



Simulated Pedestrian Modelling for Reliable Testing of Autonomous Vehicle in Pedestrian Zones

Qazi Hamza Jan, Jan Markus Arnold Kleen^(✉), and Karsten Berns^(✉)

Technische Universität Kaiserslautern, Kaiserslautern, Germany
{hamza, j_kleen15, berns}@cs.uni-kl.de
<http://www.agrosy.uni-kl.de>

Abstract. Pedestrian zones are increasing in lengths and the relevant authorities are aiming at introducing autonomous vehicles to such zones for different purposes like transportation vehicles, delivery vehicles and cleaning vehicles. As the name suggests, pedestrian zones are crowded by pedestrian having highest priority. Vehicles in such zones should take special care regarding the safety of the people. Unlimited situations can occur from the pedestrians. Hence, it becomes crucial to test such vehicles before deploying in real-world to ensure safety of the people and the environment. Therefore, there is a need to have a realistic behaviors of pedestrians in simulation for critical testing of autonomous vehicles. The major novelty in this paper is the generation of virtual pedestrians which avoid obstacles realistically and to provide a platform for accurate and easy testing of autonomous vehicles in pedestrian zones. These virtual pedestrians avoid all kinds of obstacles based on self-awareness while progressing towards the goal. Main feature of these pedestrians are based on direction and speed heuristics. The goal is to have minimum number of parameters to generate test scenarios with realistic behaving pedestrians for autonomous systems. A thorough experimentation is done to validate the system. In the final experiment, a real-world scene is recreated from the simulation and the results are compared. It is seen visually that both the trajectories from the simulation and real-world highly resemble to each other. By slightly tweaking the parameters, one can generate different test scenario.

Keywords: Pedestrian simulation · Autonomous vehicles · Pedestrian zone · Virtual pedestrians

1 Introduction

With the increase usefulness of autonomous vehicles, the demand for these systems is increasing in different sectors such as agriculture, forestry, urban transport, construction and so on. In the past few years, demand for autonomous shuttles in pedestrian zones is also growing. Pedestrian zones are public areas reserved for pedestrians and prohibited for most of the auto-mobile traffic. These areas include city centres, tourist attractions, airports, university campuses and so on. Two main reasons for having autonomous shuttles in such zones is:

- the difficulty in reaching from one point to another in pedestrian zones.
- lack of conventional transport system in such zones.

As the population is growing normal streets are converted to pedestrian zones in order to sustain the open environment. This will allow more accessibility and easy flow of huge crowds. To accomodate the elderly and disabled people to traverse such huge zones, it is relevant to have a particular transport system which can safely carry passengers through these crowded environments. Therefore, it becomes important to have a system which is safe and reliable and can carry people in a smooth manner. There are already few companies, for example, EasyMile [1] and Navya [2] which have autonomous vehicles in testing phase. These vehicles are driving in a restricted environment to verify its credibility. [3] shows lack of interaction could result in collision and how important it is to have interaction with the pedestrians.

Reliability and safety are crucial aspects when developing a system for autonomous shuttles in pedestrian zones. Pedestrians can include children, juveniles and senior. Different factors effect the behavior of such road user [17–19] Hence, these systems should ensure safety of pedestrians at all cost. For this reason, it becomes important to test the system thoroughly before launching in the real-world scenario. Since, the testing cannot be performed around real people due to high risk of system failure, therefore, we provide simulated pedestrians which exhibits a realistic behavior similar to pedestrians walking in pedestrian path for testing autonomous vehicles. These behaviors include avoiding vehicles in a uniform manner, going parallel to vehicle in a small area etc. Throughout the length of this paper, the simulated pedestrians will be called virtual pedestrians to differentiate between real-world pedestrians and simulated pedestrians. The simulated system should be capable of fast testing scenarios which include:

- fast generation of virtual pedestrians.
- randomness in their behavior.
- capability of avoiding static and dynamic obstacle.
- behavior avoidance for vehicles.
- moving of virtual pedestrians towards a specific goal.
- additional behaviors for virtual pedestrians like texting, talking on the phone and so on.

This work is the extension of already published paper [4]. This work addresses the problem of validating autonomous systems in the vicinity of pedestrians. This paper highlights a detailed description of generating virtual pedestrians for easy and quick testing of autonomous vehicles in a pedestrian zone. The generated virtual pedestrians have knowledge of their surroundings and can move on the given way-points. These virtual pedestrians can differentiate between other pedestrians and the vehicle to get a reasonable trajectory. Our simulation offers a variety of parameters to construct scenarios. The goal of this system is to have virtual pedestrians which mimic real-world pedestrians.

In the next section, related work specifically to pedestrian simulation in the domain of autonomous vehicles is presented. Following, a detailed description

of methodology is discussed. Section 5 compares the results of our method with already existing methods. In the final section, we deduce the results for our virtual pedestrians.

2 Related Work

As already mentioned, autonomous vehicles are functioning more and more in pedestrian paths. Researchers are actively working on different navigation strategies. These navigation strategies include avoiding any kind of confusion with the pedestrian along with the continuous flow of the vehicle. The authors in [5] have focused on interaction modules to grab pedestrian's attention. This strategy is useful for crowded environment to avoid freezing robot problem. Their focus is also on pedestrian zone and they have performed experiments in simulated environment.

Carla [7], another open-source tool to support autonomous driving in simulation has recently been introduced. It provides different cities and town models along with variety of sensors. It also aims at facilitating autonomous driving system in urban environment. For realistic scenes in urban environment, Carla offers pedestrian on side-walks. The pedestrian modelled in Carla navigate by location based cost. Carla requires information of the town for example map for the navigation of virtual pedestrians. These virtual pedestrians are designed to walk on footpath and occasionally cross the road. They do not provide specific avoiding behavior of pedestrians to vehicle.

Some researchers are working on crowd evacuation models. Authors in [13] have used social force model in high density environment. They have modified social force model for bi-directional pedestrian streams. In [14], the authors have used residual network based scene-independent crowd simulation (ResNet-SICS) framework to simulate crowd simulation. It is a data driven approach in which data-driven crowd properties quantization (DCPQ) method is proposed. Authors in [15] have used virtual reality (VR) head-mounted display for pedestrian simulation. This enables simulation of pedestrians in virtual environment. Users can act as pedestrians, drive a car or just visualize a scene. They have a client-server architecture where the user can interact with each other. For crowd formation, it would require large number of people for testing. Hence, for dense crowd vehicle testing is not feasible.

