

Accelerated Airborne Virus Spread Simulation: Coupling Agent-based Modeling with GPU-accelerated Computational Fluid Dynamics

Christoph Schinko^{1,5}^a, Lin Shao^{1,5}^b, Johannes Mueller-Roemer²^c, Daniel Weber²^d,
Xingzi Zhang³^e, Eugene Lee³, Bastian Sander^{4,6}^f, Alexander Steinhardt¹, Volker Sett gast^{1,5}^g,
Kan Chen³^h, Marius Erdt³ and Eva Eggeling^{1,5}ⁱ

¹Fraunhofer Austria Center for Data Driven Design, Graz, Austria

²Fraunhofer Institute for Computer Graphics Research IGD & Technische Universität Darmstadt, Darmstadt, Germany

³Fraunhofer Singapore & Nanyang Technological University, Fraunhofer IDM@NTU, Singapore

⁴Fraunhofer Institute for Factory Operation and Automation IFF, Magdeburg, Germany

⁵Graz University of Technology, Graz, Austria

⁶University of Vienna, Vienna, Austria

Keywords: Agent-based Modeling, Clustering, Computational Fluid Dynamics, Airborne Disease Transmission Modeling.

Abstract: The Coronavirus Disease 2019 (COVID-19) has shown us the necessity to understand its transmission mechanisms in detail in order to establish practice in controlling such infectious diseases. An important instrument in doing so are mathematical models. However, they do not account for the spatiotemporal heterogeneity introduced by the movement and interaction of individuals with their surroundings. Computational fluid dynamics (CFD) simulations can be used to analyze transmission on micro- and mesostructure level, however become infeasible in larger scale scenarios. Agent-based modeling (ABM) on the other hand is missing means to simulate airborne virus transmission on a micro- and mesostructure level. Therefore, we present a system that combines CFD simulations with the dynamics given by trajectories from an ABM to form a basis for producing deeper insights. The proposed system is still work in progress; thus we focus on the system architecture and show preliminary results.

1 INTRODUCTION

The Coronavirus Disease 2019 (COVID-19) has shown us, that a detailed, qualitative exploration of epidemiological cases is necessary (Shao et al., 2021). This means we have to understand its transmission mechanisms in detail, especially to establish practice in controlling such infectious diseases. COVID-19's

Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) is primarily spread through airborne droplets and aerosols (Lu et al., 2020; Shen et al., 2020). This transmission mechanism is more likely to occur in indoor than outdoor settings, when individuals are breathing, speaking, coughing, or performing any other form of exhalation (Qian et al., 2021; Leclerc et al., 2020; Stadnytskyi et al., 2020). When these particles are inhaled by other individuals (or come into contact with mucosae), pathogen transmission can occur. To lower the infection risk for individuals, recommendations for interventions have a focus on reducing indoor transmission by use of face masks, social distancing, and other measures. These general intervention recommendations have been supported by a number of mathematical models for indoor transmission analysis. The role of small, aerosolized

^a <https://orcid.org/0000-0001-5185-4565>

^b <https://orcid.org/0000-0002-5118-9515>

^c <https://orcid.org/0000-0002-0712-0457>

^d <https://orcid.org/0000-0001-6184-0570>

^e <https://orcid.org/0000-0003-3199-4434>

^f <https://orcid.org/0000-0003-1948-6621>

^g <https://orcid.org/0000-0003-3842-5349>

^h <https://orcid.org/0000-0003-3628-6699>

ⁱ <https://orcid.org/0000-0001-6703-2865>

droplets in indoor transmission for different settings has been determined by a number of models (Bhagat et al., 2020; Chen et al., 2020; Lelieveld et al., 2020; Sun and Zhai, 2020). However, these models do not account for the spatiotemporal heterogeneity introduced by the movement and interaction of individuals with their surroundings.

On the other hand, related work in the fields of computational fluid dynamics (CFD) and agent-based modelling is either infeasible due to a huge amount of computation needed for larger scale scenarios, or is missing means to simulate virus transmission on a micro- and mesostructure level. For a more detailed investigation of transmission effects, a combination of CFD simulations and the dynamics given by trajectories from an ABM seems to be a promising way.

