool in the design, development, and validation of current and future in-vehicle technologies, especially in the field of connected and automated road transport, where new challenges regarding user behavior, vehicle operation, complex scenarios, and innovative interaction capabilities are rapidly arising.

## REFERENCES


Ares, J., Brazalez, A., Busturia, J.M. 2001. Tuning and validation of the motion platform washout filter parameters for a driving simulator. In: Driving Simulation Conference, pp. 295–304.

Carsten, O.M.J., Brookhuis, K., 2005. Issues arising from the HASTE experiments. Transp. Res. Part F Traffic Psychol. Behav. 8 (2), 191–196.

Deng, W., Lee, Y.H., Zhao, A., 2008. Hardware-in-the-loop Simulation for Autonomous Driving. IEEE 2008.

Engen, T., Lervåg, L.-E., Moen, T., 2009. Evaluation of IVIS/ADAS using driving simulators comparing performance measures in different environments.

Eskandarian, A., Delaigue, P., Sayed, R., Mortazavi, A., 2008. Development and verification of a truck driving simulator for driver drowsiness studies. CISR, The George Washington University.

Gelau, C., Schindhelm, R., Bengler, K., Engelsberg, A., Portouli, V., Pagle, K., et al., 2004. AIDE Deliverable 4.3. 2: Recommendations for HMI Guidelines and Standards. Technical Report, AIDE, Adaptive Integrated Driver-vehicle InterfacE.

Gietelink, O., Ploeg, J., De Schutter, B., Verhaegen, M., 2006. Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations. Veh. Syst. Dyn. 44 (7), 569–590, July 2006.

INTERACTIVE Project, 2013. Available at <http://www.interactive-ip.eu/project.html>.

Jianqiang, W., Shengbo, L., Xiaoyu, H., Keqiang, L., 2010. Driving simulation platform applied to develop driving assistance systems. IET Intell. Transp. Syst. 4 (2), 121–127.

Kaptein, N., Theeuwes, J., Van Der Horst, R., 1996. Driving simulator validity: Some considerations. Transp. Res. Rec. J. Transp. Res. Board 1550, 30–36.

Lang, B., Parkes, A.M., Cotter, S., Robbins, R., Diels, C., HIT, M.P., et al., 2007. Benchmarking and classification of CBT tools for driver training. TRAIN-ALL.

Liu, Z.Q., 1983. A Study of Washout Filters for a Simulator Motion Base. University of Toronto.

Paul, A. Sanz, J.M., Gago, C., García, E., Díez, J.L., Blanco, R., et al., 2009. Risk analysis of road tunnel using an advanced driving simulator to assess the influence of structural parameters in tunnel safety. DSC Europe 2009, Montecarlo.

Paul, A., Baquero, R., Díez, J.L., Blanco, R., 2008. Analysis of integrated warning strategies for ADAS systems through high performance driving simulator. DSC Europe 2008, Monaco.

Peräaho, M., Keskinen, E., Hatakka, M., 2003. Driver competence in a hierarchical perspective; implications for driver education. Report to Swedish Road Administration.

Slob, J.J., 2008. State-of-the-Art driving simulators, a literature survey. DCT Report, 107.

Thomke, S.H., 1998. Simulation, learning and R&D performance: evidence from automotive development. Res. Policy 27 (1), 55–74.

SUBCHAPTER 10.2

# Traffic Simulation

**Javier J. Sánchez-Medina[1], Rafael Arnay[2], Antonio Artuñedo[3], Sergio Campos-Cordobés[4] and Jorge Villagra[3]**

[1]Universidad Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain
[2]Universidad de La Laguna, Santa Cruz de Tenerife, Spain
[3]CSIC, Madrid, Spain
[4]TECNALIA, Bizkaia, Spain

## 10.2.1 WHAT IS TRAFFIC SIMULATION AND WHY IS IT NEEDED

Traffic simulation has been a very active field in Intelligent Transportation Systems since the 1980s. But what are we talking about when we say traffic simulation? It has two parts, traffic and simulation.

Traffic, in this context is a collective word referring to a collection of transportation means operating simultaneously in a defined bounded geographical area. That may be a city, a metropolis, a number of interconnected cities, a region, etc. In that geographical circumscription we have a number of possible mobility options. In the context of this book, we are primarily talking about terrestrial transportation means.

Also, each application may restrict the range of transportation means or modes. For example it may be considered composed of only cars, or of all kinds of private vehicles, even man-powered like bicycles, or it can also be expanded to pedestrians, cablecars, etc. The multimodality level of a particular traffic simulation heavily depends on the application.

Obviously, the more modes that are included in a particular traffic simulation, the more complex it may be in terms both of mathematical model definition and algorithmic implementation.

The other part of it is the word simulation. A simulation is the recreation of a real-world phenomenon through the instantiation of a previously defined model of it. In other words, we create a mathematical model abstracting a natural phenomenon, then we carefully initialize all the variables involved, and we "run it" with the purpose of getting a reasonably accurate idea of evolution, generally across time, of that phenomenon, paying attention to some aspect of it, like maybe occupation, density, traffic volume, average speed, or many others.

Each simulation tries to answer the question "What would happen if" in the real world. That is very important and useful for many reasons. For example, thanks to the power of modern computers, it allows the evaluation of a number of alternatives and their outputs, before risking the implementation of any of them in the real world. Also, it may permit not just numerous setups, but also extreme case ones. For example, a researcher may simulate an emergency response in case of a dramatic event, that hopefully will never happen in real world.

Having said all of that, before simulations, we need models and that is the hardest part of it. There has to be a previous model definition, including all the necessary elements and concepts. That model may have probably passed through some validation process where it needs to be evidenciated that it is accurate enough for its planned application.

That famous George Box quote says: "All models are wrong, but some are useful." A model is a formal representation of a natural phenomenon. In principle, every phenomenon will always be too complex to be completely accurately represented. In other words, as soon as we create a representation, we are stepping down from absolute accuracy. Furthermore, we do not want absolute accuracy. There will always be a balance between accuracy and performance. Sometimes a model needs to be primarily fast, because it will be used in real-time applications. Some other times it may be very slow, even meant to be run only once, as much as it is very accurate.

All of that can be quickly extended to traffic. Any vehicular traffic situation is such a complex and random process, especially because of the human intervention in most of them. We definitely need to equip ourselves with models to describe and analyze that process and also to make future forecasts.

If we look at how traffic modeling and simulation has evolved through history, in the very first stages it was about creating mathematical models, generally coming from physics and fluids dynamics, but after a few years, especially after the exponential growth of computing power, more ambitious and detailed simulation paradigms were explored.

That physics-based approach is the so-called macroscopic traffic modeling. Traffic was understood as a continuum. That kind of model was quite efficient for several tasks and applications, but very soon a wildly different approach was presented, where vehicles were modeled individually with more or less the same level of detail, and more importantly, the interaction between them was also modeled. That discrete way of modeling traffic was called microscopic traffic modeling.

Even from the 1960s the very first traffic simulators where designed following one of the two philosophical approaches to traffic modeling. At some point, some hybrid models where developed incorporating the strengths and leaving out the weaknesses of both of them. That was when Mesoscopic models were defined and applied.

Back in 1956, one can find in Wilkinson (1956) one of the first publications that could be considered to be on traffic simulation modeling, where trunk traffic was modeled as a random Gaussian process, mainly described by mean and variance, assuming some level of noise.

We have to wait until the 1960s to see the first works on computer-aided simulations of traffic, although these were still rudimentary. For example, we have Stark (1962), where a custom made very simplistic vehicle simulation of nine blocks was published. It is interesting to note that at that time the simulation visualization was done by taking pictures of an oscilloscope screen every quarter of a second.

Another interesting work is Shumate and Dirksen (1965) where a programming language is presented (SIMCAR), basically for designing highways and simulating vehicles and even different driving styles, again it is rudimentary and with very little scalability.

Within microscopic traffic simulation we have the so-called car-following models. In May and Harmut (1967) some car-following models are evaluated and compared. Car following models could be referred to as the ancestors of modern microsimulation, where the front to bumper distance is modeled, showing quite realistic effects like stop and go waves.

Later, in the 1970s, we got the first simulation frameworks like FREFLO (Payne, 1979; Mikhalkin, 1972) where freeway traffic is modeled mathematically in terms of variables like density, space-mean-speed, and flow rate. Another example of macroscopic simulation was SATURN-a (Hall and Willumsen, 1980), which has a relative that is still around.

Also in the 1970s, the first microscopic simulation frameworks such as TEXAS (Rioux and Lee, 1977) showed up. Traffic simulation was prevented from simulated bit zones because of the cost of the computing power they had. They are very slow in comparison to microscopic simulators.

It was in the 1980s and especially in the 1990s when there was a real revolution regarding traffic simulators, mainly because of the exponential reduction of computing hardware costs, making powerful computers more and more affordable. Also parallel computing advances and IBM's PC standardization helped a lot to a make computing more and more available for increasingly more complex traffic simulation frameworks.

