Crowd dynamics modelling

Michał Pałucki Marcin Kosiba Piotr Sokołowski

This report describes the social force algorithm and its implementation in the context of a pedestrian simulation.

1 State of the art

1.1 Crowd dynamics modelling

In recent years, there has been a considerable advancement in the field of crowd simulation models. Microscopic models have witnessed a surge in popularity, displacing macroscopic models based on hydrodynamic principles (e.g. the Pauls model) [6]. The adoption of microscopic models facilitates more detailed simulations, accounting for the distinct behaviour of individual agents.

The Social Force model is an example of such a model, proposing that pedestrian movement can be conceptualised as being subject to ' social forces'. These 'forces' are not directly exerted by the personal environment of pedestrians, but rather represent a measure of the intrinsic motivations of individuals to perform certain actions (movements) [2]. The Social Force model is one of the most widely used crowd simulation models, but it is not the sole example. Other models include the Floor Field model and the Vicsek model.

1.2 Simulation engines

The report delineates the issue of crowd flow on the AGH campus, employing the Social Force model to illustrate the phenomenon. The model under scrutiny offers a precise representation of pedestrian behaviour in that it accurately reflects the tendency of individuals to circumvent collisions with others (as illustrated in fig. 1). In contrast to the tendency to reach one's destination expeditiously, pedestrians appear to prioritise the avoidance of collisions. Furthermore, the Social Force model incorporates the tendency of agents to maintain a safe distance from obstacles, such as lawns or walls, which aligns with empirical observations.

One alternative could be the Pygame library, which has been designed for the creation of games and multimedia applications in Python. Pygame allows for the creation of interactive simulations in which the user can change the simulation parameters in real time. It is particularly well-suited for visualising cellular automata and other discrete systems. However, due to the language used, it is not a very efficient solution.

It is evident that alternative, highly prevalent solutions comprise substantial game engines, such as Unity and Unreal Engine. These offer a substantially elevated degree of functionality in comparison to Pygame; however, they necessitate a more substantial investment of both knowledge and time to master. These tools are of a more intricate nature, yet they facilitate the development of sophisticated simulations and their subsequent visualisation.

It is also pertinent to mention the Godot engine, which is both free and open source, yet offers a plethora of features that are commonly found in paid engines. A notable advantage of the Godot engine is its smaller size compared to Unity or Unreal Engine, while retaining the capability to create sophisticated simulations.

2 Scope of the project

The report delineates the issue of crowd flow on the AGH campus, employing the Social Force model to illustrate the phenomenon. The model under scrutiny offers a precise representation of pedestrian behaviour in that it accurately reflects the tendency of individuals to circumvent collisions with others (as illustrated in fig. 1). In contrast to the tendency to reach one's destination expeditiously, pedestrians appear to prioritise the avoidance of collisions. Furthermore, the Social Force model incorporates the tendency of agents to maintain a safe distance from obstacles, such as lawns or walls, which aligns with empirical observations.

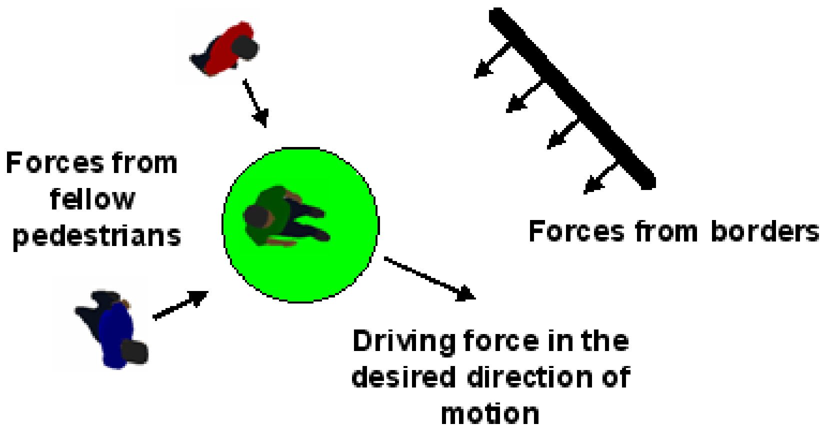


Figure 1: Attributes of the Social Force Model [3]

The focal point of the study pertained to the area demarcated by buildings A-0 and B-5 (cf. fig. 2). The investigation entailed the implementation of a flow simulation comprising 14 groups, with each group consisting of 45 agents. The simulation was executed utilising the Godot game engine.