Thus, we describe a novel system that is a combination of an agent-based model (ABM) with CFD simulations. For a given scenario, trajectories and aerosol-generating actions (e.g., coughing) are produced by an ABM. However, the sheer number of agents and the length of the produced trajectories would lead to huge computational effort on the CFD side, when the simulation is carried out for the entire scenario. This problem is tackled by intelligent clustering and selection of relevant situations, greatly reducing the input complexity for simulation.

The system lies the foundations for assessing individual infection risks of the simulated scenario. This can be done by applying risk models to the simulation results. The resulting infection risks can be applied back to their respective clusters enabling risk assessment for different infection scenarios and subsequent derivation of recommendations for safety precautions.

A first contribution of this paper is the combination of an ABM with CFD simulation based on intelligent selection and clustering of situations. With this approach, the complexity of a CFD simulation of the entire scenario including movement of all agents is reduced to small, but independent situations of relatively short duration. An additional benefit is the possibility to apply simulation results to similar situations of other scenarios. A second contribution is the development of a GPU-optimized, higher-order lattice Boltzmann fluid simulation for fast evaluation of the different situations. Since the development of the proposed system is still work in progress, we focus on the system architecture and show preliminary results.

2 RELATED WORK

This section presents important work that relates to our system architecture.

2.1 Agent-based Modeling

Generally, given two points on a model, we can find the optimal path between them in three steps. The first step is to construct a navigation mesh and turn the model into a graph. In the graph, each polygon is represented by a node and adjacent polygons are connected by an edge. Then, graph search algorithms, such as A* (Stout, 1996) and HPA* (Botea et al., 2004), are applied to find a series of adjacent nodes (polygons) on the graph that the optimal path will go through with the first node and last node. After that, path finding algorithms are used to find the optimal path, which is normally the shortest path, on the derived polygons. One of the commonly-used methods in path finding is the funnel-based method, e.g., Modified Funnel Algorithm (Demyen and Buro, 2006).

The path finding algorithms can fulfill the first requirement, leading the agents to their goals without colliding with any obstacles. To avoid collision with other agents, collision avoidance algorithms are required to adjust the agents' velocity when there is danger of collision. Much research has been done in this area, e.g., Optimal Reciprocal Collision Avoidance (ORCA) (Van den Berg et al., 2008) and Pedestrian Velocity Obstacles (PedVO) (Curtis, 2013), and open-source libraries such as RVO2 are available.

To achieve collision-free agent paths, we used the Urho3D game engine¹ and incorporate widely-used crowd simulation algorithms, e.g., Recast & Detour² for navigation mesh generation and path planning, and modified Reciprocal Velocity Obstacle (RVO)³ with adaptive sampling for collision avoidance.

2.2 Trajectory Similarity Measures and Clustering

To compute the similarity between trajectories, usually multidimensional and spatiotemporal features, e.g., position, speed, direction, and activity, have to be considered. In (Magdy et al., 2015), similarity measures for trajectories are roughly classified into two classes: (1) spatial similarity — measurements based on similar geometric shapes; and (2) spatiotemporal — measurements that spatial and temporal dimensions into account. Well-known techniques that consider spatial similarity are, for example, the Euclidean Distance, Hausdorff Distance (Chen

¹Urho3D: <https://urho3d.io/> (Accessed: 2021-10-28)

²GitHub: <https://github.com/recastnavigation/recastnavigation> (Accessed: 2021-10-28)

³Digesting Duck: <http://digestingduck.blogspot.com/2010/10/rvo-sample-pattern.html> (Accessed: 2021-10-28)

et al., 2011), and Fréchet Distance (Eiter and Man- nila, 1994). Examples for spatio-temporal measure- ments include the Longest Common Sub-Sequence (LCSS) (Vlachos et al., 2004) and Dynamic Time Warping (DTW) (Chen et al., 2005).

For trajectory clustering efficient algorithms are needed, which are scalable and flexible. Therefore, well-known clustering algorithms are often extended for trajectory data. Especially for clustering mov- ing object data, further cluster properties such as spatial information (Palma et al., 2008), time de- pendency (Nanni and Pedreschi, 2006) or local seg- ments (Lee et al., 2007), may be considered. Due to the large volume of trajectory data, we have decided to use a very fast partition-based algorithm that uses a spatial similarity measure, called minimum average direct-flip (MDF) distance (Garyfallidis et al., 2012).