Talking about the present, there are a few live challenges for traffic simulation these days. The first one is multimodality. Every traffic simulation framework needs to support multimodality, which is incorporating a wide palette of transportation means models, including pedestrians, bicycles, electric vehicles, and more. That is very important because mobility is gradually tending to become more and more diverse regarding transportation means, at least these days when some paradigm shift revolutions are in place, like transportation electrification or driverless automation. Depending on the application, every different transportation means may need to be modeled. An electric car has a quite different dynamic behavior than a gas powered one.

Another important challenge is about real-time managing of traffic. A usual demand from traffic managers is real-time monitoring of the current state of traffic. To do so, we need simulators to be computationally efficient and scalable and a part of the scientific community is devoted to that goal.

Optimization is a very important topic. With traffic being so complex, it can be hardly managed with analytical tools. Instead, it usually needs nondeterministic optimization techniques, i.e., Genetic Algorithms (Sanchez-Medina, 2008; Sanchez-Medina, 2010).

Finally, it is worthy mentioning two of the most exciting things that have happened regarding traffic simulation and modeling in recent years. The first one is the influence Open Data is having on the development of this area. In the last years more and more local governments in the World have decided to put their data in open access. This is extremely interesting, because it allows researchers all around the globe to access and use even real-time traffic data feeds for their research, scaling the number of publications and discoveries.

The second very exciting event is the open simulation shift that has been led by SUMO (Krajzewicz, 2002). In Section 10.2.4 we will talk longer about it, but it is worth advancing that this is truly game-changing in a so far quite closed traffic simulation framework business. The SUMO developers' community is a marvelous example of what are the benefits of free software: constant updates, collaboratively developed plugins, and open developers and practitioners fora.

## 10.2.2 CLASSIC TRAFFIC SIMULATION PARADIGMS

### 10.2.2.1 Macroscopic Simulation

Macroscopic simulation is the branch of traffic simulation that relies on the so-called macroscopic models. The "Macro" part in it tries to express that this kind of model views traffic from a distance, considering it as a continuum or fluid. Therefore, the objective of this kind of simulation is the spatiotemporal representation of mainly three real variables: volume $q(x,t)$, speed $u(x,t)$, and density $k(x,t)$. Volume is regarding the number of vehicles passing through a specific point in space. Speed has to do with the space traversed by a particular vehicle in a fixed period of time. Finally, density has to do with the number of vehicles occupying a fixed area (a lane, a multiple lane street, etc.). Formally speaking, the basic formula in macroscopic modeling announced by Gerlough and Matthew (1975) is the conservation or continuity equation:

$$\frac{\partial q}{\partial x} + \frac{\partial k}{\partial t} = 0 \qquad (10.2.1)$$

This kind of formulation is inherited from hydrodynamics. It basically means that, if there are no inputs or outputs, the number of vehicles must remain the same across the pipe (highway, street, etc.).

That very simple equation is considering an equilibrium situation without effects like congestion formation, stop and go waves, etc.

Therefore, it was modified by Payne (1979) in the early 1970s, resulting in the following:

$$\frac{\partial k}{\partial t} + u\frac{\partial q}{\partial x} = \frac{1}{T}[u_e(k) - u] - \frac{\nu}{k}\frac{\partial k}{\partial x} \tag{10.2.2}$$

In a summarized fashion, $u$ represents the speed—density relationship that can be as announced by May and Harmut (1967) as follows:

$$u = u_t\left[1 - \left(\frac{k}{k_{\text{jam}}}\right)^{\alpha}\right]^{\beta} \tag{10.2.3}$$

In Eq. (10.2.2) it is intended to reflex two very important elements on the nonequilibrium traffic flow effect, namely acceleration and inertia. The right side of Eq. (10.2.2) has two parts. The first one reflects the action of the driver adjusting speed aiming at the equilibrium speed. In that part, $T$ is the so-called relaxation time, and $\nu$ means the anticipation parameter.

The second half of the right-hand side of Eq. (10.2.2) reflects how drivers' reactions affect downstream traffic conditions.

According to Barceló (2010), Payne's model seems to present accuracy problems particularly in dense traffic in on-ramps or lane drops.

Microscopic models and microsimulators were the first in the field and consequently, some of the more venerable traffic simulation frameworks are based on macroscopic models, at least in origin. For example, FREFLO (Payne, 1979; Mikhalkin, 1972) and SATURN-a (Hall and Willumsen, 1980) were some of the first microsimulators back in the 1970s, mainly for highway traffic simulation. SATURN is still around these days.

Some other macroscopic traffic simulation frameworks are TRANSYT-7F (Wallace, 1998) or METANET (Spiliopoulou, 2015).

Nowadays, macroscopic simulation is itself is becoming less and less common for a simple reason: microscopic simulation is more accurate and even when in general it is much heavier in terms of computing power, it is also true that computing power is becoming more and more available at reasonable costs.

However, for some applications, or in combination with other microscopic models (mesoscopic simulation), they are still in use and one can find relevant and interesting literature on them. For example, a new generation of macroscopic simulators are based on a new paradigm. They are the gas-kinetic (GKT) traffic flow models, for example, the one used by Delis (2015) for modeling traffic flow with adaptive cruise control.

#### 10.2.2.2 Microscopic Simulation

Microscopic simulation models (multiagents) simulate the movement of individual vehicles based on vehicle traceability and on theories of change of lane. Typically, vehicles enter the transport network using a distribution probabilistic of arrivals (a stochastic process) and are followed during their passage through the network in small time intervals (e.g., one second or a fraction of a second). After entering each vehicle, each is normally assigned a destination, a type of vehicle, and a type of driver. These models are effective in evaluating traffic congestion, complex geometric configurations, or the impact of transport improvements that are beyond the limitations of other types of tools. However, these models have a high cost in time, money, and can be difficult to calibrate.

The definition and implementation of such simulators, involves knowledge of different scientific and engineering fields:

- *Microscopic Modeling of Traffic itself*. This approach constitutes the basis for traffic flow theory (e.g., Herman and Potts, 1900). As we will see later, there exist many advanced available software, both open and commercial, that are able to manage accurate and fast simulations of large geographical areas.
- *Computational Physics*. Experience of the adoption of simple and very fast models of physical processes, with lower simulation computing requirements. While physics models manage particles, here we manage people with a similar order of elements (e.g., regional and municipality microscopic simulation). To establish a compromise between model interaction detail, simulation/interaction speed, and computational requirements is mandatory.
- *Microscopic Behavioral Modeling of Demand/Agent-Based Modeling*. We can find as many definitions of "agent" as researchers, some of them are "a discrete entity with its own goals and behaviors, with capability to adapt and modify its behavior" (Macal and North, 2005) or "anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators" (Russell and Norvig 2002). As we will discuss later, these models combine elements from game theory, complex systems, emergence, computational sociology, multi-agent systems, and evolutionary programming. Agent-based modeling uses simple rules that can result in different sorts of complex behavior. The key point is the autonomously, emergence, and complexity. Samples of these models are cellular automata

models (Nagel and Schreckenberg, 1992) or the gravity model (Wilson, 1971).

- *Complex Adaptive Systems/Coevolutionary Algorithms*. Traveling and moving by means of any transport mode, from public transport, car, or walking, involves fundamentally game-theoretic reasoning: individual decisions are evaluated in specific interaction scenarios (e.g., congestion, innovative shared transport, activity grouping) resulting from decision made by the collective rather than in isolation. Different gaming strategies as Nash-equilibrium approaches, where individual decision maximizes the gain of the other, and other such as dominant strategies or mixed strategies have been deployed in transport analysis have been developed in transport assignment from the mid of the last century. As we mentioned before, metaheuristics methods combined with equilibrium logic implements schedule coevolutionary search schemes.

Some of the most well-known microsimulators are HUTSIM (Kosonen, 1999, 1996), VISSIM (Park et al., 2003), CORSIM (Owen, 2000), SESIM (Flood, 2008), AIMSUN (AIMSUN, 2017), Transims (Rilett and Kyu-Ok, 2001), and Cube Dynasim (Citilabs, 2017). The MATSIM (Balmer, 2009) are developed under the multiagent paradigm, which is of great relevance if the emergence of traffic behavior under certain traffic demands is the subject of research. SUMO (Krajzewicz, 2002, 2006) is an open source traffic simulator which is based on the Gipps-model extension (Krauß, 1998) and, more recently, also on the IDM model.

#### 10.2.2.3 Mesoscopic Simulation

These models combine the properties of macroscopic and microscopic models. As in microscopic models, the traffic unit is the individual vehicle. Its movement, however, follows the simplification performed by the macroscopic models, and is determined by the average speed of the route. Dynamic speeds or volume ratios are not considered. Therefore, mesoscopic models provide less fidelity than microscopic models, but are superior to typical analysis techniques.

Some of the relevant meso-simulators are DYNAMIT-P-X (Ben-Akiva, 2002), DYNASMART-P-X (Mahmassani and Jayakrishnan, 1991), and MesoTS (Meng, 2012). In any case, most of the previously mentioned microscopic simulators provide hybrid (micro–meso) integrated solutions.

