Self-aware Pedestrians Modeling for Testing Autonomous Vehicles in Simulation

Qazi Hamza Jan, Jan Markus Arnold Kleen and Karsten Berns

Robotics Research Lab, Technische Universität Kaiserslautern, 67663 Kaiserslautern, Germany

Keywords: Pedestrian Simulation, Pedestrian Path, Autonomous-vehicles, Testing, Virtual Pedestrians.

Abstract: With the rise of autonomous vehicles in the urban environment, the focus is also shifted towards autonomous vehicles in pedestrian zones. Pedestrian safety becomes the primary concern in such zones. Autonomous systems for these situations need thorough testing before its deployment in the real-world to ensure safety. Therefore, developing testbeds that resemble the real-world for autonomous vehicles testing in pedestrian zones are highly critical. The proposed work focuses on the modeling of pedestrian behaviors in a simulated environment for realizing autonomous vehicles in the pedestrian zones. The virtual pedestrians are modeled with the self-awareness to avoid static and dynamic obstacles when progressing towards its goal. The goal is also to have a minimum number of parameters to generate various test scenarios with realistic behaving pedestrians for the autonomous systems. The proposed system is evaluated using individual and group of virtual pedestrians. It can be seen from the experiments that simulated pedestrians show trajectories which resemble the trajectories of pedestrians in the real-world for that particular situation.

1 INTRODUCTION

The development of autonomous commute for the pedestrian zone is on the rise. Many companies are producing vehicles such as Navya [1] and Easymile [2] to provide solutions for transportation in pedestrian zones. Pedestrian zones are public areas that include vast tourist attractions, airports, university campuses, and so on. Because of environmental, social, and economic aspects, transportation authorities are aiming at turning existing streets into pedestrian zones (Diēinait-Rauktien et al., 2018). However, walking such long distances becomes tiring, especially for the elderly and disabled people. Therefore, there is a need to develop a safe and reliable transportation system, such as driverless shuttles, that can carry passengers without disrupting pedestrians in such zones. The most critical aspect of an autonomous vehicle in these areas is the safety of the pedestrian. Safe navigation becomes critical because of the random behavior of pedestrians. Since pedestrians can include juveniles, seniors, and children, therefore, the autonomous transportation system should make appropriate decisions according to the type of pedestrian.

Proposed approaches for autonomous navigation in the pedestrian zones should aim at guaranteeing performance and, more importantly, safety. Therefore, it should be tested thoroughly to drive safely among people. Generally, researchers develop autonomous transportation systems in a simulated environment to avoid any potential risks towards property or the people. Despite the benefits of developing vehicles in simulation, virtual pedestrians hardly represent the actual behavior of real-life people. Humans express a variety of behaviors according to their circumstances and personalities. Some of these behaviors can be instinctive, e.g., avoid collisions, and some behaviors are goal-oriented, e.g., take the shortest path. Additionally, some behaviors are based on social norms, such as friends or families walking right next to each other. Hence, these real-life behaviors associated with different situations should also be incorporated into the simulation to develop a realistic scenario in simulation.

The main contribution of this paper is to generate virtual pedestrians, called characters, that depict real-life behaviors of pedestrians in pedestrian zones. Every character which is spawned in the environment is given knowledge about its surrounding. From this knowledge, the characters are capable of reaching their goal position by avoiding obstacles and vehicles on their path. There are some user given parameters

[1]https://navya.tech/en/

[2]https://easymile.com/

Jan, Q., Kleen, J. and Berns, K.
Self-aware Pedestrians Modeling for Testing Autonomous Vehicles in Simulation.
DOI: 10.5220/0009377505770584
In *Proceedings of the 6th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2020)*, pages 577-584
ISBN: 978-989-758-419-0

to make the characters not to dodge the vehicles for exceptional testing and validation of autonomous vehicles themselves. The implementation is done in Unreal Engine [3].

In the next section, related work is presented, which discusses similar pedestrian simulations already existing. Section 3 gives the overall architecture of the system. In this section, the main components of the proposed system are summarized. In section 4, the system approach is discussed in detail. Finally, the results of the experiments are evaluated in section 5.