A similar case for testing conflict of pedestrian and vehicle is presented in [16]. They have used Surrogate Safety Measures (SSMs) to validate Pedestrian Protection System (PPS). They have considered three SSMs; Pedestrian Classification Time to Collision (PCT), Total Braking Time to Collision (TBT) and Total Minimum Time to Collision (TMT). They have used PreScan for building the simulation. There some pedestrian simulation softwares like [20] used for planning, managing and optimizing pedestrian flow in urban environment in buildings. This simulation would be useful for vehicles inside buildings. But they do not offer high quality rendering. Another open-source framework for pedestrian simulation is [21]. It offers pedestrian dynamics at microscopic level. The models include gradient navigation model, social force model, and optimal steps model. For variety of situations, another pedestrian simulation tool [22] can be used. This simulation

offers to study the flow of pedestrians. This software does not specialize in vehicle pedestrian interaction since we expect different reaction of pedestrians to a vehicle.

However, our system offers virtual pedestrians which can freely walk around a pedestrian path towards their goal point. They have high rendering and perform different animations like texting, talking on a phone etc.

3 System Architecture

Due to increase in pedestrian zone lengths and growth in automated systems, driver-less vehicles are surfacing in pedestrian zones as well. These vehicles can be delivery vehicles, transport vehicles or waste collection vehicles. Each vehicle has its own purpose and drives accordingly. For example, a delivery vehicle has to make way door to door to deliver packages. On contrary, transport vehicles need to carry passengers from one point to another. For all such vehicles it is observed that pedestrians show a common behavior. Pedestrians would always try to avoid colliding with all such vehicles only if they are aware of the approaching vehicle. For unaware pedestrians, it becomes important to either give signals from the vehicle or cross them safely from a side, if space available. Due to high dynamicity in pedestrian zones, all such vehicles are obligated to be tested thoroughly before inaugurating it for its main purpose. For testing, a similar virtual dynamic environment must be created.

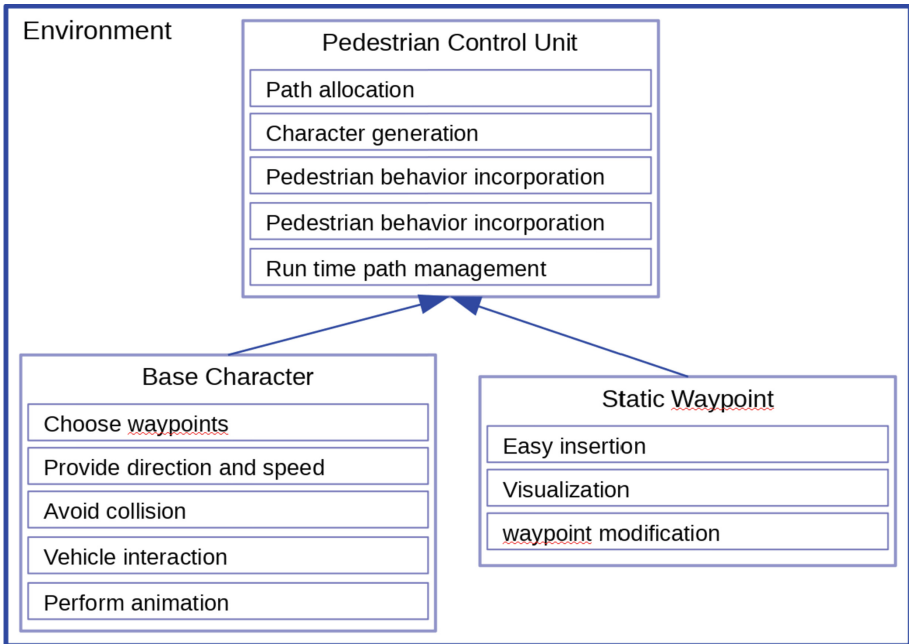


Fig. 1. A block diagram of the system is shown. It consists of three main modules. All the functionalities are enlisted in their respective module briefly [4].

This paper proposes to implement pedestrians in simulation for such driverless vehicles which are designed to navigate in pedestrian path. The main objective in designing such system is to provide virtual pedestrians with identical characteristics to that of real-world pedestrians. The fundamental behavior of the virtual pedestrians to account for is walking towards their goal and avoiding any kind of obstacle. For testing purposes, the system should provide flexibility to incorporate other behaviors as well. According to Federal Highway Administration University Course on Bicycle and Pedestrian Transportation [6], there exist pedestrians with different age groups. Based on their age group pedestrians have different actions and responses towards vehicles. The age groups are summarized in the following:

- Infants and toddler (age 0 to 4)
- Young children (age 5 to 12)
- Pre-teens (age 13 to 14)
- High school aged (age 15 to 18)
- Adults (age 19 to 40)
- Middle aged adults (ages 41 to 65)
- Senior Adults (ages 65+)

Based on the experience and peripheral vision, every age group reacts to a particular situation. Infant and toddlers are impulsive and unpredictable and are developing peripheral vision. Young children lack experience and short for drivers to see. Pre-teens also lack experience but ride more frequently under risky condition. They have a sense of invulnerability. High school aged are very active and are capable of travelling at higher speed. Adults are fully aware of traffic environment and interested in improving conditions. Whereas, middle aged adults may slowing of reflexes. The senior adults have difficulty hearing vehicles approaching from behind.

Therefore, it is essential to have different age groups of pedestrians in testing environment. This system develops virtual pedestrians that can be easily spawned in the environment and have randomness in their behaviors. It enables different crossing behaviors for the same trajectory of the vehicle and the same goal points of the virtual pedestrian. In addition, this behavior can be changed by changing the position of goal points which are discussed in detail in the later section.

Outline of our system architecture is given in Fig. 1. The figure shows main modules of the system in the environment. Each of the module is dependent on each other. The Pedestrian Control Unit (PCU) handles major of processing, i.e., from path allocation to run time path management. Base Character (BC) passes waypoints given from user interface in PCU to every virtual pedestrians. It also does the collision avoidance for each virtual pedestrian. Static Waypoint (SW) are separate unit which are automatically attached to the PCU. SW are the trajectory points on which virtual pedestrians are intended to navigate. Each module in the Fig. 1 has been explained in the following subsections.