## 10.2.3 SOME (TRADITIONAL) SIMULATION FRAMEWORKS

### 10.2.3.1 CORSIM

Corridor Simulator (CORSIM) (Halati, 1997) is a microscopic simulation framework developed by the Federal Highway Administration (FHWA) in the United States. Therefore, it is a sort of standard simulation framework for many research groups, especially in that country when they have to deal with the administration. Historically CORSIM is the evolution of two older models: FRESIM (FREeway SIMulation) and NETSIM (NETwork SIMulation). FRESIM (Halati, 1990) is a microscopic simulator for highways traffic. NETSIM (Rathi and Santiago, 1990) does the same but for urban traffic.

FRESIM's antecesor is INtegrated TRAffic Simulation (INTRAS) (Wicks and Andrews, 1980), a microscopic traffic simulator from the early 1980s.

They main characteristic of CORSIM is that it is fundamentally based on Car-Following models, also known as time-continuous models. Car-Following models are defined through differential equations to describe the position and speed of every vehicle. The aim of this kind of model is to approximate the bump to bump distance ($s_\alpha$) between two consecutive cars like in Eq. 10.2.4:

$$s_\alpha = x_{\alpha-1} - x_\alpha - l_{\mathrm{alpha}-1} \tag{10.2.4}$$

$x_{\alpha-1}$ is the position of the vehicle in front (leader), $x_\alpha$ the position of the current vehicle, and $l_{\mathrm{alpha}-1}$ the length of the leader vehicle.

CORSIM comes along with Traffic Software Integrated System (TSIS), a MS Windows-based application that takes care of the visualization layer of CORSIM. That is a very important addition because as both a scientist and a manager, it is extremely useful to get a visual of what is happening in a simulation to truly understand the possible effects of every setup on real-world traffic. TSIS includes a rich set of useful tools from the traffic network design and simulation to its analysis.

There are hundreds of works based on CORSIM in literature for both freeway and urban traffic. A key element with CORSIM is its calibration. The CORSIM framework requires the calibration of a big number of variables, before its exploitation at a particular case. There are parameters regarding the drivers, the vehicles, the roads, etc. Therefore, there are literarily hundreds of works on CORSIM calibration methods, using Artificial Intelligence techniques like in Cobos

(2016), Statistical Techniques (Paz, 2015), Bayesian methods (Bayarri, 2004), and more.

Currently, CORSIM is being maintained by the University of Florida's McTrans Center. Here are some of the listed capabilities of CORSIM in McTrans website:

- Public presentation and demonstration.
- Freeway and surface street interchanges.
- Signal timing and signal coordination.
- Diverging diamond interchanges (DDI).
- Land use traffic impact studies and access management studies.
- Emergency vehicles and signal preemption.
- Freeway weaving sections, lane adds, and lane drops.
- Bus stations, bus routes, carpools, and taxis.
- Ramp metering and High Occupancy Vehicles (HOV) lanes.
- High occupancy toll (HOT) lanes.
- Unsignalized intersections and signal warrants.
- Two-lane highways with passing and no-passing zones.
- Incident detection and management.
- Queuing studies involving turn pockets and queue blockage.
- Toll plazas and truck weigh stations.
- Origin-destination traffic flow patterns.
- Traffic assignment for surface streets.
- Statistical output postprocessing.
- Adaptive cruise control.
- Importing and exporting to TRANSYT-7FTM (Wallace, 1998) (TRAffic Network StudY Tool, version 7 F), the descendant of TRANSYT, developed by the Transport Research Laboratory in the U.K. in 1969.

#### 10.2.3.2 MATSIM

The conventional and widely extended trip-based model of travel demand forecasting has been the reference model in urban mobility planning for the last decades. Nevertheless, this model was conceived for evaluating the impact of infrastructure investment options at the strategic planning stage. In fact, this model is not able to deal with real day-to-day issues such as time-dependent and spatial neighbourhood effects or collective decisions.

On the other hand, from a social and behavioral explanation of mobility, we have the following principles:

- The travel demand is conducted by the specific needs and wish of the individual.
- Social relationships influence on displacements and mobility habits and patterns.
- There exists some relevant constraints around travel: spatial, temporal, collective, facilities, and transportation accessibility barriers, among many others.
- While other models do not imply any kind of sequence or dependence on travel, it is needed to reflect the sequencing of activities in time and space.

The activity-based model (Rasouli, 2016; Castiglione, 2015) gives response to these questions and adds to the classic four stages model questions (mode, route, location, and timing), the following decisions:

- *Activity type choice*: Which activity should I do?
- *Activity chain choice*: In which order should I do my activities?
- *Activity starting time choice*: When should I start the activity?
- *Activity duration choice*: How long should I do the activity?
- *Group composition choice*: Who should I take along in the activity?

Summarizing, this approach or model is aimed at identifying and predicting for how long and with whom an activity is conducted, additionally to the classic parameters.

The "Multiagent transport simulation toolkit" (MATSim) simulation (Horni, 2016), is based on the agent concept and adopts the just mentioned activity-based approach. Each traveler, here the concept of driver is extended to any person traveling by any transport mode, of the target population is modeled as an individual agent able to take independent decisions.

The simulation consists of two sides mutually coupled:

- On the demand side, agents predefine a preliminary and independent plan that specifies its intentions during the time period under analysis. This plan is the output of an activity-based model that comprises route choice among other stages as seen, depending on expected network, public transport, or road, conditions.
- On the supply side, a mobility (including traffic flow) simulation or real operation takes place, executing all the plans of the predefined agents.

A learning mechanism for the agents is implemented by the iterative coupling of demand, defined as the agent generation and supply, obtained by traffic flow simulation. Basically, it takes the candidate agent's plan, evaluates their performance and adopts the best options (including metaheuristics methods, to evolve the set of solutions, avoiding local optimums).

The control flow or process of MATSim is composed by the following iterative activities:

- *Initial Demand.* MATSim requires a synthetic population of agents, each with individual transport-related attributes and daily activity plans, being representative of the population. These parameters are managed for each activity instance.
- *Simulation.* The traffic flow simulation runs the expected plans, emulating the interactions between agents and transport system according to its characteristics and constraints.
- *Scoring.* MATSim uses a simple utility-based approach to calculate a plan score, with positive values for time dedicated to perform "productive" activities and negative for traveling and delays on displacement activity locations.
- *Replanning.* Sequence of configurable algorithms that iterate on the population plans, Usually, adopting population-based heuristics, the algorithm considers sets of individuals (population) where each one represents a solution to the problem and evolves the set leading to improvements for average plan scores and travel times.

MATSim requires complete data containers or inputs to perform any transit simulation:

- A simulation configuration (e.g., parameters, modules to use for each step, iterations);
- A multimodal network and transit schedules (e.g., public transport agencies, lines, services, and vehicle/rolling stock characteristics and in the other side, the navigable road network);
- Time-dependent network attributes (manages parameters per segment, such as free speed, lanes, and capacity, that can change during the day, due to incidences or dynamic traffic adaptive solutions);
- Mobility plans (defines subpopulations, person attributes, their mobility plans, and transport demand of the analysis population);
- Some facilities that specify where the agents realizes the different activities; and
- Counts, taken from real operation that allows to compare and calibrate the simulation and specific scenarios.

MATSIM adopts a modular concept, in a broad sense of the word, referring to components at different levels, from functions, components or third party extension tools and frameworks. In any case, we can substitute a module by a specific functionality. Some relevant extensions currently available cover: freight management, car sharing, joint trips, parking, electric vehicles, pricing, emission calculation, travel time calculation, advanced analysis, multimodal transport, traffic signaling, among many other.

MATSim is written in the Java programming language and distributed under the GNU Public License (GPL), being available for download, use, and extension. Extensive documentation is accessible for developers, including specification of key-aspects of MATSim, configuration and underlying theory, guidelines, details for most of the extension packages, and data/samples for testing.

### 10.2.3.3 AIMSUM2

Urban congestion has a high impact on pollutant and energy consumption KPIs. Cities and their citizens have a fundamental role to play, since they concentrate the largest number of vehicles and the greatest problems of congestion, generating in these urban centers much of the total emissions of the planet. There is ample room for improvement, through a more intelligent use of means of transport and the integration of advanced technology as support for the improvement of mobility services.

By Active Transportation and Demand Management (ATDM) (US DoT, 2017), we integrated different strategies to provide solutions for congestion by combining public policy and private sector innovation to encourage people to change their transport habits, increase the share of sustainable mobility, prevent breakdown conditions, improve safety, and maximize transport efficiency and performance in general.

By definition ATDM implementation strategies fall under three major categories:

- *Active Demand Management (ADM)*, by using information and technology to dynamically manage demand, including redistributing travel, or reducing vehicle trips by influencing mode choice, adoption of more sustainable transport modes;
- *Active Traffic Management (ATM)*, that tries to dynamically manage recurrent and nonrecurrent congestion based on current and predicted traffic ; and finally

- *Active Parking Management (APM)*, parking facilities management to optimize performance and utilization of those facilities while influencing travel behavior.

All these strategies are based on estimations of traveler behavior; external factors, effects, and effectiveness of the actions themselves are subject to a high uncertainty.