Figure 2: Map of the AGH campus in Godot engine

The agents are charged with the representation of the university community, and their task is to move from one of the university buildings to the AGH Main Library building. Agents are not permitted to enter the buildings or the green area, but they are permitted to move on the asphalt and pavements (fig. 3). During the simulation, agents must avoid collisions with other agents and obstacles on the road.

3 Material and methods

3.1 Resources

The decision was taken to base the model on the papers ‘Social force model for pedestrian dynamics’ by Helbing and Molnár [2] and ‘How simple rules determine pedestrian behaviour and crowd disasters’ by Moussaïd, Helbing and Theraulaz [4]. In both of these papers, the authors describe a model based on social forces to simulate pedestrian behaviour in crowds. This model is widely regarded as one of the most prominent crowd simulation models, and it possesses the notable capability of accurately representing pedestrian behaviour, including the avoidance of collisions with other pedestrians and

Figure 3: Screenshot of the simulation in Godot engine

the maintenance of a safe distance from obstacles.Furthermore, we have drawn upon a range of additional sources, which are listed in sec. 6.

The model incorporates a series of inputs, including the initial position of the pedestrians, their intended destination, the average velocity at which they move, their mass, the maximum field of view that is available to them, the range of their vision, the time allocated for them to reach their destination, and the location of any obstacles that they might encounter. The outputs of the model include the time taken for the pedestrians to reach their destination, as well as the visualisation of the pedestrian crossing and the formation and dissolution of the crowd. A visualisation of the model is presented in fig. 4.

The model under consideration utilises the following variables:

- i, j - pedestrians,
- W - wall,
- x_i - position vector,
- $r_i = \frac{m_i}{320}$ - radius,
- v_{0i} - pedestrian normal velocity,
- v_i - velocity vector,
- v_{des} - desired velocity,
- m_i - mass of the pedestrian,

- O_i - target point of the pedestrian,
- H_i - range of vision,
- phi - maximum angle of view,
- $\alpha = [-phi, -phi]$ - direction,
- α_0 - target direction,
- α_{des} - desired direction relative to H_i,
- t - relaxation time (time required to change pedestrian behaviour),
- t - time,
- d_{max} - greatest distance seen,
- d_h - distance between pedestrian and obstacle.

The function $f(\alpha)$ returns the distance to an obstacle in the pedestrian's line of sight, depending on the direction. If there is no obstacle in the direction α, the value $d_{\max}$ is returned. The direction $\alpha_{\text{des}}(t)$ is calculated as shown in eq. 1.