For the sake of high quality perception, we have used Unreal Engine [8]. Unreal Engine is a state of the art gaming engine. It has a photo-realistic rendering, dynamic physics and effects and lifelike animations. It offers built-in class for character animation. Unreal engine uses Artificial Intelligence to perform certain actions. The decision making is done using behavior trees. It also uses artificial intelligence perception system to retrieve information from the environment. The collision avoidance in Unreal Engine is done using Reciprocal Collision Avoidance (RVO) [12]. This method is based on avoiding collision using velocity of the agents. Every agent should have information about the velocity of other agent. This is done by observing the velocity of other agent. No central coordination among the agents is required to solve the collision problem in RVO. Consequently, the agent can only observe the velocity of other agents. If all the agents select velocity other than the reciprocal velocity then it guarantees that the agent will not collide. Comparison of RVO method with our system is given in the experimentation.

3.1 Pedestrian Control Unit

PCU has been designed for managing all the virtual pedestrians in the environment. PCU acts as the main unit in all the modules given in Fig. 1. The idea to have a main unit is to make an interface where user can easily define parameters for test scenarios. This interface is made simple to accommodate all possible parameters. All the information given by the user is further carried out to other modules.

Suitable parameters are specified for a user to shape a scenario. Some fundamental parameters for scene creation are as following:

- number of pedestrians existing in the environment
- pedestrians existing as single or in group
- walking speed of virtual pedestrians
- type of animation required along with walking
- number of people in a group
- visible way points for debugging
- removal of virtual pedestrians after path completion

3.2 Base Character

Base Characters are the physical entity in the environment. To show the physical entity in the simulation, geometrical shapes exist in simulation. These pieces are known as mesh. Every model designed in Unreal Engine is a static mesh. These meshes can be made complex to clearly replicate a human. Advantage of the meshes in Unreal Engine is that it is rendered using graphic card, hence, enabling high quality replicas. For autonomous vehicle testing it becomes important to use these complex meshes to offer the system a realistic stream of virtual pedestrians. Later, these are used for object detection algorithms [9] and specifically for face detection [10].

3.3 Static Waypoint

When creating a test scenario, it is already defined where the virtual pedestrians can maneuver. It is done by assigning them goal points. These goal points are known as Static Waypoint (SW). SWs are placed in the environment where the pedestrians are required to walk. Normally, they are placed on the pedestrian path where the vehicle is required to drive. For example, when placed exactly at the centre of the pedestrian path, the virtual pedestrians and vehicle always comes face to face. In this case either pedestrian or the vehicle has to give the way.

A threshold is defined to detect when the virtual pedestrian has reached the SW. This radius around the SW is known as epsilon. Every virtual pedestrian enters the radius of the SW at least once when following the path assigned. If the radius is large, the virtual pedestrians upon entering the radius of the SW directly turns towards the next SW. It would create problems at the corner SWs because the virtual pedestrian might walk outside of the pedestrian path to follow its trajectory if there is curvy path. If the radius is very less and there is an obstacle on the SW, then either the virtual pedestrians will wait for the SW to be cleared or go to the next SW after waiting too long. For testing purposes,

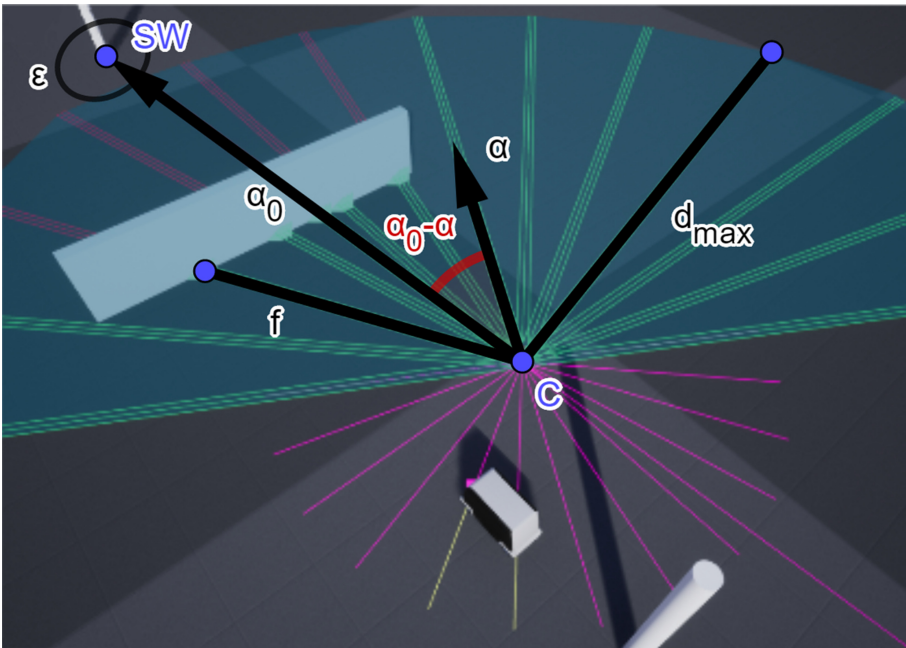


Fig. 2. Simulation view of virtual pedestrian c in the vicinity of vehicle and an obstacle. Two static waypoints as start and goal are show by cylinder shape. The maximum view of the virtual pedestrian c is d_{max} . α are the directions defined and α_0 is the direction towards the goal point. f is the distance to the obstacle [4].

it should be decided how big epsilon should be. For different values of epsilon, virtual pedestrians behaves differently.

Behavior of virtual pedestrians also depends on the placement of the SWs. The further the SWs are placed, easily the pedestrians give way to the vehicle. Nearer the SWs, more precisely the virtual pedestrians follows the path.

4 Proposed Approach

To provide a realistic testing platform for autonomous vehicle testing in a pedestrian path, we propose to generate virtual pedestrians that move along a pre-defined path and avoid obstacles on its way. This reaction is also observed in humans. Generally, a person tends to avoid collision with other people or any object in their path. This behavior can be achieved if the person is aware of its surrounding. For the same reason, the virtual pedestrians in the proposed approach are aware of its surrounding. It can sense the obstacles in their path.