AIMSUN (AIMSUN, 2017) is a widely used commercial transport modeling software, developed and marketed by Transport Simulation Systems (TSS). It integrates microscopic and mesoscopic components allowing dynamic simulations. AIMSUN provides the tools to carry out traffic assessment, in terms of environmental impact, capacity, or safety analysis, of some of the main actions to implement the previously mentioned categories:

- Feasibility studies for High Occupancy Vehicle (HOV) and High Occupancy Toll (HOT) lanes (ADM).
- Impact analysis of infrastructure design such as highway corridors (ADM) (Silva, 2015).
- Toll and road pricing (ADM).
- Evaluation of Variable Speed policies and other Intelligent Transportation Systems (ITS) (ADM).
- Bus Rapid Transit (BRT) schemes (ADM).
- Workzone management (ATM).
- Signal control plan optimization and adaptive control evaluation (ATM).
- Assessment and optimization of Transit Signal Priority (TSP) (ATM).
- Proactive Traffic Management, evaluating in real time the effect of decisions.

The simulator is highly configurable and extensible with new features and capabilities. By default, it can manage different traffic networks, demand modeling as flows at sections or O/D matrices, etc. but also can be extended by programming, enabling the modification of the behavioral models and the addition of new functionalities to the application. A detailed description of such parameters and extension capabilities can be found in the tool's manual (AIMSUN, 2014).

The commercial references of its application to urban and interurban transport planning are large. Specifically, for C-ITS deployment studies (Aramrattana and Maytheewat, 2016), the execution in combination with network simulators is mandatory (e.g., OMNeT++ simulator), and we can find an exhaustive list of such experiences in Segata (2014).

More recently, AIMSUN is being used to evaluate deployments of new mobility solutions, such as electromobility or autonomous driving vehicles. In the case of EVs, an extension of AIMSUN was implemented in the context of FP7 EMERALD project (Boero, 2017) to evaluate and optimize the impact of recharging infrastructure design in urban and interurban traffic management. This project supported the development of new transport models and algorithms (specifically oriented to FEVs), evaluation of intelligent transport systems and cooperative systems (V2I/I2V), and control plan optimization. Other additions were the third dimension in the maps and several types of behavior driven and consumption equations for each FEV type. In the context of autonomous vehicles, the project FLOURISH managed by the UK government have adopted AIMSUN to support the assessment of different scenarios from motorway to urban use; in this case the focus is on the user, their demands, expectations, and challenges for specific collectives, such as elderly people.

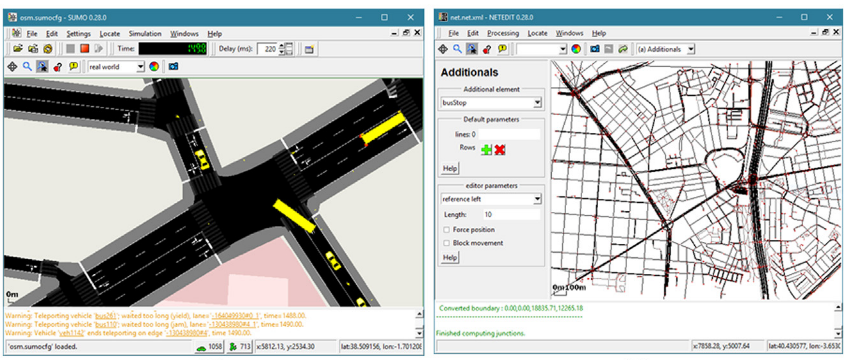
### 10.2.4 OPEN TRAFFIC SIMULATION: SUMO

"Simulation of Urban Mobility" (SUMO) (Krajzewicz, 2012) is a microscopic, multimodal, space-continuous, and time-discrete road traffic simulator. It is open source software licensed under the GNU GPL (General Public License) that is mainly developed by the German Aerospace Center (DLR). The development of this simulation tool started in 2000 having in mind portability and extensibility as main design criteria. Moreover, the need for handling large road networks required the taking into account of the execution speed and memory footprint as further guidelines.

The simulation scenario for SUMO has to be defined through a road network and a traffic demand. The road network can be defined either manually by generating XML files describing the network or by importing the network from other formats such as: OpenStreetMap (OSM), PTV VISUM and VISSIM, OpenDRIVE, MATsim, ArcView, etc. Besides that, SUMO is able to generate random networks under some rules (random-networks, spider-networks, and grid-networks).

The traffic demand can be defined in some different ways depending in the available input data: trip definitions, flow definitions, randomization, OD-matrices (VISUM/VISION/VISSIM formats), etc. Also for traffic demand XML files are used. SUMO supports different vehicle types such as motorcycles, trucks, buses, bicycles, or railways. Pedestrians are also supported. Moreover, some useful tools are provided for traffic demand modeling. For example, ActivityGen allows generating traffic demand from a description of the population data in the net through some parameters such as population's age brackets, school locations, bus lines, or work hours. Within the simulation, the vehicle movements are based on the longitudinal and lateral models. Both models can be chosen for each vehicle type among some that are already implemented in SUMO.

The outputs that SUMO can generate in each simulation include a number of different data: vehicle-based information over time (vehicle positions, pollutant emission values based on HBEFA database), lane/edge-based network performance information (vehicular noise emission based on HARMONOISE model), simulated detectors, and traffic lights information, among others. A useful graphical user interface (GUI) is also included in SUMO package. Besides making easier the basic use of the simulator, this GUI is very useful for monitoring the evolution of the simulation through a 2D representation as well as diverse aspects of the simulation (e.g., lane and vehicle coloring based on current occupancy or, CO2/CO/NOx/PMx/HC emissions, noise emission, average speed, etc.) at runtime (Fig. 10.2.1).




(A) (B)

**Figure 10.2.1** (A) SUMO-GUI; (B) NetEdit tool.

Due to its flexibility, SUMO is currently one of the most used simulation frameworks in the research, academic, and industry sectors. Among its applications, it is worth mentioning traffic forecasting, traffic management evaluation, route choice and re-routing evaluation, logistics, and traffic surveillance methods.

The possibility of adding extensions makes SUMO extend to new applications. One of the main extensions is "Traffic Control Interface" (TraCI). This interface enables the online interaction between SUMO and external applications. It allows to manipulate the behavior of simulated objects and to retrieve values at runtime. TraCI is available in different programming languages: C++, Java, Python, and Matlab. This interface is commonly used for providing the simulation platform with new functionalities such as external 3D visualization, traffic lights control simulation, or couplings with communication networks simulators (ns3, JiST/SWANS, or OMNeT++). For example, TraCI was extensively used in iTETRIS project (iTETRIS, 2017) whose goal is to couple SUMO with a communication network simulator using a middleware (iCS) for V2X applications. This work has been extended within the COLOMBO project (COLOMBO, 2017). In addition, SUMO has been used in further European projects: AMITRAN (AMITRAN, 2017), DRIVE C2X (DRIVE, 2017), among others; which shows the continuously growing use of this simulation framework for different mobility-related purposes.

Besides TraCI, SUMO package includes a great number of tools for different purposes: traffic assignment, dealing with real life induction loop data, traffic analysis, importing data, traffic light systems, trip generation, graphical evaluation of SUMO-outputs, working with sumo output files, etc. Most of them are Python scripts that help to perform different tasks. Some remarkable tools are: osmWebWizard (for quickly creating simulation scenario from a web browser by selecting a geographic region on a OSM map and specifying random traffic demand), sumolib (it is a set of modules for working with SUMO networks), tools for making easier parsing and visualizing simulation results, and conversion tools for data analysis.

The extensive use of this simulator makes it well maintained and constantly developing, fixing bugs and adding new features in each new update. In the last versions some new remarkable features have been added. Some examples are the availability of the NetEdit tool (since version 0.25.0) for graphical network creation and edition, and the inclusion of a mesoscopic model (available since version 0.26.0).

### 10.2.5 FUTURE TRENDS AND HOPES

The purpose of this section is not to divine the future development of traffic modeling and simulation, but to comment on some of the trends that seem to be present in the years to come regarding this topic.

First, we must say that simulation will likely be more and more online in the next decades. The ongoing revolution in the fields of Big Data and Data Stream Mining will possibility deeply affect traffic simulation, in particular when applied to the traffic managing, decision making, emergency managing, and advanced travelers information systems (ATIS).

The Internet of Things, sensor networks, and pervasive computing, including personal smart devices, are all expanding areas; they are likely to be useful sources of information on the current traffic situation for every transport network. That amazing torrent of information needs to be exploited. It is very urgent that data stream mining, which is the brand in Data Mining and Big Data thought to cope with live data, serves updated models and predictions on traffic states.

With the same purpose, simulation frameworks will need to be configured to feed from online information and to run in real time. Some simulation platforms, like SUMO, will need to improve its performance to guarantee the real-time restrictions, maybe through parallelization and/or rewriting its libraries in faster programming languages (Romero-Santana, 2017).