2 RELATED WORK

The study of autonomous vehicles in the pedestrian zones is growing. Researchers are developing interaction strategies to communicate with pedestrians. Jan et al. (Jan et al., 2020) have developed interaction modules to indicate pedestrians for avoiding collisions with vehicles in advance. The authors have used simulation to examine their results from the vehicle's perspective. Similarly, Rasouli and Tsotsos (Rasouli and Tsotsos, 2019) have surveyed to identify factors that affect the behavior of pedestrians and how to solve interaction problems. They mention various design approaches to see the intentions of pedestrians. For a thorough analysis of vehicle crowd interaction in an urban environment, Yang et al. (Yang et al., 2019) have proposed pedestrian trajectory datasets to validate pedestrian motion models influenced by the vehicles. The authors collect the trajectories of pedestrian and vehicle from a down-facing camera attached to a drone. It particularly helps the community, which is closely working in the interaction between pedestrians and vehicles.

There exists a well-known open-source simulation environment, Carla (Dosovitskiy et al., 2017), which is developed for autonomous driving research. Besides vehicle driving and town modeling, Carla also offers virtual pedestrians that navigate by location-based cost. This cost encourages the pedestrian to walk specifically on footpaths and main crossings. However, Carla does not offer pedestrian behavior explicitly for vast pedestrian zones in which pedestrians are not bounded by footpath like structures but can walk freely along the stretch of the pedestrian zone as individuals or groups. Another simulation system specifically for representing the crowd is given in (Kimura et al., 2019), in which the authors are using a multi-agent model for representing the crowd in the SimTread. They have mainly analyzed evacuation cases and portray behaviors in bottleneck areas. SimTread allows user-definable pedestrians and spatial models. The overhead of using SimTread for any environment is to have a spatial model of the scene.

Many different approaches have been implemented for modeling character movements. Mehdi Moussaïd et al. (Moussaïd et al., 2011) have addressed the issue of crowd disaster and modeled the pedestrian flow in simulation. Authors have used a cognitive approach to adapt the walking speed and direction of characters giving way to realistic modeling of collective social behaviors. It uses vision to perceive the environment, walking speed, and direction as heuristics. A local method for collision avoidance between characters has been done in (Karamouzas et al., 2009), in which the collision is predicted, and a smooth trajectory is calculated. The authors have modeled the pedestrian as a disk on a 2D plane with rotation. It mainly uses forces to calculate the collision-free path until the goal position is reached. In one of the recent work, data-driven approach has been presented (Martin and Parisi, 2019) in which experimental data has been collected from motion capture sensors and fed into a generalized regression neural networks. However, authors assume obstacle avoidance between one pedestrian and a fixed obstacle. A classical approach for avoiding collision with other bodies is presented in (van den Berg et al., 2011). Their approach is based on velocity obstacle (Fiorini and Shiller, 1998) for the collision-free motion for different bodies. In most recent works, Yin et al. (Yin et al., 2019) have used this approach for virtual pedestrian simulation. However, the authors have not differentiated pedestrian behavior for the case of vehicles moving along their path. Inspired by pedestrian dynamics using social force models in (Helbing and Molnar, 1995),(Helbing et al., 2000), (Alahi et al., 2016), and (Cosgun et al., 2016), authors of (Chao et al., 2019) also use scalable force-based framework for simulating pedestrians bicycle and vehicles. Their forces are dependent on the structured environment, such as lanes and crossings. Our system offers a solution for unstructured pedestrian zones where exceptional vehicles can also navigate.

3 SYSTEM ARCHITECTURE

The architecture of the proposed system is designed to create several characters in the environment with their behaviors. It also generates different scenarios with a minimal number of parameters for testing different aspects of autonomous vehicle systems. In pedestrian zones, plenty of behaviors are exhibited by the

[3]https://www.unrealengine.com/

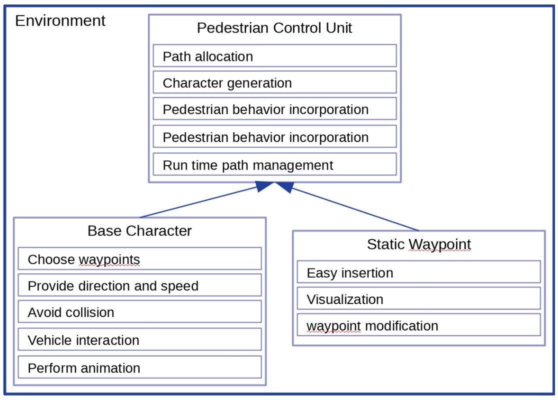


Figure 1: The overall architecture of the system. It consists of three main modules in which every module defines its function.