2.3 Computational Fluid Dynamics

CFD simulations have proven to be an important tool in the modeling of the spread of airborne pathogens such as the novel SARS-CoV-2 virus (Vuorinen et al., 2020). However, full-fledged CFD simulations are computationally expensive. To simulate a large number of situations within an acceptable time frame, we examined massively parallel and approximate CFD simulation approaches. While the popular finite vol- ume approach allows for unconditionally stable sim- ulation (Stam, 1999), it requires the compute- and memory-intensive solution of a large linear system, limiting throughput and/or resolution even when ad- vanced solvers are used (Weber et al., 2015). Further- more, this approach suffers from significant numeri- cal diffusion, limiting simulations to small Reynold- s numbers, while coughing and movement in a cluttered environment result in highly turbulent flow (i.e., very high Reynolds numbers).

Therefore, we focus on the lattice Boltzmann method (LBM) using a $D3Q27$ discretization. Recent improvements to the LBM allow for highly turbulent flow with adaptive time steps for fast and stable sim- ulation, while decoupling obstacle geometry and fluid grid resolution (Li et al., 2020). Due to its locality, the LBM is ideally suited to massively parallel sim- ulation on GPUs. Especially when coupled with semi- analytic inversion of the collision operator, very low compute costs are achievable (Chen et al., 2021).

We combine the state-of-the-art LBM simulation, with a particle-based aerosol transport model (Guha, 2008). The particles are modeled with one-way coupling with the background fluid and no particle- particle interaction, allowing for full use of the GPU's parallel computing power.

3 SCENARIO

While the proposed system is not tailored towards a specific scenario, we are currently implementing its components with a typical supermarket scenario as a first use case. It resembles a typical European supermarket with one large customer area (approx. 19×52 m) with multiple shelves and counters. The combined entrance and exit area is accessible through a single door as it is common in many European supermarkets. Hence, customers entering the building are near to others exiting. There are three counters where customers need to check out before leaving the supermarket. We use a simplified geometry for the shelves and counters (see Figure 1).



Figure 1: A typical supermarket serves as an example. The simplified geometry contains shelves (green) and check-out counters (blue). Entrance and exit are located at the top left.

For the ABM, information about the accessible ar- eas, the position of the entrance and exit as well as target spots at the shelves and the check-out counters is created manually. To generate a sufficient amount of data, 88 agents are generated in front of the build- ing with a random number of goals (items to shop). To simulate the shopping process, agents move at a typical pedestrian speed through the market, stopping at their goals (locations at shelves) from time to time. After all goals have been reached, each agent will go to the check-out counter and then leave the market. Also, there is a stationary agent at each of the three check-out counters to represent a person working at the supermarket.

4 SYSTEM ARCHITECTURE

The proposed system consists of three interacting components: (1) The scenario is populated with agents doing specific tasks while acting as similar as possible (feasible) to real-world persons. The agent simulation generates trajectories and aerosol events. (2) Situations with agents are identified and clusters are generated out of trajectory and aerosol event data. Cluster representatives are identified. (3) The CFD simulation is performed on selected representatives. Particle trajectories are computed (spherical volume

around mouth and nose of an agent).

A set of interfaces ensures the interoperability of the system components. We differentiate between generic (mainly scenario-specific) input as well as input and output of specific components. The geometry of the simulated scenario and the reference coordinate system are input to all components.

ABM-specific input parameters are the number of agents to simulate, the starting positions of agents and parameters defining the agent's goals: number and positions of goal points, goal properties such as max duration to stop for at each goal and a sequence, which indicates the order of goals.

Next to the output produced by the ABM, clustering and selection of situations is provided with a target budget of situations to produce for the CFD simulation. The generated output consists of identified situations including participating agents as well as their respective trajectories and aerosol events. It is defined by a bounding box and a time frame.

In addition to the underlying scene geometry and the agent geometry, the mouth location/area must be specified. Other parameters such as air density and viscosity, aerosol particle drag factors, cough air velocity profiles, and particle counts are chosen based on the literature. Finally, the radius of the sphere in which particles inhaled by "receiving" agents are collected is required. Initially, we use a constant radius, but this could be extended to be based on the inflow/breathing rate.