Also, another hot topic where traffic simulation is playing a mayor role is on the Connected Vehicle move. It seems quite clear that driverless mobility is transiting from the autonomous car paradigm, where the intelligent vehicle is equipped with enough sensors and computation to accurately perceive its environment, calculate trajectories, predict surrounding vehicles' intentions, etc., to a connected car paradigm, where perception, 3D reconstructions, etc. can be shared with the infrastructure and all the other connected vehicles, exponentially extending the "safety bubble" of each one of them. In this new connected setup, at some point future forecasts will be required to predict traffic state, to dynamically propose alternative routes, to warn of possible future hazards, etc. Fast online traffic simulation will definitely play an important role there.

Multimodality is also a greatly challenging topic for traffic modeling and simulation. New kinds of vehicles, with new dynamic behavior, such as GPL, electric, or hybrid powered platforms, will need to be considered in order to accurately simulate their behavior. Also, driverless driving is challenging since it has been proven that driverless cars will behave quite differently (usually more conservatively) than human drivers (Kaber and Endsley, 2004). Finally, many cities in the world are working hard to foster greener transportation modes into the system, with the very interesting benefits for both traffic management and quality of life of citizens it may bring along (Rietveld, 2000).

One final piece to be incorporated in a near future in simulation frameworks are Unmanned Aerial Vehicles (UAV). There are already studies and proof of interest coming from both academia and industry on incorporating UAVs for surveillance, fast delivery, and other applications (Shim, 2005; Coifman, 2004).

## REFERENCES


AIMSUN, 2014. Aimsun 8 Users' Manual, TSS-Transport Simulation Systems, 2014.

AIMSUN, 2017. AIMSUN – the Integrated Transport Modelling Software, <http://www.aimsun.com/site/>.

AMITRAN, 2017. AMITRAN Project. <http://www.amitran.eu>.

Aramrattana, M., 2016. Modelling and simulation for evaluation of cooperative intelligent transport system functions.

Balmer, M., Rieser, M., Meister, K., Charypar, D., Lefebvre, N., Nagel, K., 2009. MATSim-T: architecture and simulation times.

Barceló, J., 2010. Models, traffic models, simulation, and traffic simulation. Fundamentals of Traffic Simulation. Springer, New York, pp. 1–62.

Bayarri, M.J., James, O.B., Molina, German, Rouphail, Nagui M., Sacks, Jerome, 2004. Assessing uncertainties in traffic simulation: a key component in model calibration and validation. Transp. Res. Rec.32–40.

Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H.N., Mishalani, R., 2002. Real time simulation of traffic demand-supply interactions within DynaMIT. Transportation and Network Analysis: Current Trends. Springer, pp. 19–36.

Boero, M., et al., EMERALD- Energy ManagEment and RechArging for Efficient eLectric Car Driving Project, vol. 2017. <http://www.fp7-emerald.eu/> .

Castiglione, J., Bradley, M., Gliebe, J., 2015. Activity-based travel demand models: a primer.

Citilabs, 2017. Software for the modeling of transportation systems, <http://www.citilabs.com/>.

Cobos, C., Erazo, C., Luna, J., et al., 2016. Multi-objective memetic algorithm based on NSGA-II and simulated annealing for calibrating CORSIM micro-simulation models of vehicular traffic flow.

Coifman, B., McCord, M., Mishalani, M., Redmill, K., 2004. Surface transportation surveillance from unmanned aerial vehicles. In: Paper Presented at Proceedings of the 83rd Annual Meeting of the Transportation Research Board.

COLOMBO, 2017. COLOMBO Project. <http://colombo-fp7.eu>.

Delis, A.I., Nikolos, I.K., Papageorgiou, M., 2015. Macroscopic traffic flow modeling with adaptive cruise control: development and numerical solution. Comput. Math. Appl. 70, 1921–1947.

DRIVE, 2017. DRIVE C2X Project. <http://www.drive-c2x.eu>.

Flood, L., 2008. SESIM: a Swedish micro-simulation model. Simulating an Ageing Population: A Microsimulation Approach Applied to Sweden. Emerald Group Publishing Limited, pp. 55–83. Chapter 3.

Gerlough, D., Huber, M., 1975. Transportation research board special report 165: traffic flow theory: a monograph. Transp. Res. Board.

Halati, A., Torres, J., Mikhalkin, B., 1990. Freeway simulation model enhancement and integration—FRESIM Technical Report. Federal Highway Administration, Report No.DTFH61-85-C-00094.

Halati, A., Lieu, H., Walker, S., 1997. CORSIM- Corridor traffic simulation model.

Hall, M., Willumsen, L.G., 1980. SATURN-a simulation-assignment model for the evaluation of traffic management schemes. Traffic Eng. Control21.
Herman, R., Potts, R.B., 1900. Single lane traffic theory and experiment.
Horni, Andreas, Nagel, Kai, Axhausen, Kay W., 2016. The multi-agent transport simulation MATSim. Ubiquity, London9.
iTETRIS, 2017. ITETRIS Platform. <http://www.ict-itetris.eu>.
Kaber, David B., Endsley, Mica R., 2004. The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. Theor. Issues Ergon. Sci. 5, 113–153.
Kosonen, I., 1999. HUTSIM-urban traffic simulation and control model: principles and applications, vol. 100.
Kosonen, I., 1996. HUTSIM: simulation tool for traffic signal control planning.
Krajzewicz, D., Bonert, M., Wagner, P., 2006. The open source traffic simulation package SUMO. RoboCup.
Krajzewicz, D., Hertkorn, G., Rössel, C., Wagner, P., 2002. SUMO (Simulation of Urban MObility)-an open-source traffic simulation. In: Paper presented at Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002).
Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L., 2012. Recent development and applications of SUMO-simulation of urban mobility. Int. J. Adv. Syst. Measure.5.
Krauß, S., 1998. Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics. D L R - Forschungsberichte.
Macal, C.M., North, M.J., 2005. Tutorial on agent-based modeling and simulation.
Mahmassani, H.S., Jayakrishnan, R., 1991. System performance and user response under real-time information in a congested traffic corridor. Transp. Res. Part A-Policy Pract. 25, 293–307.
May Jr, A.D., Harmut, E.M., 1967. Non-integer car-following models. Highway Res. Rec.
Meng, M., Shao, C., Zeng, J., Dong, C., 2012. A simulation-based dynamic traffic assignment model with combined modes. Promet - Traffic - Traffico 26, 65–73.
Mikhalkin, B., Payne, H.J., Isaksen, L., 1972. Estimation of speed from presence detectors.
Nagel, K., Schreckenberg, M., 1992. A cellular automaton model for freeway traffic. J. de Phys. I 2, 2221–2229.
Owen, L.E., Zhang, Y., Rao, L., McHale, G. 2000. Traffic flow simulation using CORSIM.
Park, B., Schneeberger, J.D., 2003. Microscopic simulation model calibration and validation: case study of vissim simulation model for a coordinated actuated signal system. Transp. Res. Rec.185–192.
Payne, H.J., 1979. FREFLO: a macroscopic simulation model of freeway traffic. Transp. Res. Rec.
Paz, A., Molano, V., Sanchez-Medina, J., 2015. Holistic calibration of microscopic traffic flow models: methodology and real world application studies.
Rasouli, S., 2016. Uncertainty in modeling activity-travel demand in complex urban systems. TRAIL Research School.
Rathi, A.K., Alberto, J.S., 1990. Urban network traffic simulations TRAF-NETSIM program. J. Transp. Eng. 116, 734–743.
Rietveld, P., 2000. Non-motorised modes in transport systems: a multimodal chain perspective for the Netherlands. Transp. Res. Part D Transp. Environ. 5, 31–36.
Rilett, L.R., Kim, Kyu-Ok, 2001. Comparison of TRANSIMS and CORSIM traffic signal simulation modules. Transp. Res. Rec.18–25.

Rioux, T.W., Lee, C.E., 1977. Microscopic traffic simulation package for isolated intersections. Transp. Res. Rec.

Romero-Santana, S., Sanchez-Medina, J.J., Alonso-Gonzalez, I., Sanchez-Rodriguez, D., 2017. SUMO performance comparative analysis: C Vs. Python. In: Paper Presented at In International Conference on Computer Aided Systems Theory, EUROCAST2017 (in press). Springer Berlin Heidelberg. Las Palmas de Gran Canaria. Spain.

Russell, S.J., Norvig, P., 2002. Artificial intelligence: a modern approach (International Edition).

Sanchez-Medina, J.J., Galan-Moreno, M.J., Rubio-Royo, E., 2008. Applying a traffic lights evolutionary optimization technique to a real case: "Las Ramblas" area in Santa Cruz De Tenerife. IEEE Trans. Evol. Comput. 12, 25–40.

Sanchez-Medina, J.J., Galan-Moreno, M.J., Rubio-Royo, E., 2010. Traffic signal optimization in "La Almozara" district in saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing. IEEE Trans. Intell. Transp. Syst. 11, 132–141.

Segata, M., Joerer, S., Bloessl, B., Sommer, C., Dressler, F., Cigno, R.L., 2014. Plexe: a platooning extension for veins. In: Paper Presented at Vehicular Networking Conference (VNC), 2014 IEEE.