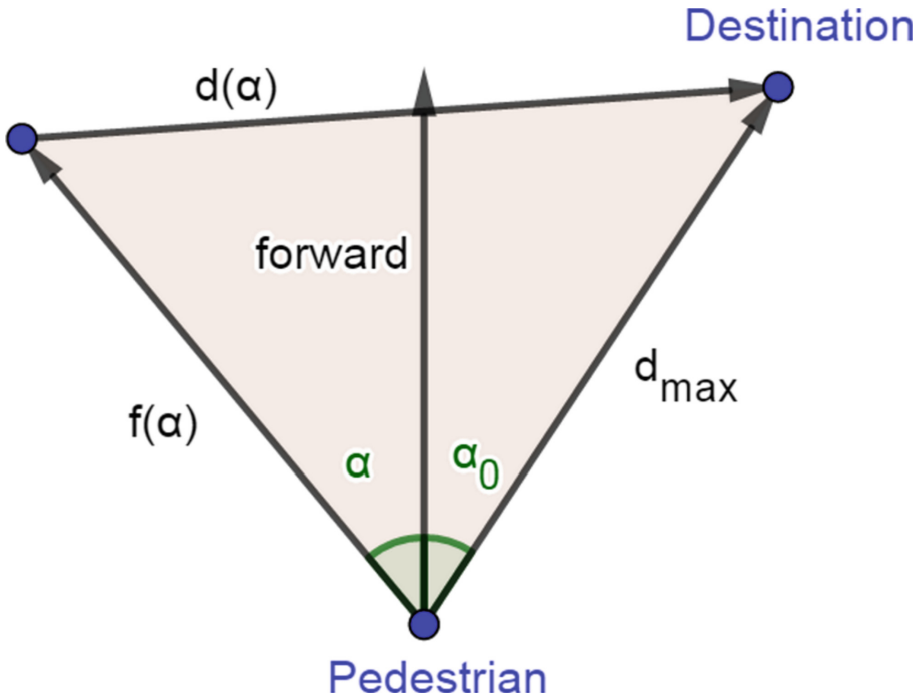


Fig. 3. To find the direction we use law of cosine. $d(\alpha)$ is the weighted distance towards the goal form the direction of α [4].

4.1 Walking Direction of Virtual Pedestrians

The direction of each virtual character is defined by using direction heuristics. At every time step, the virtual character has to re-evaluate its direction. It would calculate directions over its surrounding based on its awareness. As can be seen in Fig. 2, point c is the center of the virtual character from where it finds the direction. The green lines show the view of the virtual pedestrian. d_{max} is maximum distance the virtual pedestrian can sense. Discrete viewing angles have been defined by α , and α_0 is the direction to the goal point. We have used the minimization problem from [11].

$$d(\alpha) = d_{max}^2 + f(\alpha)^2 - 2d_{max}f(\alpha)\cos(\alpha_0 - \alpha) \quad (1)$$

$f(\alpha)$ is the distance to collision in direction of α . As the angle from the goal point α increases, the $\cos(\alpha_0 - \alpha)$ decreases making the whole term $2d_{max}f(\alpha)\cos(\alpha_0 - \alpha)$ in Eq. 1 to decrease. Hence, the weighted distance in direction $d(\alpha)$ for that particular angle will increase. This can be clearly seen in Fig. 3

The value with the highest $d(\alpha)$ is selected which will be the shortest path towards the goal. The term $2d_{max}f(\alpha)\cos(\alpha_0 - \alpha)$ is also effected by $f(\alpha)$. This is to make the robot avoid obstacles. If the value of $f(\alpha)$ is small means less distance to obstacles in the direction of α . This will again decrease the value of the term $2d_{max}f(\alpha)\cos(\alpha_0 - \alpha)$ and hence the $d(\alpha)$ for that particular α will have higher value. By calculating $d(\alpha)$ for every α at particular time step, we can find the optimal direction towards the goal. $f(\alpha)$ will make the virtual pedestrian to move away from the obstacle in the direction of goal and $\cos(\alpha_0 - \alpha)$ will make the virtual pedestrian to go towards the goal. This action is also observed in real-world pedestrians.

To avoid big obstacles like vehicles and add avoiding behavior for vehicles coming from behind, the Eq. 1 needed to be modified.

The foremost modification was done by removing d_{max}^2 . It was noted that it did not have any effect the direction having smallest value. Experiments with the heuristics determined that it works well for small objects.

Next, $2d_{max}f(\alpha)\cos(\alpha_0 - \alpha)$ causes the characters to only choose very direct directions, even if that means walking against a wall. This is due to the fact that every $d(\alpha)$, where $(\alpha_0 - \alpha) > 90^\circ$, is larger than $d(\alpha_0)$. But characters should be able to avoid not only very small but also bigger obstacles mainly vehicles. For this reason, $\alpha_0 - \alpha$ is valued less and $2d_{max}f(\alpha)\cos(\alpha_0 - \alpha)$ is replaced with $2d_{max}f(\alpha)\cos(\frac{\alpha_0 - \alpha}{x})$. In experiments $x = 2$ seems to work quite well. This way, directions with $(\alpha_0 - \alpha) > 90^\circ$ have a chance to be picked. It is noteworthy that avoidance of small objects does not get worse because characters still choose the most direct direction. Hence, the Eq. 1 is modified to Eq. 2.

$$d(\alpha) = f(\alpha)^2 - 2d_{max}f(\alpha)\cos(\frac{\alpha_0 - \alpha}{x}) \quad (2)$$

As already mentioned, the reaction of real-world pedestrians to static and dynamic obstacles is always different. For the static obstacle, they feel easy

passing nearby but when subjected to a vehicle, these pedestrians have mixed reactions. Normally, they already set out a way for the vehicle to pass. On this account, $f(\alpha)^2$ is added to Eq. 2. This ensures that the virtual pedestrian will not collide with the vehicle. If there is no other way to avoid the vehicle then the virtual pedestrian will choose a different walking speed. A sample of implementation is shown in Fig. 4.

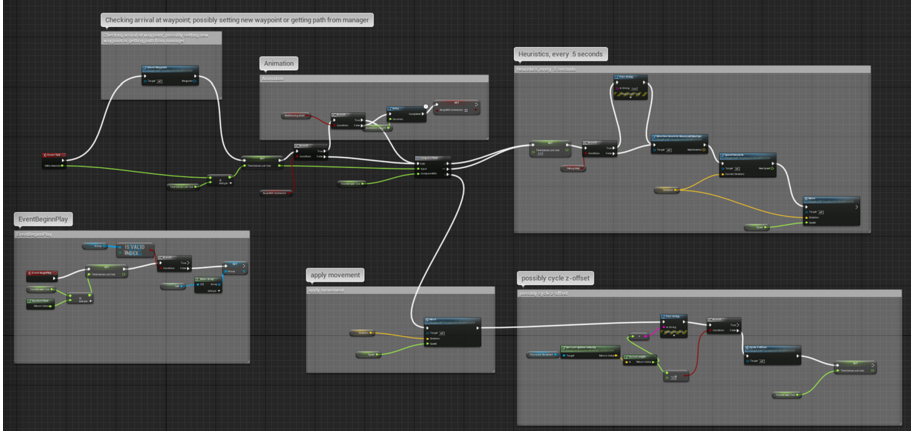


Fig. 4. Direction heuristics implementation in Unreal Engine using blueprints.