4.1 Agent-based Modeling

Normally in ABM, agent movement is steered by goals. Given a supermarket scenario context, we design a goal sequence for the agents based on the common behaviors of in-store customers, which includes ENTER, SHOP, QUEUE, PAY, and EXIT. The goal points on the supermarket model are defined that going to (or leaving) these goal points means starting (or finishing) this goal. To meet the requirements of both randomness and universality in customer behavior, the target points for the goal SHOP are randomly chosen and iterated multiple times, since customers are likely to buy more than one item in the supermarket. All scenario-specific information is predefined and provided as input for our modeling.

After generating the goal sequence, a series of target points can be automatically selected for each agent at run time. Then, a navigation mesh based on the supermarket model can be build and the path finding method from the Detour library can be used to find the shortest path. However, scenarios which contain narrow paths, e.g., a supermarket with narrow aisles,

often cause unrealistic behavior of agents such as getting stuck, as shown in Figure 2. The reason for this is that agents may have identical targets and go through a narrow aisle from both directions at the same time.

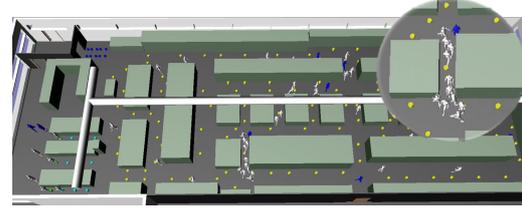


Figure 2: Agents tend to get stuck in narrow aisles if the effect of agent density is not considered in path finding.

To avoid such situations, we incorporate the Density-Cost algorithm in (Van Toll et al., 2012) into A* search (Stout, 1996) to find the optimal path between goal points. A density value is computed and maintained for each polygon in the navigation mesh that is proportional to the number of agents and inversely proportional to the area of a polygon.

Since agents move around, the density values may vary constantly and cause re-planning agent paths on the fly. Instead of considering all the density values along the way, incremental path finding is implemented by considering nearby density information and finding intermediate points for navigation to avoid unrealistic look-ahead in crowd situations. To prevent indecisive behaviors caused by the rapid change in density, when an agent reaches/leaves a polygon, we delay adding/removing the agent's density contribution to this polygon. Furthermore, a threshold for the maximum number of agents per goal is used to control the density of agents in each polygon and help agents avoid congested routes.

Besides movement simulation, we generate the aerosol-emitting events breathing (default), coughing, and talking for each agent. Here, coughing and talking events are randomly triggered for each agent. Empirically, once generated, coughing and talking events last for 5 to 10 seconds.

4.2 Clustering and Selection of Situations

The purpose of this component is to perform a clustering of the multidimensional data received from agent-based modeling and a selection of cluster representatives to be handed over to the simulation of airflow and aerosol distribution.

The clustering of the agent-based simulation data is performed in multiple stages.

Situation Extraction: In a first step, all situations are extracted from input trajectory data. In the context of this work, a situation is defined as the continuous period, where the distance between any two agents is less than 2 meters. The resulting situations are assigned to 5 classes: very short, short, medium, long and very long. Parameters for these classes are chosen to ideally obtain an even distribution of situations. A detailed overview of the situation classes for the supermarket scenario including number of trajectories and duration is shown in Table 1.

Spatial Clustering: In a second step, the trajectories of each situation class are further clustered based on the position and orientation to incorporate spatial mappings of the supermarket. To consider subsequent as well as oncoming trajectories, the QuickBundles clustering algorithm (Garyfallidis et al., 2012) along with the minimum average direct-flip (MDF) distance is used. QuickBundles is a clustering algorithm for sequences of points in 3D space that considers both directions, original and flipped (swapped start and end) versions of a path. For each situation, which consists of two agents, the orientation of one agent is rotated by 180 degrees to increase the weighting of situations where agents meeting one another halfway.

Representative Selection: To choose a set of varying situations for the simulation of airflow and aerosol distribution (cf. Section 4.3), different features like aerosol events, spatial positions, and situation classes need to be taken into consideration. Therefore, multiple situations in each cluster and for each situation class are computed as representatives. As criteria for choosing the representatives, trajectories including minimum, maximum, and the average number of occurrences of the aerosol event talking and coughing are considered. In the end, the set of representatives consists of situations with different trajectory lengths (situation classes), spatial positions (clusters), and aerosol events (criteria).