Shim, D., Chung, H., Kim, H.J., Sastry, S., 2005. Autonomous exploration in unknown urban environments for unmanned aerial vehicles. In: Paper Presented at AIAA Guidance, Navigation, and Control Conference and Exhibit.

Shumate, R.P., Dirksen, J.R., 1965. A simulation system for study of traffic flow behavior.

Silva, A.B., Mariano,, P., Silva, J.P., 2015. Performance assessment of turbo-roundabouts in corridors. Transp. Res. Proc. 10, 124–133.

Spiliopoulou, A., Papamichail, I., Papageorgiou, M., Tyrinopoulos, I., Chrysoulakis, J., 2015. Macroscopic traffic flow model calibration using different optimization algorithms,. Oper. Res. 17.

Stark, M.C., 1962. Computer simulation of traffic on nine blocks of a city street. Highway Res. Board Bull.

U.S. DoT, 2017.Active Transportation and Demand Management. <http://www.its.dot.gov/research_archives/atdm/index.htm>.

Wallace, C.E., Courage, K.G., Hadi, M.A., Gan, A.C., 1998. TRANSYT-7F user's Guide. Transportation Research Center. University of Florida, Gainesville, Florida.

Wicks, D.A., Andrews, B.J., 1980. Development and testing of INTRAS, a microscopic freeway simulation model. Volume 2: User's Manual. Final Report.

Wilkinson, R.I., 1956. Theories for toll traffic engineering in the U. S. A. Bell Syst. Technol. J. 35, 421–514.

Wilson, A.G., 1971. A family of spatial interaction models, and associated developments. Environ. Plan. A 3, 1–32.

SUBCHAPTER 10.3

# Data for Training Models, Domain Adaptation

**Antonio M. López, David Vázquez and Gabriel Villalonga**
CVC-UAB, Barcelona, Spain

## 10.3.1 TRAINING DATA AND GROUND TRUTH

Nowadays it is rather clear that sensor-based perception and action must be based on data-driven algorithms. In other words, we must use machine learning techniques for developing the algorithms that automatically perform the required tasks, from perception to action. Obviously, resorting to machine learning implies relying on data. For the sake of simplicity, in the following we assume we are discussing visual data, i.e., images, but the considerations we are going to introduce can be also extrapolated for other types of data, such as LIDAR, RADAR, etc. However, working with images is specially challenging, not surprisingly, since the sense of sight and how to understand the world through it is incredible difficult for machines.

Applying machine learning for developing a visual task, i.e., to learn a visual model, implies having three different datasets of images: (1) training; (2) validation; (3) testing. Equivalently, we can consider that a single dataset can be randomly split into those three to have several combinations. The training images are used to learn the desired visual model, i.e., its parameters. Such models usually have hyper-parameters that are set by trial and error (a very primitive and costly form of learning). Given a trial of hyperparameters the model parameters are learned and tested on the validation dataset. Then, it is selected the trial of hyperparameters that shows the best results on this dataset. Finally, by applying the learned model to the testing dataset, we obtain a proxy of its expected accuracy in real-world conditions.

In terms of relative sizes, usually the training set is much larger than the others. The validation set used is the smallest one. A typical split of all available data can be 60% for training, 10% for validation, and 30% for testing. The reason is that the machine learning algorithms in general produce better models when more data is available provided it has been collected randomly, i.e., without any undesirable bias (obviously less data but better selected may produce better models than lots of redundant data). Thus, it is desired to use most of the data for training. Note that training, validation, and testing datasets cannot overlap for ensuring that the measured accuracy of the learned model makes sense in terms of generalization, i.e., in terms of how the model will behave under previously unseen data. Altogether, this implies that most of the data should be used for training. Moreover, when the absolute amount of data is low, then, as mentioned before, the train—validation—test splitting is performed several times to come up with different models, which brings a more realistic assessment of both the machine learning method in use and the usefulness of the learned models in terms of their accuracy. In the following, we will use the term "training" to refer to both "training" and "validation" as defined here, because these stages are part of the process of developing a model, while testing is used for assessing its performance. In this way we simplify the terminology without losing generality.

Once we have introduced the critical need for training data we have to add what, in fact, is the most challenging point. It is not only that we need images for training, but also the ground truth associated to them. Fig. 10.3.1 draws the idea for two specific vision-based tasks: object detection and semantic segmentation. In the former case, the ground truth consists of the bounding boxes (BBs) framing the objects (cars in the example). In the latter case, the ground truth consists of the silhouettes of all the semantic classes in consideration (road surface, sky, vegetation, building, vehicles, pedestrians, etc.); in other words, a class must be assigned to each pixel of each image used for training. What is the problem? These ground truths are provided manually, which is a tiresome procedure prone to errors. Obviously, both for validation and testing ground truth is also required, but the most time-consuming part is due to training since, as we have mentioned, this is the stage that requires most of the data. It is worth mentioning that in the machine learning literature there are proposals that try to train models without the use of ground truth; however, the models that are really accurate do need such ground truth. These are the so-called supervised machine learning methods, in contrast to the unsupervised ones (no ground truth used). Well-known examples of supervised machine learning methods are support vector machines (SVM), logistic regression, Adaptive Boosting (AdaBoost), Random Forest, and Convolutional Neural Networks (CNN).

In the fields of advanced driver assistance systems (ADAS) and autonomous driving (AD), we can find several examples of datasets with ground truth publicly available. In the ADAS community a popular pioneering

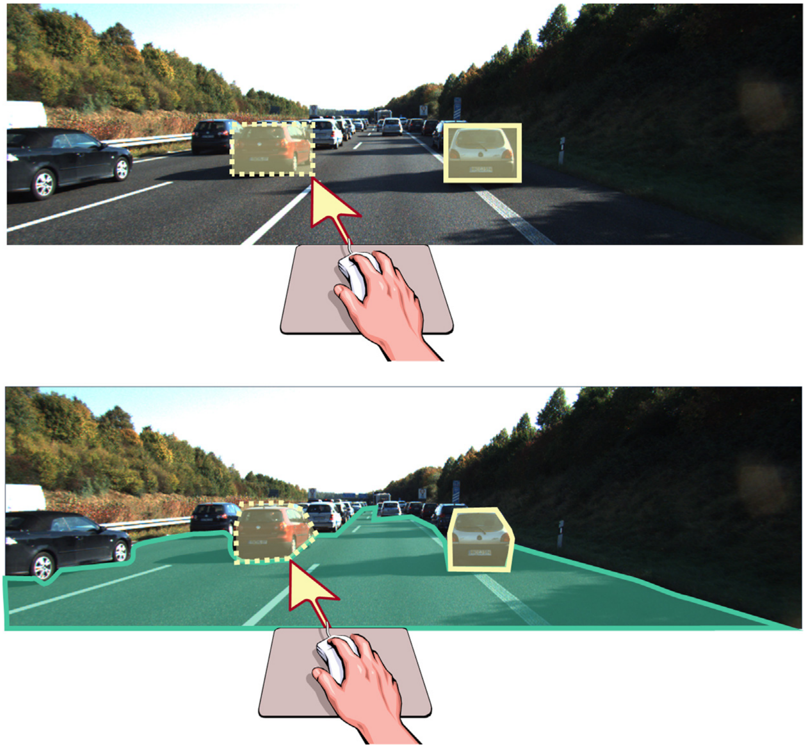


**Figure 10.3.1** Manual annotation (labeling) of: (top) bounding boxes (BBs) that frame objects (cars in this case); (bottom) silhouettes of all the semantic classes, i.e., pixel-wise assignment of semantic classes (road surface, vehicles, etc.).

example was the Daimler Pedestrian dataset of Enzweiler and Gavrila (2009), which includes 3915 BB-annotated pedestrians and 6744 pedestrian-free images (i.e., image-level annotations) for training, and 21,790 images with 56,492 BB-annotated pedestrians for testing. Another pioneering example corresponds to the pixel-wise class ground truth provided in Brostow et al. (2009) for urban scenarios; giving rise to the popular CamVid dataset which takes into account 32 semantic classes (although only 11 are usually considered) and includes 701 annotated images, 300 normally used for training and 401 for testing. A few years after, the KITTI Vision Benchmark Suite of Geiger et al. (2016) was an enormous contribution for the research focused on ADAS/AD given the high variability of the provided synchronized data (stereo images, LIDAR, GPS) and ground truth (object bounding boxes, tracks, pixel-wise class, odometry). More recently, Daimler has led the release of the so-called

Cityscapes dataset of Cordts et al. (2016), which tries to go beyond KITTI in several aspects. For instance, it includes 5000 pixel-wise annotated (stereo) images covering 30 classes and per-instance distinction, with GPS, odometry, and ambient temperature as metadata. In addition, it includes 20,000 more images but where the annotations are coarser regarding the delineation of the instance/class contours. This kind of dataset is difficult to collect since driving through 50 cities covering several months and weather conditions was required. In order to appreciate how difficult it is to provide such ground truth, we can mention the fact that annotating one of those images pixel-wise may take from 30 to 90 minutes of human labor depending on the image content. Thus, assuming an average of 60 minutes, annotating the 5000 images mentioned before, requires 5000 working hours for a person. Fig. 10.3.2 shows examples of Cityscapes images: each color represents a different urban semantic class (e.g., light pink means sidewalk, dark pink road, red pedestrian, etc.). In the top there is an example of a finely annotated image, in the bottom we see a coarsely annotated one. Note how the silhouettes of the classes are not accurately traced in the coarse case.