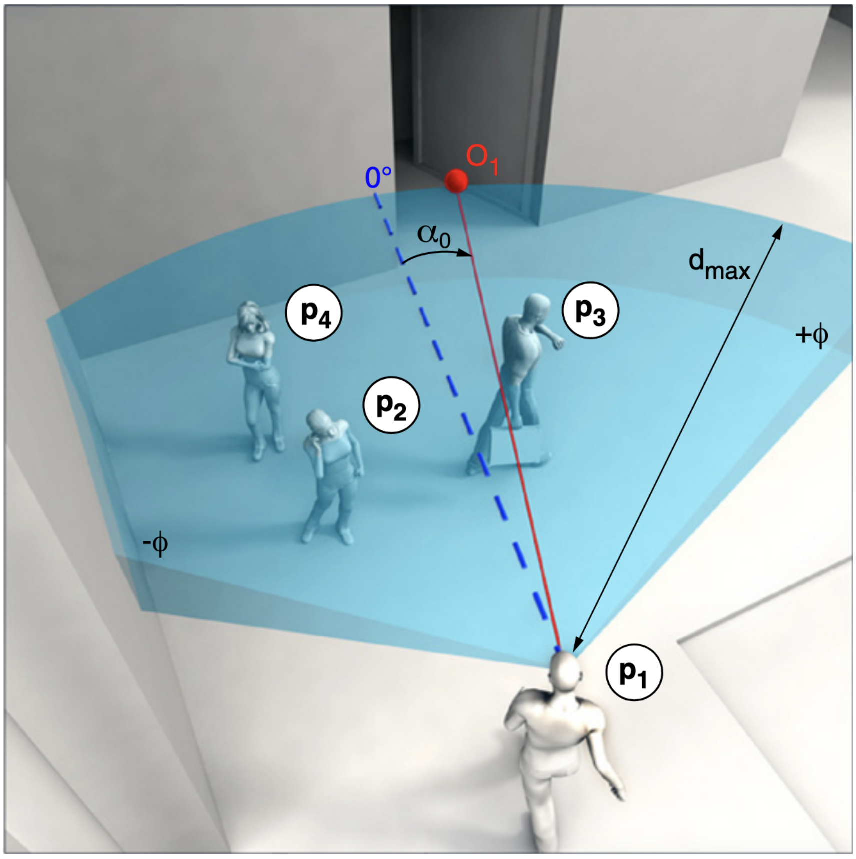


Figure 4: Illustration of a pedestrian p_1 facing three other objects and trying to reach the target point O_1 marked in red; the blue dashed line corresponds to the line of sight [4]

$$
\begin{aligned}
d_{\max} &= H_i \\
d(\alpha) &= d_{\max}^2 + f(\alpha)^2 - 2d_{\max} \times f(\alpha) \times \cos(\alpha_0 - \alpha) \\
\alpha_{\text{des}} &= \min(d(\alpha))
\end{aligned} \tag{1}
$$

The calculation of the desired velocity of the pedestrian is achieved through the utilisation of eq. 2, while the acceleration of the pedestrian is calculated via eq. 3.

$$
v_{\text{des}}(t) = \min(v_{0i}, \frac{d_h}{\tau}) \tag{2}
$$

$$
\frac{dv_i}{dt} = \frac{(v_{\text{des}} - v_i)}{\tau} \tag{3}
$$

In situations of high traffic density, the forces exerted on pedestrians due to direct physical contact (i.e. collision) with other objects must also be taken into account.

The final acceleration vector of the pedestrian can be calculated using eq. 4.

$$
\begin{aligned}
f_{ij} &= k \times g(r_i + r_j - d_{ij}) \times n_{ij} \\
a_i &= \frac{v_{\text{des}} - v_i}{\tau} + \sum_{j, j \neq i} \frac{f_{ij}}{m_i} + \sum_{W} \frac{f_{iW}}{m_i}
\end{aligned} \tag{4}
$$

where:

- f_{ij} - contact force between pedestrians i and j,
- k - scaling factor,
- $g(x)$ - x if pedestrians are touching, otherwise 0,
- d_{ij} - distance between centres of mass of pedestrians i and j,
- n_{ij} - normalised vector from pedestrian j to pedestrian i.

3.2 Framework

In order to enhance the robustness and realism of the simulation, a sophisticated pathfinding algorithm was employed. This approach represented a significant enhancement over the previously discussed methodology that relied on raycasts (see fig. 6). The employment of this sophisticated pathfinding algorithm enabled more efficient and optimal crowd movement, particularly in complex scenarios. The integration of the A* algorithm and Corridor-funnel path post-processing played a pivotal role in the implementation of pathfinding.

In order to leverage pathfinding, it is first necessary to model the Navigation Region. This was created within the environment by combining multiple Navigation Obstacles. In some parts of the Navigation Region, Collision Polygons were utilised to model Bottlenecks more realistically. The collision shapes ensure that each agent has an awareness of its surroundings, thus allowing for smooth navigation through obstacles and other agents.