4.2 Walking Speed of Virtual Pedestrians

Since it is impossible to assign walking speed to countless virtual pedestrians, a range of walking speed is given. The upper and lower limit of the walking speed range for every virtual pedestrian is defined by the user. A preferred walking speed v_{pref} is defined and used as the upper limit for every virtual pedestrian. Velocity for every virtual pedestrian is then calculated as:

$$v = \min(v_{pref}; \frac{f(\alpha)}{t_{coll}}) \tag{3}$$

A constant minimum time to collision t_{coll} is always positive, so the character can never walk against any obstacle. The virtual pedestrian waits if the vehicle crosses the chosen direction of the pedestrian, similar to a pedestrian crossing a road in real life. For near moving vehicle v is set to zero.

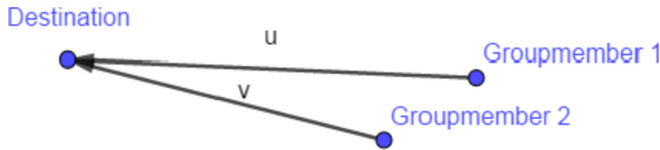


Fig. 5. The distance to group for group member 2 is $\|u - v\|$ [4].

Grouping of Virtual Pedestrians. When driving a vehicle in a pedestrian zone, different interactions can occur with pedestrians. One aspect of these interactions depends on the number of pedestrians in the environment. Pedestrians are walking in singles or a group. Single pedestrians have their independent path whereas pedestrians in a group need to remain together. Virtual pedestrians in a group walk on their own and there is no central decision-maker. Making virtual pedestrians walk in a group is done by changing their walking speeds. Conventionally, the speed for each virtual pedestrian in the group is equal, so the group remains together. It can happen that virtual pedestrian can walk away from the group. For such a scenario, the character who is away from the group needs to wait for the whole group. But being in the group every virtual pedestrian needs to respect each other personal space similar to real-world pedestrians. To know when a character needs to wait, two constants are defined; a preferred distance and a maximum distance to the group. Figure 5 shows how these distances are calculated. Considering that every member of the group should have the same SW. Here the distance of every virtual pedestrian to the group is the difference between the personal distance to current SW and the maximum distance to the current SW in the group. The reason to measure distance in this way is to allow the group to split or walk around the obstacle. This behavior is important for narrow paths where vehicle crosses from the center. This check always guarantees that at least one member of the group is not waiting for the whole group, hence, avoiding any deadlocks caused by the functionality. When the distance between the character and group increases, the character waits for the group by looking towards the group. As the group reaches the near the distant character and maximum distance becomes equal to the preferred distance, the character starts walking at its regular speed.

In PCU, the grouping behavior can be selected. There is an option to enable or disable grouping behavior. When enabled, also designate the number of groups and how many characters in a group.

5 Experiments

Pedestrian zones mainly involve pedestrians. Each of these pedestrians acts differently. There is very little probability that a pedestrian going towards the same goal at different times will react to a certain situation in exactly the same way. Due to this reason, there is an indefinite number of reactions from a single situation happening repeatedly. These reactions depend on the size of the crowd, the maturity of people, their motives, and awareness level.

In order to validate the virtual pedestrian's behavior, we have conducted several experiments that include experiments from individual characters to a group of characters and from tackling static obstacles to dynamic obstacles.

5.1 Two Individual Characters and a Vehicle Crossing

A test scenario is created which involves two individual virtual pedestrians crossing a vehicle from the front. This test is done to comprehend the behavior of

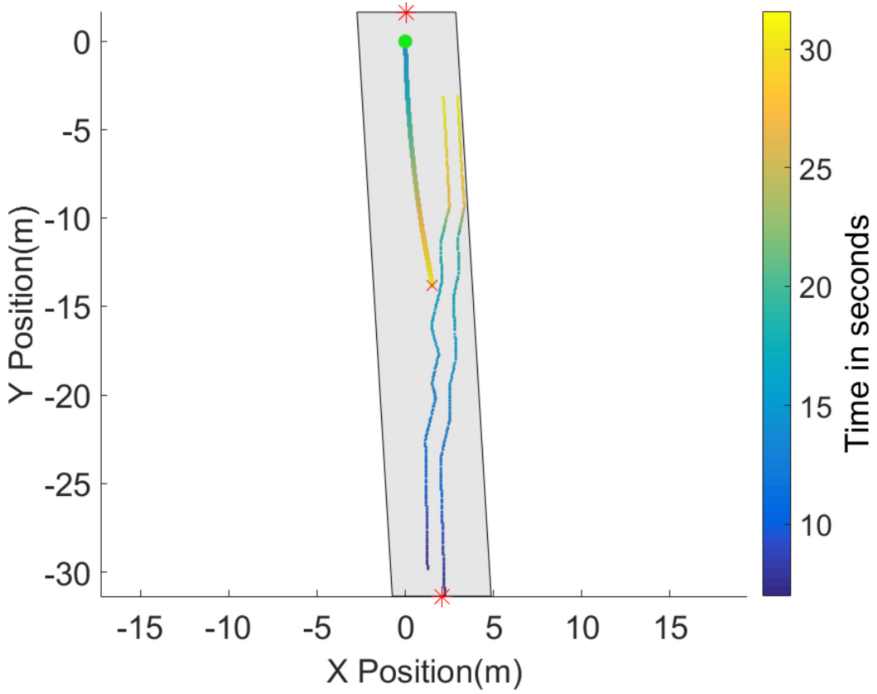


Fig. 6. Time-plot of two individual characters crossing a vehicle from the front [4].

single virtual pedestrians in a pedestrian zone. A time-plot of the test is shown in Fig. 6.