In order to shorten the annotation time and be more robust to erroneous annotations, we can think about crowdsourcing this task. For instance, this was the approach followed in the computer vision community where tools such as Amazon's Mechanical Turk (AMT) and LabelMe of Russell et al. (2008) were used to annotate popular publicly available datasets such as ImageNet (see Deng et al., 2009), and PASCAL VOC (see Everingham et al., 2010). However, crowdsourcing usually seeks low cost and, therefore, is not based on professional annotators. As a consequence, methods to automatically assess the quality of the ground truth are still required. In fact, since ADAS and AD face mobility safety, companies must rely on a more professional pipeline with many qualified annotators involved in the annotation of the data. In addition, not all kinds of ground truth can be provided by relying on manual annotations. For instance, we may need to develop a dense (pixel-wise) depth estimation algorithm or an optical flow one. A person cannot manually provide the pixel-wise ground truth desired to train and/or test such algorithms.

The reader may appreciate already how difficult is ground truth collection. However, we can see that the situation is even worse by introducing a very relevant point not yet mentioned here. As we introduced in Section 9.1.4, deep learning and, in particular, CNN architectures are the core of the state-of-the-art of many computer vision tasks, including

**Figure 10.3.2** Ground truth examples from Cityscapes dataset.

those related to ADAS/AD, such as object detection and semantic segmentation. The starting point of this breakthrough was the task of image classification (i.e., assigning a single label to a full image), for which the AlexNet of Krizhevsky et al. (2012) just smashed the previous state-of-the-art. This work already pointed out one of the reasons for the success of deep CNNs in general, namely the massive availability of data with ground truth. In particular, AlexNet was trained on the ILSVRC dataset, with about 1000 images of 1000 categories; overall, about 1.2 million images for training, 50,000 for validation, and 150,000 for testing, all of them with image-level class annotation.

The publicly available datasets of reference in ADAS/AD, i.e., KITTI and Cityscapes, are orders of magnitude away from ILSVRC. Moreover, image-level ground truth is too poor for ADAS/AD tasks where objectwise (BBs) ground truth is a minimum, but most of the times pixel-wise ground truth (Fig. 10.3.2) is required. Nowadays, even for ADAS/AD tasks, the fine-tuning approach is followed; i.e., taking a deep CNN such as AlexNet and somehow reusing it by exposition to the more scarce annotated data acquired for the new (ADAS/AD) tasks.

## 10.3.2 VIRTUAL WORLDS AND DOMAIN ADAPTATION

Due to all these considerations, a totally different way of addressing the ground truth acquisition problem has been assessed. It started timidly in 2010 but nowadays there is an explosion of works in this line, with even workshops devoted to it. We refer to the use of virtual worlds for generating realistic images (and potentially data from other simulated sensors) with automatically generated ground truth. Fig. 10.3.3 illustrates the idea with the pioneering work of Marin et al. (2010). In this case, a modification of the videogame Half-Life 2 was used for automatically generating pixel-wise ground truth for the pedestrians contained in virtual-world RGB images. These images are acquired on board a virtual car that drives along an urban scenario of the virtual city. In Marin et al. (2010) it was demonstrated that using the state-of-the-art pedestrian detector at that time (i.e., pyramid sliding window, HOG/Linear-SVM classifier, non-maximum suppression), the accuracy of the classifier trained on the virtual environment and the accuracy of an analogous classifier trained on real-world images (having access to the same number of samples) was statistically the same.

Further extensions of Marin et al. (2010) demonstrate that the results were not always directly as good as expected (see Vazquez et al., 2014). In particular, the accuracy obtained by different object detectors was lower when training with virtual-world data and testing in a given real-world dataset, than when training with the data of such real-world dataset. However, it was demonstrated by Vazquez et al. (2014) that the accuracy gap was not really specifically due to virtual-to-real differences. Virtual-to-real was shown to be a special case of a more generic problem, namely sensor-to-sensor differences. In other words, training an object detector with images of a given camera model and testing with images of another camera model, also ends up in worse results than if training and testing images come from the same camera. Note how important this problem is; if we annotate a large dataset of images for ADAS/AD and later we change the camera, we may need to annotate again another large dataset

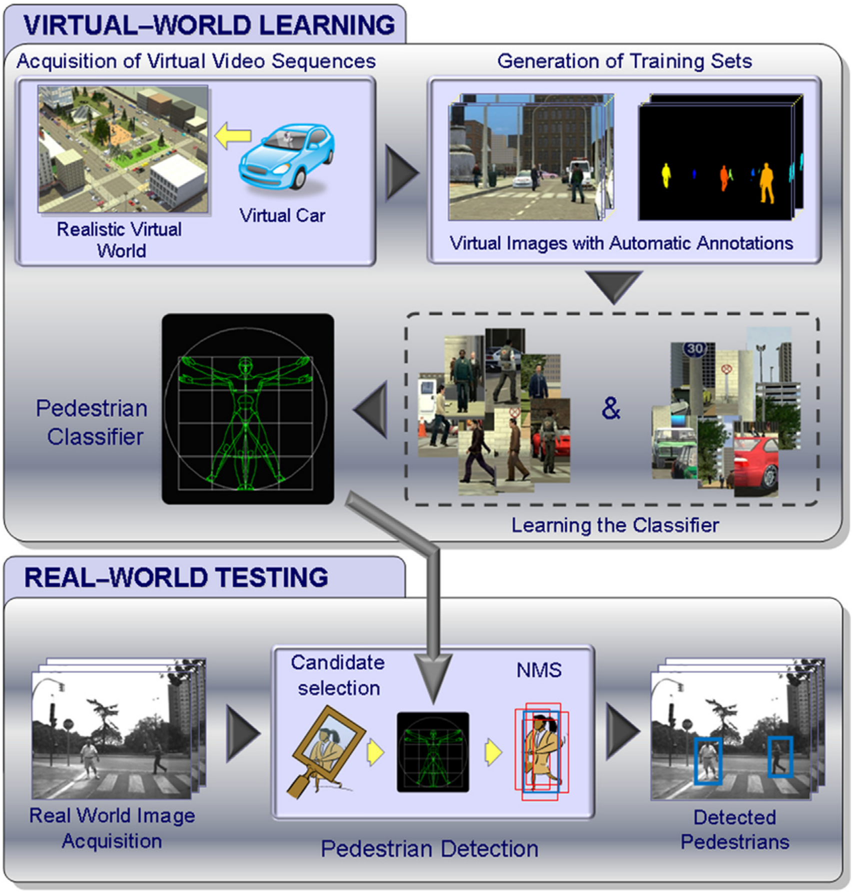



**Figure 10.3.3** Example of pedestrian detector where the pedestrian classifier is trained in a virtual world, and then plugged into a detection pipeline to process real-world images.

to achieve the same accuracy. In fact, the problem is even more generic than sensor-to-sensor differences, the discrepancies in the statistics of the image content cause accuracy drops as well. As a matter of fact, the computer vision community started to realize this problem, which was just ignored for a long time (see Saenko et al., 2010, 2011). In particular, the so-called domain adaptation (DA) and transfer learning (TL) techniques started to gather relevance among the computer vision community since they pursued reusing previous knowledge (in the form of model or annotated data) for performing accurately in new domains and tasks, but using either many data without annotations (unsupervised DA/TL) or few data with annotations (supervised DA/TL). For instance, in Vazquez et al. (2014) supervised DA based on active learning was used to adapt virtual and real domains and, therefore, recovering the above mentioned accuracy gap of the pedestrian detector developed in the virtual world. In Xu et al. (2014) a more sophisticated technique was used to adapt deformable part-based models (DPM) from virtual to real domains, in this case and contrarily to Vazquez et al. (2014), without revisiting the source (virtual-world) data. Fig. 10.3.4 illustrates the idea: an initial DPM is learned by using virtual-world data with automatic ground truth for pedestrians. The model is applied to real images. Some pedestrians are not detected and background regions are classified as pedestrians due to the domain gap (virtual-to-real). To solve this gap the initial DPM is refined by either actively collecting errors with a human oracle in the loop, or a procedure is able to automatically collect annotations without human intervention. The DPM refinement can be done progressively by iterating this procedure. As guiding information, in Vazquez et al. (2014) and Xu et al. (2014) the proposed DA techniques saved 90% of the annotation effort that would be needed to obtain the same accuracy in the real-world (target) domain.