pedestrians, for instance, crossing in close vicinity of the vehicle, walking in different group sizes in which each group member has his distinct behavior, and so on. Producing each of these behaviors for prolonged test cases in the simulation is cumbersome. Therefore, the system implemented in this paper also focuses on giving the user a simple API for generating a variety of scenarios for testing interactivity between pedestrians and autonomous vehicles. Some of these parameters are included in the 4.6. The system is divided into three main components; Pedestrian Control Unit (PCU), Static Waypoint (SW), and Base Character (BC). Figure 1 shows the system architecture with three components and their respective properties. Each component is briefly explained in the following sub-sections.

3.1 Pedestrian Control Unit

PCU is designed for the systematic execution of the characters and their behaviors. PCU manages all the characters from spawning to despawning in the environment. PCU directly provides access to different parameters, which can be defined by the user to generate different behaviors for characters. These parameters are briefly given in 4.6. PCU entirely handles all the work, from path making to managing characters on these paths. The process is as follows:

- In the start of the simulation, PCU spawns the characters in the environment. The spawning could be as individual characters or group of characters depending on the input from the user.
- Characters get a random path assigned to it and spawned at the first SW of the given path.
- The PCU also assigns different options given by the user, e.g., number of characters, their group size, and their speed.
- Assigning animations for each character, such as talking on the phone, and texting.

3.2 Base Character

The entity to show the real-world pedestrians are the BCs. BCs are the set of virtual pedestrians that move along the path. The unreal engine includes highly-featured Character class, which makes it walk, jump, run, messaging, and talking on the phone. These particular set of animations plays an important role in behavior and activity recognition of pedestrians in the real world. A different attribute of characters such as male/female, child/young, different complexions, and size are included in order to fully exploit the perception system of autonomous vehicles. Each BC has its individual movement. Obstacle avoidance is implemented in BC. Therefore, each character uses its perception to avoid an obstacle. Based on the available path, it decides for its speed and direction.

3.3 Static Waypoint

The path must be defined to have a meaningful strategy for every BCs. It is done by placing SWs where the characters are intended to walk. SW is the fundamental component for making paths. Each BC starts from the initial SW and traverses all the SWs in a path until it has reached the endpoint. At this point, either it can select a new path or despawn based on the information from the PCU. SW can be placed anywhere in the environment. However, for testing the algorithms in an autonomous vehicle, SWs are placed at the same locations where the vehicle has to navigate. By slightly changing the location and number of SW, different interactions between characters and vehicles can be obtained. It is because characters need to pass through every SW once, and if placed at the same place as the vehicle, the characters can have a waiting behavior until the SW is available to pass.

4 PROPOSED APPROACH

The main idea of this work is to provide real-life pedestrian behaviors to characters in the simulation environment for testing of autonomous systems. These behaviors include moving towards a goal on a suitable path and avoiding obstacles or vehicles on the same path. For this reason, there is a need to define heuristics for every character based on user parameters. The approach builds on providing the surrounding knowledge to every character and defining the set of rules for speed and direction.

Since the path for the characters is defined at the start of simulation by giving SWs, the focus here is on collision avoidance, local pathfinding, and random behaviors. Each character possesses its behavior and visualization so that they depict their particular behavior. Additionally, the characters are also modeled as individuals and in groups and how it reacts to autonomous vehicles. In this paper, every walking character gets some features for a realistic reaction to autonomous vehicles. These features are awareness, direction heuristics, speed heuristics, SW identification, grouping, and can be modified by user-defined parameters. All these features are explained in the subsection below.

4.1 Awareness

Rationally, it is not possible to avoid obstacles without perceiving the environment. For realistic trajectories of characters, every character is equipped with, at least, the basic ability to understand the environment. It includes the ability to know if a moving vehicle would be crossing their current direction. Additionally, characters have the knowledge if the vehicle is moving behind them and in the same direction. This knowledge is given by using the "LineTrace" method in Unreal Engine, which returns first hit to an object in the environment, as shown in figure 2. The width of the person is also considered for obstacle avoidance at every detection angle. Within already defined resolution for the field of view, every possible walking direction is checked. The additional direction towards the target is considered in order to avoid the zig-zag movement.