The gray color presents the pedestrian zone where the virtual pedestrians cannot strictly walk out of the area. The lower and upper red stars are the start and goal static waypoint respectively. The green dot and red cross is the start and goal point of the vehicle respectively. It can be noticed that as the characters walk towards the goal SW, it can realize the approaching vehicle from the front. The characters do not wait for the vehicle to get near rather they in advance start making way for the vehicle to pass. In this test, the vehicle was intentionally steered towards virtual pedestrians to analyze their behavior. The more vehicle is steered towards the characters, the more characters go away from the path. Both virtual pedestrians adopt almost the same pattern.

5.2 Group of Virtual Pedestrians

As defined earlier, group of virtual pedestrians are number of pedestrians which remain together. In order to analyse the grouping behavior of characters, two different situations are considered.

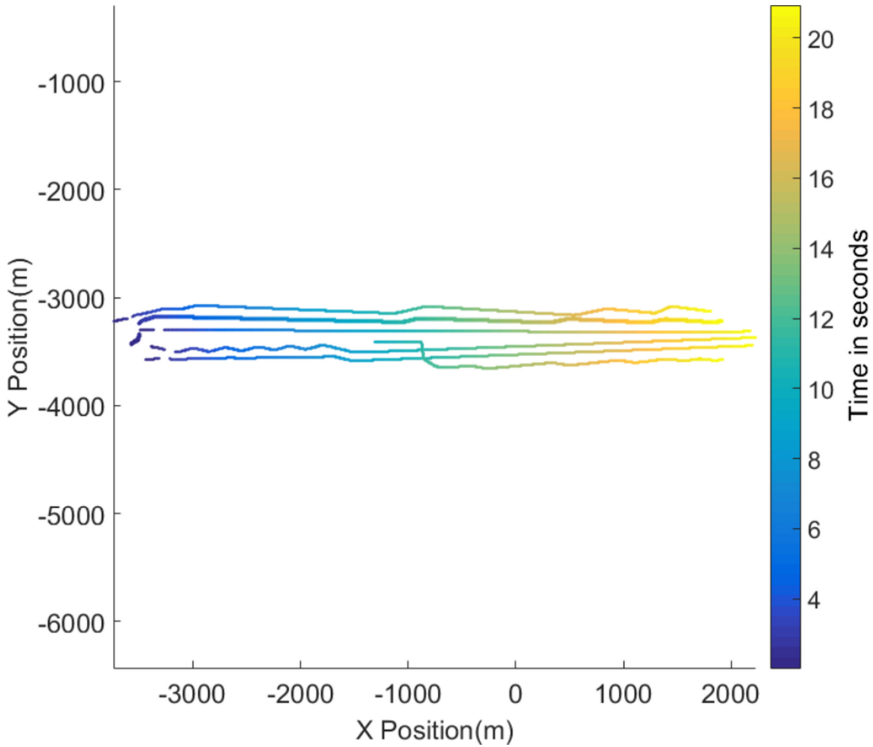


Fig. 7. Group of characters walking towards the goal without obstacle.

Group of Characters Without Obstacle. Firstly, group of six virtual pedestrians are spawned in an open environment as can be seen in Fig. 7. These virtual pedestrians simply walk from start to goal SW. It can be observed that one virtual pedestrian at 12s waits for the whole group. As the group approaches, the character starts walking along. There is also randomness in their path. This is also realistic since people sometimes change position during their walk.

Group of Characters with Obstacle. Secondly, group of five virtual pedestrians are spawned in open environment. However, there is a static obstacle in red shown in Fig. 8. As the virtual pedestrians are spawned, they directly walk towards the corner of the obstacle. Despite having an obstacle in their path, the virtual pedestrians remain in a tight group and in between they change their position. This kind of stochasticity of individual pedestrian in the group is routinely observed in real world situation.

5.3 Comparison with Real World Pedestrians

To validate our system, we compare it with the real world pedestrians. Our campus is modeled in the simulation as can be seen in Fig. 9. The same area in

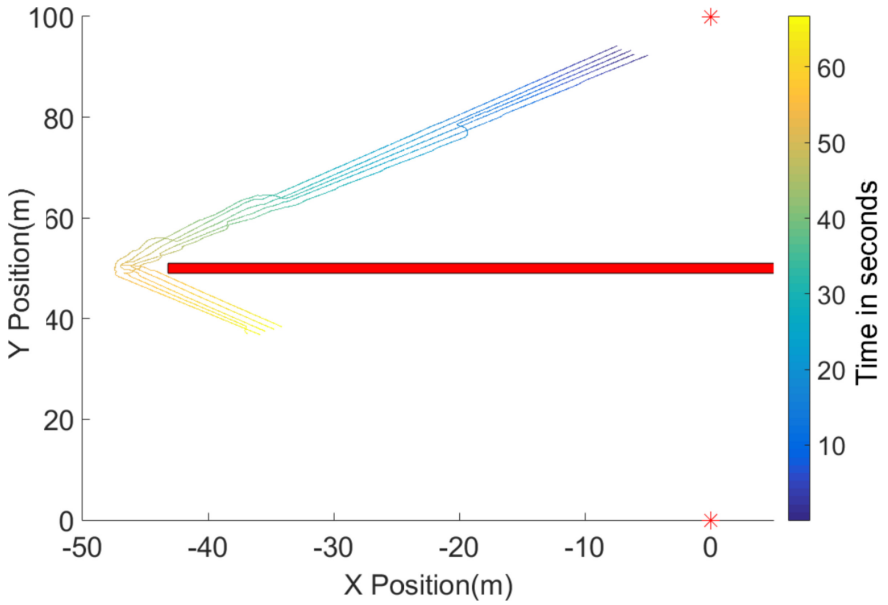


Fig. 8. Group of characters walking towards the goal with obstacle [4].