In the initial stage, the pathfinding algorithm is initiated as soon as an agent is spawned in the simulation. The algorithm calculates a precise path, presented in fig. 5, by dividing the environment into manageable segments. Each agent is then tasked with navigating to the next designated point along this path, ensuring steady progress towards its final destination. This stepwise approach not only simplifies the navigation process but also enhances performance by breaking down complex paths into smaller, more manageable segments.

Figure 5: The initial common path was updated to avoid collisions between agents

Furthermore, the raycast methodology is maintained as an ancillary system with the objective of enhancing interactions between agents and achieving current targets along the designated path. While pathfinding directs agents along optimal routes, raycasts augment their capacity to respond to immediate, short-range interactions, such as avoiding collisions or dynamically adjusting their paths based on proximate agents. This dual-system approach amalgamates the strengths of long-range planning with real-time responsiveness.

Due to the high computational cost of ray-casting, which resulted in a significant reduction in simulation speed, a number of optimisations and improvements to the model were made to enhance its efficiency for advanced scenarios. Caching was utilised to store the results of heuristic calculations and agent speed, thereby reducing the number of calculations and enhancing the overall efficiency of the system. This optimisation proved particularly beneficial in scenarios involving a large number of agents or collision shapes, where the computational cost of ray-casting was especially high.

The development of the simulation was accomplished through the utilisation of Godot 4.3, a game engine that has been demonstrated to be well-suited for handling the computational and graphical demands of such projects. Godot provides a native implementation of collisions, which was adapted and integrated into the system. Furthermore, custom raycast functionality was implemented to align with the unique requirements of the simulation. These customisations were essential for achieving the desired level of precision and realism in agent behaviour.

The integration of pathfinding and raycasts for local interactions has significantly enhanced the efficacy and realism of the simulation. This methodology ensures that agents can navigate complex environments with ease, while maintaining dynamic, responsive interactions with their surroundings and other agents.

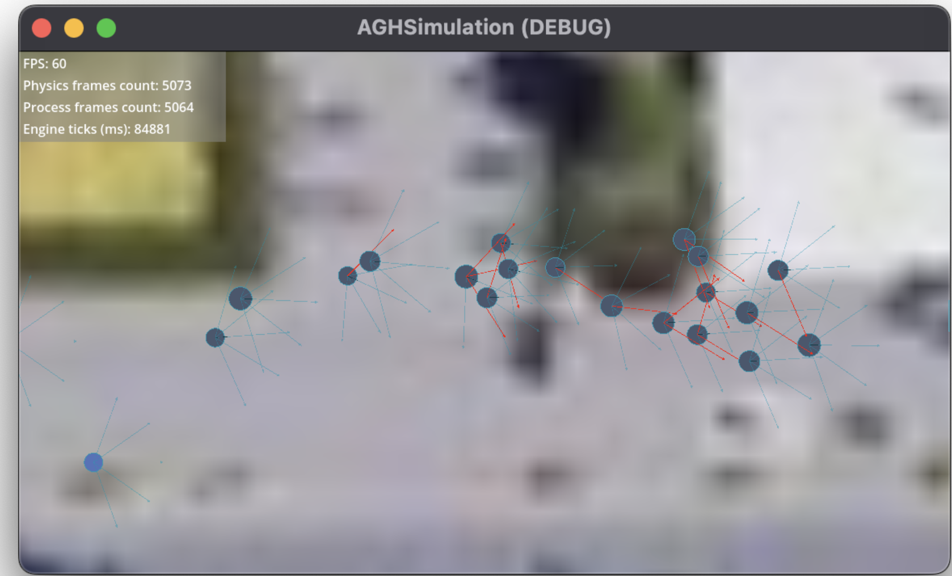


Figure 6: Raycasts may be considered as a means of representing the field of view of an agent

4 Results

A series of experiments were conducted with the objective of gathering all agents in front of the AGH Main Library building. The number of agents in the group was selected empirically during the course of the work. An excess of agents resulted in congestion in the bottlenecks, which include the stairs and the wheelchair ramp, as illustrated in fig. 7.