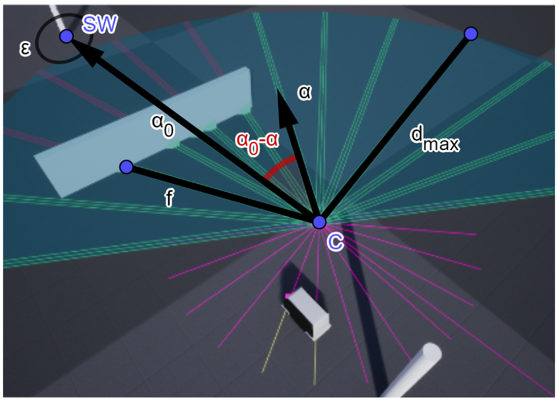


Figure 2: Visualization of a pedestrian c and its awareness in simulation. α_0 directs towards the goal position. Green area outlines the front of pedestrian whereas pink lines show the rear of pedestrian, mainly, for capturing approaching vehicle from behind.

4.2 Direction Heuristics

Every t seconds, a character has to re-evaluate their direction. With the given information they can choose between a certain, finite amount of directions. A distance value for each direction is calculated, based on the given information and how direct the target is. Then the direction with the smallest value is chosen. This paper takes the minimization problem from (Moussaïd et al., 2011) and modify and extends it:

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

Where d is the (weighted) distance in direction α between two end points which is shown in figure 3; α_0 is the direction of the destination point; $f(\alpha)$ is the distance to first collision in direction α; and d_{max} is the maximal viewing distance of a character.

First, d_{max}^2 is removed because it does not have an effect on which direction has the 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 $(\alpha_0 - \alpha) > 90°$, is larger than $d(\alpha_0)$. But characters should be able to avoid not only very small but also bigger obstacles. 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 $(\alpha_0 - \alpha) > 90°$ 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 equation 1 is modified to equation 2.

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

Subsequently, characters should avoid walking in front of or crossing paths with a vehicle, so the value for these directions is increased by adding $f(\alpha)^2$ to equation 2. In other words, the distance to the collision is valued double. The reason to double is to avoid the vehicle in advance, which ensures that this direction is not chosen. However, if all other directions have a terrible distance to the collision, it still can be chosen. If such a direction is chosen, characters may want to have a different walking speed.

4.3 Speed Heuristics

After choosing a direction, characters have to choose their walking speed. A constant minimum time to collision t_{coll} is set and used to calculate the character's speed. A preferred walking speed of an individual,

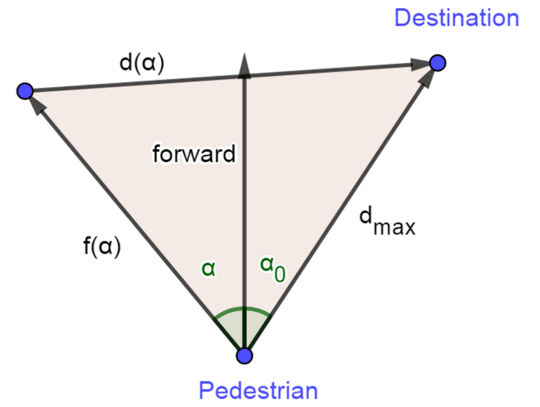


Figure 3: The law of cosines applied to character movement.

v_{pref}, is set by the user and used as an upper bound. Equation 3 shows the speed of the character.

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

Since t_{coll} is constant and positive, the characters can never walk against any object. The next contribution of this system is that the characters of this paper should also try to avoid getting hit by vehicles. If the chosen direction crosses the current direction of a near moving vehicle, the character waits for the vehicle to pass or stop, just like pedestrians crossing a road in real life. For directions crossing a near moving vehicle, $v = 0$, else v is chosen according to equation 3.

4.4 SW Identification

A character constantly checks if it has reached its current SW in the path. Reaching means the distance to the SW is smaller than a distance ε (radius define for every SW). When this happens, the character either sets the next SW in its path as their current SW or it has finished its current path. In the latter case, the character either despawns from the environment or queries for a suitable new path from the PCU depending on given user settings in the PCU.

It is important to note that SWs should not be placed in a way that they are blocked continuously, but they can be blocked temporarily by an autonomous vehicle. In this case, the simulation increases the distance, ε, temporarily.