Overall, these experiments showed that appearance models trained in virtual worlds act as strong priors with the potential of saving a large amount of human annotation effort. Interestingly, the winner of the first pedestrian detection challenge in the KITTI dataset was based on a

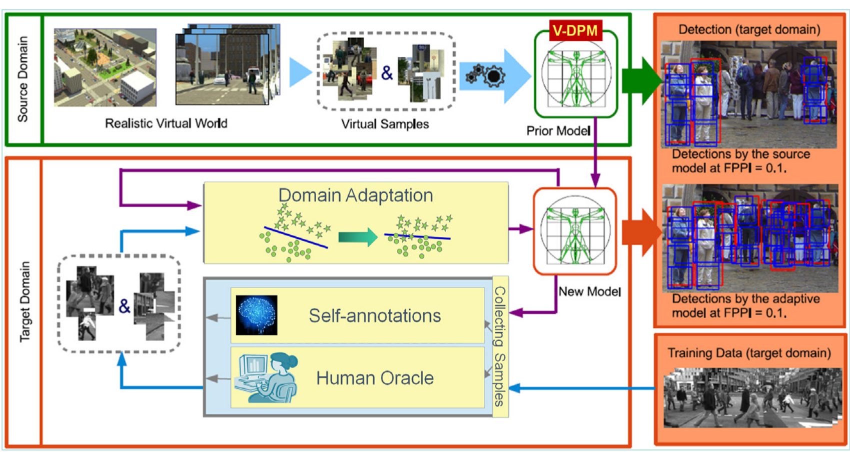



**Figure 10.3.4** Domain adaptation when the source domain is a virtual world and the target domain is the real world.

virtual-to-real-world domain-adapted classifier (Xu et al. (2016)). Note also that automatically collected ground truth is more precise than that collected by humans. It is also worth mentioning that even deep CNNs require DA/TL (see Tommasi et al., 2015).

Driven by the success of the use of virtual-world data and domain adaptation for training the appearance pedestrian models, in the last two years new works have been presented going far beyond pedestrian detection. For instance, in Ros et al. (2016) a very large city was created, named SYNTHIA, which allowed generating hundreds of thousands of RGB images (random and arranged as sequences) with all kinds of interesting ground truth automatically generated: pixel-wise class ID, instance ID, and depth; vehicle odometry; and 360 degrees views. In order to force variability, the city includes many pedestrian models, vehicles, city styles, highways, vegetation, lighting conditions, and four seasons. Fig. 10.3.5 shows different snapshots of the city content and ground truth. Using data from SYNTHIA and basic domain adaptation techniques, Ros et al. (2016) show that it is possible to boost the accuracy of deep CNNs designed for semantic segmentation. Season- and lighting-dependent images together with vehicle odometry can be used to train place recognition methods that may be part of vehicle localization in maps (a key component nowadays of prototypes of self-driving cars).

Analogously, Gaidon et al. (2016) presented a virtual environment that mimics KITTI, termed as Virtual KITTI, showing its usefulness for

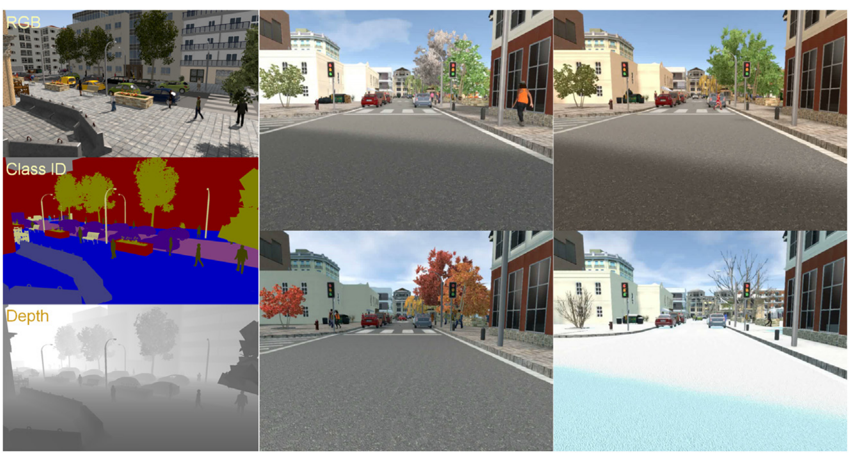



**Figure 10.3.5** SYNTHIA: RGB image, ground truth for Class ID and depth; images acquired from the same camera location at different seasons.

designing object trackers (e.g., for tracking cars). Following the same line of work, Richter et al. (2016) show semantic segmentation results with deep CNNs using the GTA-V videogame world. Interestingly, other lower level visual tasks such as depth estimation and optical flow estimation are being currently addressed by the use of deep CNNs trained on virtual data, see Mayer et al. (2016). Note that ground truth for such tasks cannot be collected by humans.

In this setting one of the arising questions is how the degree of photorealism of the virtual images affects training visual models. By comparing SYNTHIA images and GTA-V ones (Fig. 10.3.6), in Lopez et al. (2017)


**Figure 10.3.6** Comparing photorealism. GTA-V (top) is more photorealistic than SYNTHIA (bottom).

it is shown that even for the most realistic video-games, the virtual-to-real domain gap is still an issue. This is not surprising, since we have mentioned before that there may be sensor-to-sensor domain gaps even for real-world sensors.

We would like to highlight that virtual environments are gaining attention not only for understanding the sensor raw data, but also for learning to act (see Dosovitskiy and Koltun, 2016); in other words, given an image, a deep learning architecture directly outputs the control commands for self-driving (e.g., steering angle, brake/accelerate, etc.), without explicitly creating an intermediate 3D understanding of the driving scenario.

Finally, we would like to note that virtual worlds are not only useful for training models, in fact, they can be a very convenient tool for exhaustive simulations that allow the setting of hyperparameters, debugging the behavior of algorithms, experimenting on corner cases, etc.; in other words, the more traditional functionalities assigned to simulators of any kind. Obviously this is a more standard use of virtual environments, and the revolution has been to see that they can be also used for training models, especially visual deep models.

## REFERENCES

Brostow, G.J., Fauqueur, J., Cipolla, R., 2009. Semantic object classes in video: A high-definition ground truth database. Pattern Recogn. Lett. 30 (20), 88–89.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., et al., 2016. The Cityscapes dataset for semantic urban scene understanding. In: IEEE Conference on Computer Vision and Pattern Recognition.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition.

Dosovitskiy, A., Koltun, V., 2016. Learning to Act by Predicting the Future. arXiv 1611, 01779.

Enzweiler, M., Gavrila, D.M., 2009. Monocular pedestrian detection: Survey and experiments. Trans. Pattern Recogn. Mach. Anal. 31 (12), 2179–2195.

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The PASCAL visual object classes (VOC) challenge. Int. J. Comput. Vis. 88 (2), 303–338.

Gaidon, A., Wang, Q., Cabon, Y., Vig., E., 2016. Virtual worlds as proxy for multi-object tracking analysis. In: IEEE Conference on Computer Vision and Pattern Recognition.

Geiger, A., Lenz, P., Stiller, C., Urtasun, R., 2016. Vision meets robotics: The KITTI dataset. Int. J. Robot. Res. 32 (11), 1231–1237.

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. ImageNet classification with deep convolutional neural networks. In: Annual Conference on Neural Information Processing Systems.

Lopez, A.M., Xu, J., Gomez, J.L., Vazquez, D., Ros, G., 2017. From virtual to real-world visual perception using domain adaptation — the DPM as example. Domain adaptation in computer vision applications, Springer Series: Advances in Computer Vision and Pattern Recognition, Edited by Gabriela Csurka.
Marin, J., Vazquez, D., Geronimo, D., Lopez, A.M., 2010. Learning appearance in virtual scenarios for pedestrian detection. In: IEEE Conference on Computer Vision and Pattern Recognition.
Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., et al., 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: IEEE Conference on Computer Vision and Pattern Recognition.
Richter, S.R., Vineet, V., Roth, S., Vladlen, K., 2016. Playing for data: ground truth from computer games. In: European Conference on Computer Vision.
Ros, G., Sellart, L., Materzyska, J., Vazquez, D., Lopez, A.M., 2016. The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: IEEE Conference on Computer Vision and Pattern Recognition.
Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T., 2008. LabelMe: a database and web-based tootl for image annotation. Int. J. Comput. Vis. 77 (1—3), 157—173.
Saenko, K., Kulis, B., Fritz, M., Darrell, T., 2010. Adapting visual category models to new domains. In: European Conference on Computer Vision.
Tommasi, T., Patricia, N., Caputo, B., Tuytelaars, T., 2015. A deeper look at dataset bias. In: German Conference on Pattern Recognition.
Vazquez, D., Lopez, A.M., Marin, J., Ponsa, D., Geronimo, D., 2014. Virtual and real world adaptation for pedestrian detection. Trans. Pattern Recogn. Mach. Anal. 36 (4), 797—809.
Xu, J., Ramos, S., Vazquez, D., Lopez, A.M., 2014. Domain adaptation of deformable part-based models. Trans. Pattern Recogn. Mach. Anal. 36 (12), 2367—2380.
Xu, J., Ramos, S., Vazquez, D., Lopez, A.M., 2016. Hierarchical adaptive structural SVM for domain adaptation. Int. J. Comput. Vision 119 (2), 159—178.