4.3 Simulation of Airflow and Aerosol Distribution

In order to assess the infection probabilities, a simulation of the airflow and the aerosol distribution is conducted using a GPU-accelerated LBM.

Integration: A part of the scenario is selected by specifying its bounding box and the relevant geometry is extracted within the solver and used as boundary condition as shown in Fig. 3. In addition, the agent geometry and movement data are imported to be able to reproduce agent interactions. A simulation run then computes the creation and the movement of aerosol

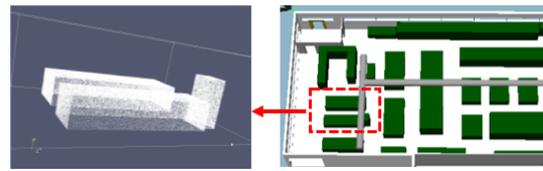


Figure 3: Specifying a bounding box, the corresponding geometry can be extracted and used as boundary conditions.

particles according to the input data (see Section below). In order to assess infection probabilities, the number and sizes of aerosol particles from the source agent are counted (and deactivated) when they are in proximity to another agent's nose and mouth. The accumulated numbers can be used to approximate the amount of inhaled aerosol and to eventually judge the probability of infection. This mechanism allows to select the infectious agents after the CFD simulation to assess diverse infection scenarios.

CFD Simulation: We implemented a GPU-optimized, higher-order LBM fluid simulation code based on the works of (Li et al., 2020; Chen et al., 2021). The LBM discretizes the microscopic (particle-level) Boltzmann equation using a particle distribution $f(\mathbf{x}, \mathbf{c}, t)$ with discretized locations \mathbf{x}_k (the lattice nodes) and a discrete set of particle velocities \mathbf{c}_i . The developed code uses a D3Q27 discretization, i.e., a three-dimensional lattice with 27 discrete fluid velocities $\mathbf{c}_0 = (0, 0, 0)^T$, $\mathbf{c}_{1,2} = (\pm 1, 0, 0)^T$, ..., $\mathbf{c}_{7-10} = (\pm 1, \pm 1, 0)^T$, ..., $\mathbf{c}_{20-27} = (\pm 1, \pm 1, \pm 1)^T$ for $\Delta x = \Delta t = 1$. These are advanced by alternating between a streaming (transport) kernel and a collision (relaxation) kernel, which both operate locally, allowing for full use of the GPUs massively parallel compute power. Macroscopic Navier-Stokes variables can be derived from the moments of the distribution.

The exterior boundary is modeled using extrapolating boundary conditions (Geier et al., 2015) for the open boundary and slip boundary conditions for the floor, implemented using the specular reflection method (Li, 2011). Obstacles within the domain, including the agents themselves, are modeled using the immersed boundary method (Li et al., 2020). This requires resampling geometry as a point cloud that matches or exceeds the resolution of the grid using Poisson disk sampling (Yuksel, 2015), as can be seen in Fig. 3 on the left.

We combine the turbulent LBM fluid simulation with aerosol particle simulation using Lagrangian particles following the equations of motion described in (Guha, 2008). Coupling to the fluid is unidirectional, and aerosol particles do not interact. These approximations are used to reduce computational load. We model cough velocity profiles based on measure-

ments from (Gupta et al., 2009), while particle size distributions follow (Wang et al., 2020).

Since the computation of the drag force in Guha’s transport model includes division by a factor τ that can become arbitrarily small, the model can cause aerosol transport to become unstable when $\tau < 1$ in lattice units, since the velocity difference $\mathbf{u}_{\text{particle}} - \mathbf{u}_{\text{fluid}}$ is over-corrected. However, this only applies to particles with a very small diameter, which we then model as massless particles, i.e., $\mathbf{u}_{\text{particle}} \leftarrow \mathbf{u}_{\text{fluid}}$. Alternatively, time step subdivision could be used for these particles. However, others have shown that the behavior is sufficiently similar when using the massless approximation (Vuorinen et al., 2020).