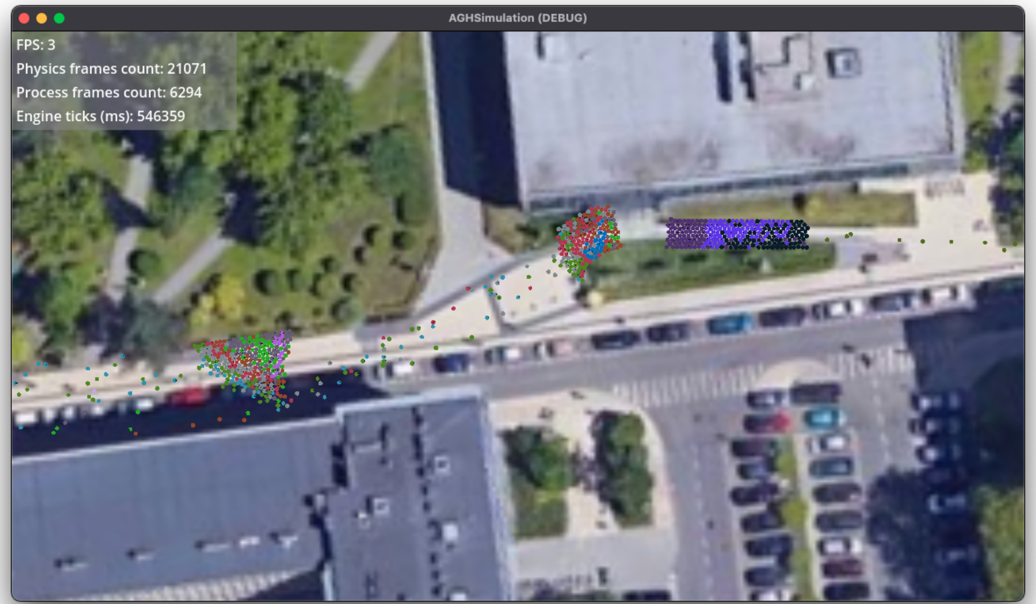
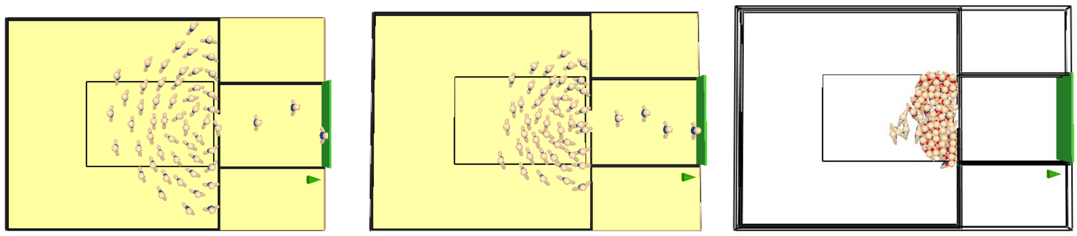


Figure 7: An excess of individuals has the potential to result in congestion

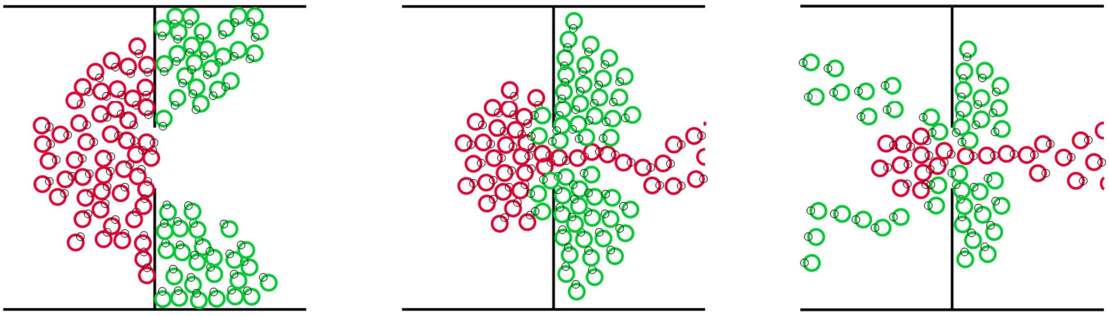
When agents are moving freely, the interpersonal distance is sufficiently large to have no effect on their routing, as illustrated in fig. 8a.However, when the density is higher, as depicted in fig. 8b, the interpersonal distance decreases and the agents begin to avoid each other. Further reduction in interpersonal distance results in agents blocking and pushing each other, thereby preventing movement (see fig. 8c). It is also assumed that agents falling down and impeding the passageway [5] are present.