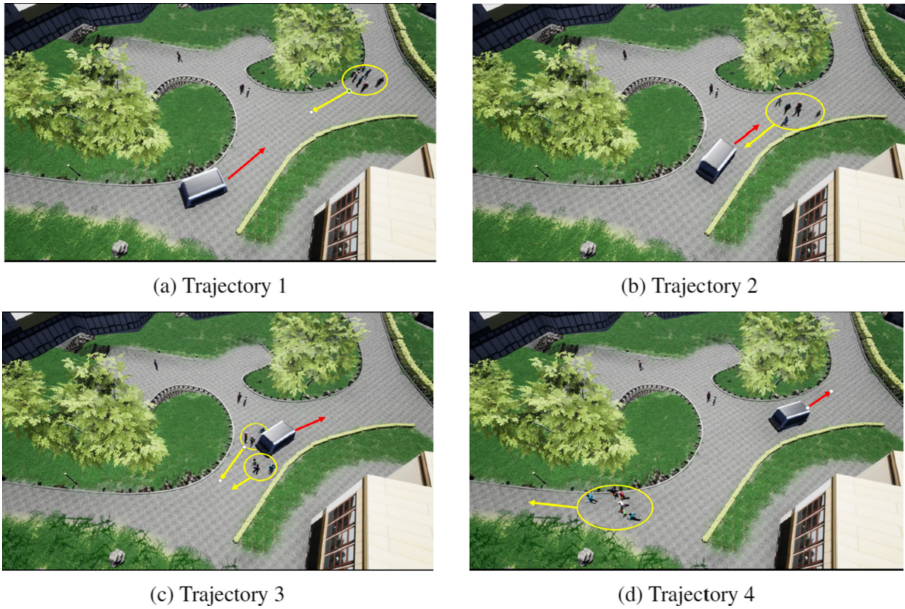


Fig. 9. Group of six characters walking towards an approaching vehicle in a narrow pedestrian zone [4].

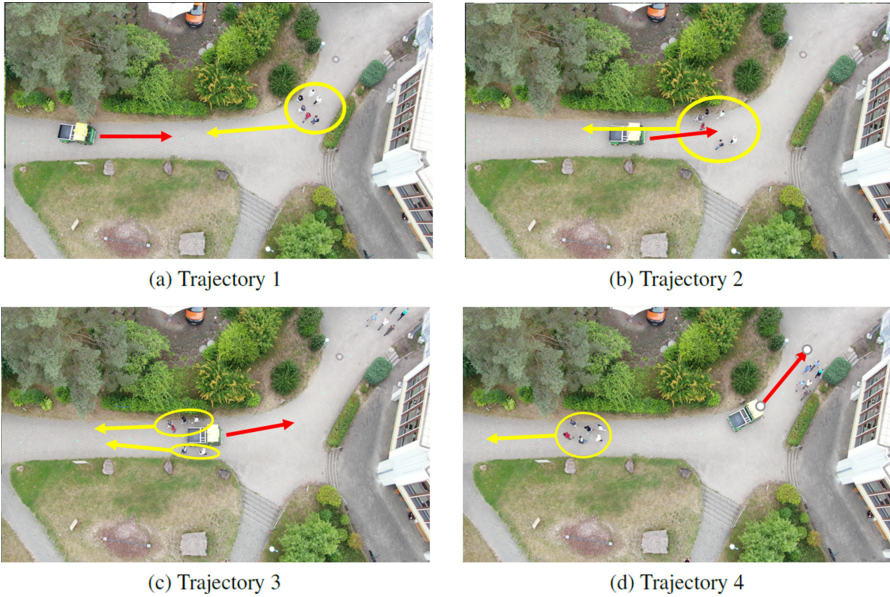


Fig. 10. Group of six pedestrians walking towards an approaching vehicle in a narrow pedestrian zone.

the real world can be seen in Fig. 10. Initially, testing is done in simulation and later the same scenario is re-created in the real-world. Figure 9 shows a group of six characters walking towards a goal. A vehicle is made to pass through this group to see a splitting behavior. In Fig. 9a, the yellow circle shows characters walking in a group in one direction. As the vehicle approaches in Fig. 9b, the group starts to split in advance to give way to the vehicle. Since the path has a small width and the vehicle is in the center, it is rational to give way to the vehicle by splitting. This phenomenon is seen in Fig. 9c. As the vehicle goes away, the group joins together.

In real-world experimentation, the same location and the same number of pedestrians are told to walk towards the same goal point as in simulation. It can be seen from the Fig. 10 that as the vehicle approaches, the pedestrians show the same behavior done in simulation. Although they do not cross the vehicle exactly at the same location neither do they split exactly in the same way as in simulation but overall behavior remains the same. It is impossible to recreate the same situation from simulation to real-world and vice versa. From Fig. 10a, it is clear that the group pedestrians and vehicle are approaching each other. The vehicle in this experiment is driven by a driver. In Fig. 10b, the pedestrians become aware of the vehicle and start splitting. Figure 10c shows almost similar trajectories as Fig. 9c in simulation. And finally, all the pedestrians get together in the last image.

5.4 Comparison with RVO

This paper also compares our virtual pedestrians with characters using RVO offered by Unreal Engine. The comparison can be seen in the plot shown in Fig. 11. Two virtual pedestrians of our system and two characters using RVO are made to cross each other. The above trajectories crossing each other is from our virtual pedestrians and the lower two trajectories from the RVO method. Our system equally shares the same responsibility to avoid each other, whereas, in RVO method one of the virtual pedestrian changes its trajectories. The other one does not even slightly change.

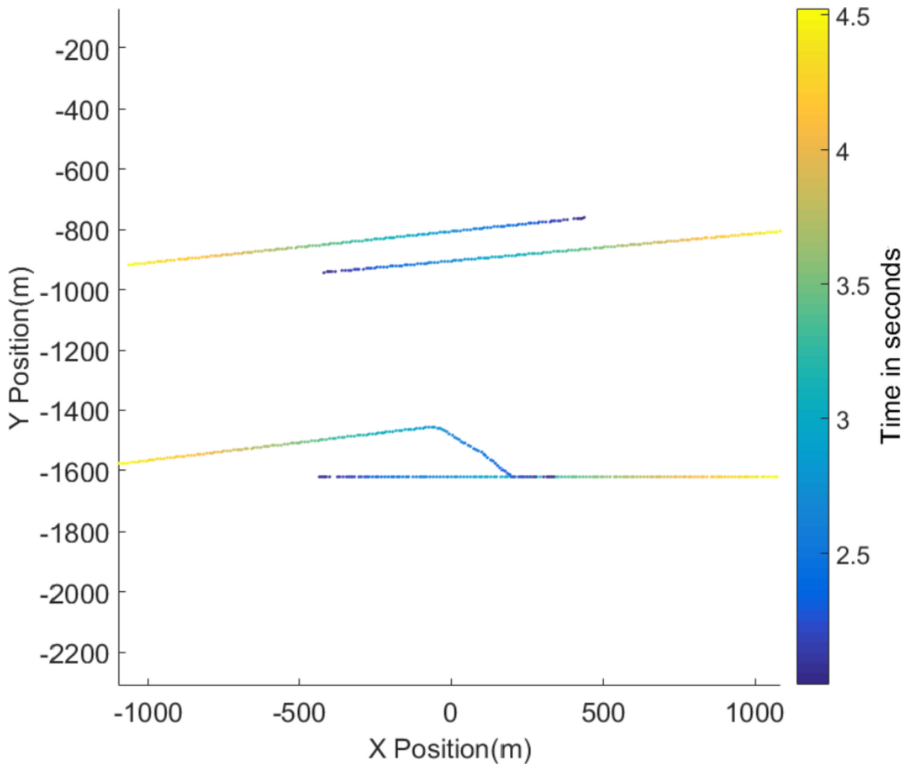


Fig. 11. Group of six pedestrians walking towards an approaching vehicle in a narrow pedestrian zone.