4.5 Grouping

One usual occurrence in real life is the pedestrians walking in groups. There is no central decision-maker for the characters in the group. Characters in a group can still walk on their own, and they all have the same

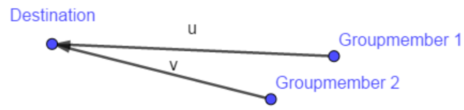


Figure 4: The distance to group for group member 2 is $\|u - v\|$.

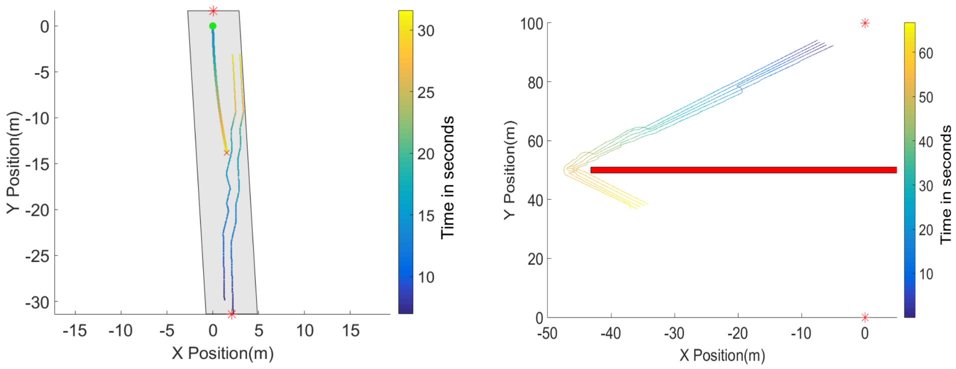
SW at all times, as described previously. However, they need to stay as a group. This requirement is fulfilled by changing the speeds of characters in one group.

First, the preferred walking speeds(ranging between 1.2 m/s to 3.6 m/s) for each character in the group are equal, so without disruption, the group stays together. It can still happen that members of the group get so far away that it would not resemble a group anymore. In this case, a single character needs to wait for the rest of the group. Characters need to determine when they should wait. Normally, people in a group stay close together but are also comfortable with their personal space. This concept is realized by using two constants, a preferred and a maximum distance to the group. Distance to a group, in this context, is measured by the difference in the personal distance to the current SW and the maximum distance to the current SW in the group, as shown in figure 4. Distance is measured in this way to allow a group to split up in order to walk around an object. This check always guarantees that at least one member of the group is not waiting for the group, hence, avoiding any deadlocks caused by this functionality. If the distance between the character and its group is greater than the threshold distance, then the character waits for the group to join by looking towards the rest of the group, signaling they are waiting for them. As the distance to the group becomes smaller or equal to the preferred distance, they start walking at their regular speed. A maximum distance is an upper bound to the preferred distance to the group.

4.6 User Defined Parameters

To generate each BC with different behavior is reflected with the use of behavioral options that can either apply to all or random characters. These options are set by the PCU when spawning a BC according to what parameters are set in the PCU.

As an example, one option that can be randomly assigned to every character, which can be described as irrational or "childlike", is the ignoring of vehicles. It can be split into two different options. First, ignoring if a direction crosses the path of a vehicle when choosing it and second, the option to not wait for vehicles. The first only affects the direction, the second



(a) Scenario 1 (b) Scenario 2

Figure 5: Two 2D-time plots showing the trajectory of characters. Plot(a) has two characters walking in the direction of the vehicle. The start and end of vehicle path is shown with green dot and red cross respectively. Red asterisk shows the SW. Gray area shows the pedestrian path. Plot(b) depicts trajectories of five characters walking in a group avoiding a static obstacle(in red).

only speed. It is implemented to reflect reality, where not all characters behave rationally.

Moreover, there are a series of more technical options. These include the already mentioned maximum group size, the delay between spawns, the distance epsilon for reaching an SW, and if characters should despawn on finishing a path. The rest of the parameters include character speed, which is a range from walking to running, animation chance, for having random animation upon reaching an SW.