It is also pertinent to mention the scenario in which agents attempt to traverse a bottleneck in opposite directions, as depicted in fig. 9. In such circumstances, the agents begin to impede each other's progress until one of them is impeded by the others. This phenomenon can occur during entry or exit from a building, as agents endeavour to navigate through narrow doorways. Although the present simulation does not include such door-passing, this behaviour has been observed between building C-4 and the AGH Main Library (agents marked in green and pink in fig. 3).



(a) (b) (c)

Figure 8: The present study explores the impact of interpersonal distance on bottleneck movement [5]



t = 0 s   t = 5 s   t = 10 s

Figure 9: Agents attempting to move in opposite directions through a bottleneck of the metro train boarding process [1]

5 Discussion

The primary enquiry concerned the feasibility of simulating the movement of students between the halls of the C-2 building. However, it was swiftly determined that this endeavour would necessitate a substantial investment of time and resources, which was deemed to be beyond the available timeframe. Consequently, the focus was shifted to the simulation of the academic community's movement within the university campus.

Following the modification of the project assumptions, the feasibility of executing a flow simulation in accordance with the stipulated timetable was called into question. In order to address this query, it is imperative to obtain information pertaining to the timetable, including the number of students residing in the building's halls. The USOS system is capable of providing such data; however, the process of acquiring it would be excessively time-consuming. This is attributable to the random nature of the URLs associated with the timetables of individual subjects.

The analysis of crowd flow on campus gives rise to a significant hardware concern: namely, the capacity of our computing devices to simulate such a substantial number of agents.During the course of the experiments, it was observed that the simulation of over 1,000 agents concurrently resulted in a substantial increase in system load and a decline in performance (to less than 10 frames per second). Furthermore, it was observed that performance degradation was particularly pronounced in scenarios characterised by high agent density, as evidenced in fig. 7. This figure depicts the performance of 630 agents navigating a map, where a clear decline in performance is evident.

6 Conclusion

Hardware limitations rendered it unfeasible to simulate pedestrian flows that were comparable to real-life scenarios. As a consequence, it can be concluded that the simulation results do not accurately reflect the university's reality. Notwithstanding, the selected model facilitates the examination of pedestrian behaviour in various locations across the AGH campus.

The solution prepared enables the observation of pedestrian behaviour during an evacuation, thus facilitating the determination of optimal evacuation routes and the avoidance of congestion and pushing. Another potential application of the simulation is the adjustment of timetables to enhance the movement of students between university buildings.

References

[1] Francesco Farina, Daniele Fontanelli, Andrea Garulli, Antonio Giannitrapani, and Domenico Prattichizzo. 2017. Walking ahead: The headed social force model. *PloS one* 12, 1 (2017), e0169734.

[2] Dirk Helbing and Peter Molnar. 1995. Social force model for pedestrian dynamics. *Physical review E* 51, 5 (1995), 4282.

[3] Julian Laufer and Principal Transport Planner. 2008. Passenger and pedestrian modelling at transport facilities. In *2008 annual AIPTM conference. Perth, australia*, 2008.

[4] Mehdi Moussaïd, Dirk Helbing, and Guy Theraulaz. 2011. How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences* 108, 17 (2011), 6884–6888.

[5] Peng Wang, Xiaoda Wang, Peter Luh, Christian Wilkie, Timo Korhonen, and Neal Olderman. 2022. Simulation of crowd egress with environmental stressors. *arXiv preprint arXiv:2206.01393* (2022).

[6] Jarosław Wąs. 2011. Modelowanie dynamiki tłumu. *Pomiary Automatyka Robotyka* 15, 12 (2011), 217–219.