6 Conclusion

This paper proposes and implements virtual pedestrians in simulation, which is easily configurable for continuous testing. The generated virtual pedestrians possess realistic characteristics in terms of interacting with the vehicle and other pedestrians. These interactions include avoiding vehicles, crossing vehicles from

a safe distance, going nearby vehicles when there is no space. From the experiments, it can be inspected that characters for different instances show reasonable behaviors, which is also expected in the real-world. The plots in Fig. 6 demonstrates that the characters distinguish between static and dynamic obstacle by giving way beforehand. The virtual pedestrians in plot 8 show the grouping behavior and how important it is to remain together to form the group. Also, the group passes around the obstacle to follow the shortest distance towards the goal. These behaviors are normally expected in humans when they want to cross a vehicle or wall in their path. Finally, a comparison with the real-world situation is done in Fig. 9 where the same scenario was recreated from Fig. 10. Both show very matching results. There is a number of parameters to create each scenario in a simple manner. This makes the testing of autonomous vehicles fast and easy.

The shortcomings in the given approach are that the virtual pedestrians cannot perform very detailed animations like grabbing the children when the vehicle is nearby. Humans use other gestures for communication to find a consensus for their preference while crossing other people or vehicles. The virtual pedestrians in the proposed system only uses visual aid to avoid obstacles. They can be enhanced by adding hearing aid and signaling gestures as well in future work. The combined effect of these aids can help in better realizing the interaction and trajectory between the characters and the vehicle. Also, these pedestrians are ignorant of the weather conditions. For example, if the rain starts, the pedestrian would start running or if there is strong sunlight then the pedestrians would prefer to walk in the shade even if the route becomes long enough.

References

1. EasyMile Homepage. <https://easymile.com/>
2. Navya Homepage. <https://navya.tech/en/usecases/city/>
3. <https://www.tampabay.com/news/perspective/a-google-cars-crash-shows-the-real-limitation-of-driverless-vehicles/2267980/>
4. Jan, Q.H., Kleen, J.M.A., Berns, K.: Self-aware pedestrians modeling for testing autonomous vehicles in simulation. In: VEHITS, pp. 577–584 (2020)
5. Jan, Q.H., Klein, S., Berns, K.: Safe and efficient navigation of an autonomous shuttle in a pedestrian zone. In: Berns, K., Görges, D. (eds.) RAAD 2019. AISC, vol. 980, pp. 267–274. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-19648-6_31
6. Turner, S., Sandt, L., Toole, J., Benz, R., Patten, R.: Federal Highway Administration University Course on Bicycle and Pedestrian Transportation. Publication No. FHWA-HRT-05-133. US Department of Transportation (2006)
7. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: an open urban driving simulator. arXiv preprint [arXiv:1711.03938](https://arxiv.org/abs/1711.03938) (2017)
8. EasyMile Homepage. <https://www.unrealengine.com/en-US/>
9. Molchanov, V.V., Vishnyakov, B.V., Vizilter, Y.V., Vishnyakova, O.V., Knyaz, V.A.: Pedestrian detection in video surveillance using fully convolutional YOLO neural network. In: Automated Visual Inspection and Machine Vision II, vol. 10334, p. 103340Q. International Society for Optics and Photonics (2017)

10. Sun, X., Wu, P., Hoi, S.C.H.: Face detection using deep learning: an improved faster RCNN approach. *Neurocomputing* **299**, 42–50 (2018)
11. Moussaïd, M., Helbing, D., Theraulaz, G.: How simple rules determine pedestrian behavior and crowd disasters. *Proc. Nat. Acad. Sci.* **108**(17), 6884–6888 (2011)
12. Lake, A.T.: Reciprocal collision avoidance and navigation for video games (2012)
13. Taherifar, N., Hamedmoghadam, H., Sree, S., Saberi, M.: A macroscopic approach for calibration and validation of a modified social force model for bidirectional pedestrian streams. *Transportmetrica A Transp. Sci.* **15**(2), 1637–1661 (2019)
14. Yao, Z., Zhang, G., Dianjie, L., Liu, H.: Learning crowd behavior from real data: a residual network method for crowd simulation. *Neurocomputing* **404**, 173–185 (2020)
15. Perez, D., Hasan, M., Shen, Y., Yang, H.: AR-PED: Augmented Reality Enabled Pedestrian-in-the-Loop Simulation. No. 19–06083 (2019)
16. Alghodhaifi, H., Lakshmanan, S.: Simulation-based model for surrogate safety measures analysis in automated vehicle-pedestrian conflict on an urban environment (Conference Presentation). In: *Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2020*, vol. 11415, p. 1141504. International Society for Optics and Photonics (2020)
17. Ishaque, M.M., Noland, R.B.: Behavioural issues in pedestrian speed choice and street crossing behaviour: a review. *Transp. Rev.* **28**(1), 61–85 (2008)
18. Willis, A., Gjersoe, N., Havard, C., Kerridge, J., Kukla, R.: Human movement behaviour in urban spaces: implications for the design and modelling of effective pedestrian environments. *Environ. Plan. B Plan. Des.* **31**(6), 805–828 (2004)
19. Harrell, W.A.: Factors influencing pedestrian cautiousness in crossing streets. *J. Soc. Psychol.* **131**(3), 367–372 (1991)
20. <https://www.anylogic.com/airports-stations-shopping-malls/>
21. <http://www.vadere.org/>
22. <https://www.oasys-software.com/products/pedestrian-simulation/>