5 EXPERIMENTS AND EVALUATION

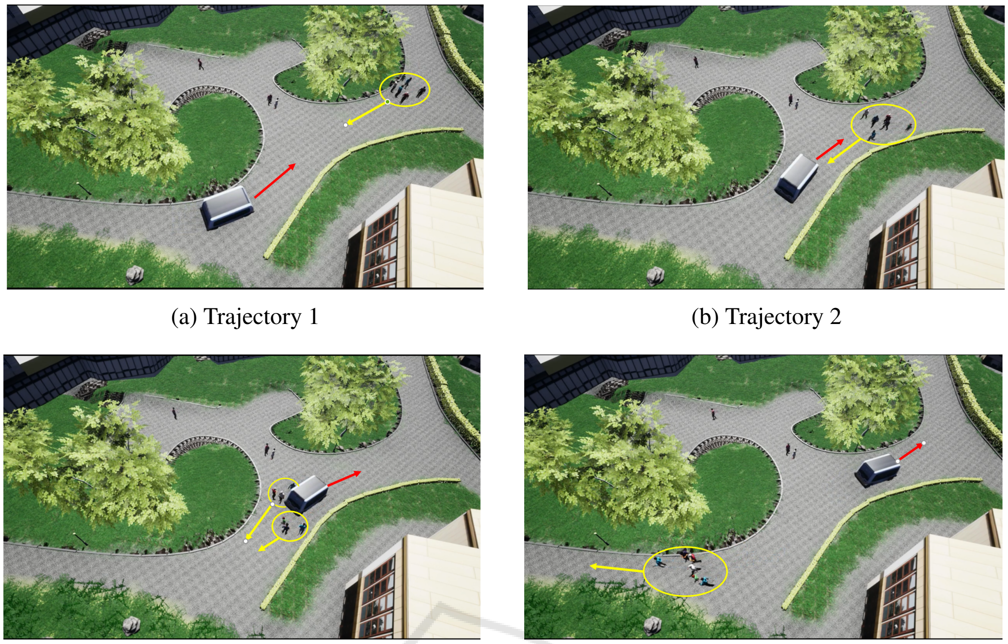
In real-world pedestrian zones, there exists an enormous number of situations where pedestrians behave differently. This variation in behavior depends on the number of pedestrians, their age group, their motives, awareness level, and many more.

To exploit every behavior of pedestrians existing in the real-world is impractical in the length of the paper. For this reason, three particular scenarios are included, which subsume additional behaviors occurring in the pedestrian zone. The scenario consists of a campus-like environment modeled in the simulation where there are no specific road markings, and pedestrians are expected to walk anywhere on the pathway. These scenarios are plotted using 2D time graphs of ground truth values of characters and also, later, by showing simulation images to prove the discernability of the implemented system.

In the first scenario, a vehicle crossing two characters is presented. The plot for this experiment is shown in figure 5a. The two characters are given two SWs (shown with a red asterisk) for the start and goal position. In an ordinary situation where there is no obstacle, the characters walk straight towards the goal position. But as a vehicle approaches towards them, the characters start making way for the vehicle to pass. This behavior can be seen by inspecting the graph from "-23" Y-position onwards. Both characters change the course of their way. It can be seen from the vehicle trajectory with a start as a green dot and end as a red cross that the vehicle is slightly made to steer towards the characters. By doing so, the characters have reached the end of the width of the pedestrian path sketched as a gray parallelogram. In this experiment, both the vehicle and the characters are moving with walking speed.

A group of five characters is designated by defining the parameters in the PCU for the second scenario to evaluate the grouping behavior of the proposed system. To justify their resemblance in their trajectory, a static obstacle is placed between their goal and end positions, as can be seen in figure 5b. The plot shows the trajectory of the group avoiding the obstacle. From the trajectories of every character, it can be seen how closely they remain throughout their path. This behavior defines the characteristics of a group. At some point in the plot, the uniformity of the group is lacking. However, considering the same in a real-world scenario, uniformity does not always exist. People incline in different directions within the group. For example, if someone stops, the person behind would try to cross him/her.

The final scenario is to illustrate the behavior of the group for a crossing vehicle. It is shown by using images directly from our model in the simulation. The environment is shown in figure 6. The trajectory traversed by each character in the group clarifies



(a) Trajectory 1 (b) Trajectory 2

(c) Trajectory 3 (d) Trajectory 4

Figure 6: Group of six characters walking towards an approaching vehicle in a narrow pedestrian zone.

that giving way to an oncoming vehicle is more important than staying in a group. The vehicle is made intentionally to move among the group to observe the group's reaction. The characters are closely grouped when the vehicle is far away from them, as can be seen in figure 6a. As the vehicle approaches towards the group of characters, figure 6b shows the characters start making space for the vehicle to pass. Since there is no other way for the vehicle to evade the whole group of characters from both the sideways, the group divides in a reasonable way to avoid confusion, as can be viewed in figure 6c. Finally, figure 6d spots that the group integrates back once there is no other obstacle on the way.

The system validation can be done by comparing the trajectories from the plots in this section to real-world pedestrians. In a real-world situation, similar characteristics of the splitting of groups can be seen. However, the trajectories of pedestrians may differ for every individual. Our experiments show trajectories similar to shown in(Appert-Rolland et al., 2018), who have reported the study of pedestrian trajectories as individuals and groups. They have shown pattern formation of a pedestrian in a different medium of crowd density.

6 CONCLUSION

This paper proposes and implements virtual pedestrians in simulation, which carry real-world pedestrian like behavior. 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 figure 5, demonstrate that the characters distinguish between static and dynamic obstacle. The characters in plot 5a constantly change in their trajectory as the vehicle approaches them, but the characters in plot 5b pass 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.

It is understood that humans use different gestures for communication to find a consensus for their preference while crossing other people or vehicles. The characters in the proposed system only use 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.

REFERENCES

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. (2016). Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971.

Appert-Rolland, C., Pettré, J., Olivier, A.-H., Warren, W., Duigou-Majumdar, A., Pinsard, É., and Nicolas, A. (2018). Experimental study of collective pedestrian dynamics. *arXiv preprint arXiv:1809.06817*.

Chao, Q., Jin, X., Huang, H.-W., Foong, S., Yu, L.-F., and Yeung, S.-K. (2019). Force-based heterogeneous traffic simulation for autonomous vehicle testing. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8298–8304. IEEE.

Cosgun, A., Sisbot, E. A., and Christensen, H. I. (2016). Anticipatory robot path planning in human environments. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 562–569. IEEE.

Diēinait-Rauktien, R., Valēiukien, J., Parsova, V., and Maliene, V. (2018). The importance of environmental criteria for kaunas city pedestrian zones. *Opportunities and Constraints of Land Management in Local and Regional Development: Integrated Knowledge, Factors and Trade-offs*, page 133.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16.

Fiorini, P. and Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772.

Helbing, D., Farkas, I., and Vicsek, T. (2000). Simulating dynamical features of escape panic. *Nature*, 407(6803):487.

Helbing, D. and Molnar, P. (1995). Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282.

Jan, Q. H., Klein, S., and Berns, K. (2020). Safe and efficient navigation of an autonomous shuttle in a pedestrian zone. In Berns, K. and Görges, D., editors, *Advances in Service and Industrial Robotics*, pages 267–274, Cham. Springer International Publishing.

Karamouzas, I., Heil, P., van Beek, P., and Overmars, M. H. (2009). A predictive collision avoidance model for pedestrian simulation. In Egges, A., Geraerts, R., and Overmars, M., editors, *Motion in Games*, pages 41–52, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kimura, T., Sano, T., Hayashida, K., Takeichi, N., Minegishi, Y., Yoshida, Y., and Watanabe, H. (2019). Representing crowds using a multi-agent model–development of the simtread pedestrian simulation system. *Japan Architectural Review*, 2(1):101–110.

Martin, R. F. and Parisi, D. R. (2019). Data-driven simulation of pedestrian collision avoidance with a nonparametric neural network. *Neurocomputing*.

Moussaïd, M., Helbing, D., and Theraulaz, G. (2011). How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences*, 108(17):6884–6888.

Rasouli, A. and Tsotsos, J. K. (2019). Autonomous vehicles that interact with pedestrians: A survey of theory and practice. *IEEE Transactions on Intelligent Transportation Systems*.

van den Berg, J., Guy, S. J., Lin, M., and Manocha, D. (2011). Reciprocal n-body collision avoidance. In Pradalier, C., Siegwart, R., and Hirzinger, G., editors, *Robotics Research*, pages 3–19, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yang, D., Li, L., Redmill, K., and Özgüner, Ü. (2019). Top-view trajectories: A pedestrian dataset of vehicle-crowd interaction from controlled experiments and crowded campus. *arXiv preprint arXiv:1902.00487*.

Yin, Z., Liu, J., and Wang, L. (2019). Less-effort collision avoidance in virtual pedestrian simulation. In *Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science*, AICS 2019, pages 488–493, New York, NY, USA. ACM.