5 PRELIMINARY RESULTS

This section presents the results of preliminary experiments with the proposed system.

For the supermarket scenario, the ABM component described in Section 4.1 is used to simulate the spawned agents. The agents spawn random in position and interval in front of the entrance of the supermarket. Trajectories and aerosol-generating events are recorded between entering and exiting the supermarket at an interval of 0.02 seconds. We define the agent speed following a normal distribution with mean 1.34 and standard deviation 0.16 to cover a typical range of walking speed of different ages. The customer agents follow the same goal sequence, which is ENTER, SHOP, QUEUE (optional), PAY and EXIT, and the duration of each agent varies.

The situation extraction step, described in Section 4.2, returns a total of 4494 situations, where two agents are approaching each other with a distance of less than 2 meters. All extracted situations are then classified based on the trajectory lengths, see table 1. The class *very short* contains situations of short encounters of agents within the range of 0.02–1.30 seconds, whereas *very long* situations imply all encounters longer than 5.08 seconds (c.f. Table 1).

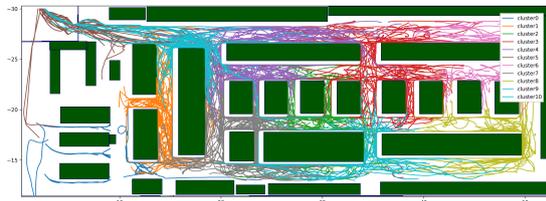


Figure 4: QuickBundles clustering results on trajectories of the situation class *very long*.

The QB clustering is a partitioning-based algorithm that relies on a distance threshold ζ , which

controls the heterogeneity of clusters inverse proportional. In order to achieve a good spatial separation of situations in the supermarket, a threshold $\zeta = 9m$ is used, which subdivides the area of the supermarket into 8 to 11 sub-areas (clusters) for the five different situation classes. Figure 4 shows the spatial clustering of situations with very long trajectories and Figure 5 the associated incidence of aerosol event 1 (talking) and aerosol event 2 (coughing). One can see that the overall area of the supermarket including meaningful areas like entrance, checkout area and individual supermarket sections, can be well covered by the 10 defined clusters and that aerosol events are normally distributed in the supermarket.

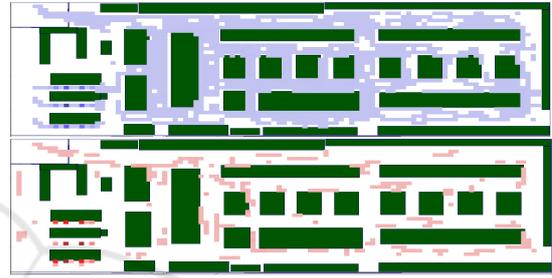


Figure 5: Incidence plots of the aerosol event 1 talking (top) and 2 coughing (bottom).

Table 1: Details of situation extraction and spatial clustering. This table shows for each situation class the number of trajectories, the duration of situations in seconds, the number of computed clusters and the computation time.

| Situation | Traj. | Duration (s) | Clusters | Time (s) |
|------------|-------|--------------|----------|----------|
| very short | 1802 | 0.02–1.30 | 9 | 11.66 |
| short | 1804 | 1.31–2.10 | 8 | 13.59 |
| medium | 1786 | 2.11–3.03 | 9 | 13.66 |
| long | 1798 | 3.04–5.07 | 11 | 15.29 |
| very long | 1798 | > 5.08 | 10 | 19.57 |

Finally, the situations identified by the QB clustering are imported into the airflow and aerosol simulation described in Section 4.3, by extracting the bounding box and trajectory of the encounter. Figure 6 shows excerpts from a simulation of a cough event near the cash registers. The simulation was computed with ParaView-compatible file output for the figures only, as file writing takes up significantly more time than the simulation itself (214s vs. 629s on a laptop equipped with an NVIDIA GeForce RTX 3070M GPU). For the final simulations, no particle and velocity output is written to disk, only particle histograms over time near the at-risk agent’s mouth. The simulation shown in Fig. 6 was computed at a resolution of $133 \times 49 \times 99$, with 207736 surface samples for the immersed environment geometry, and 7213 surface samples on the agent geometry. The figure shows the

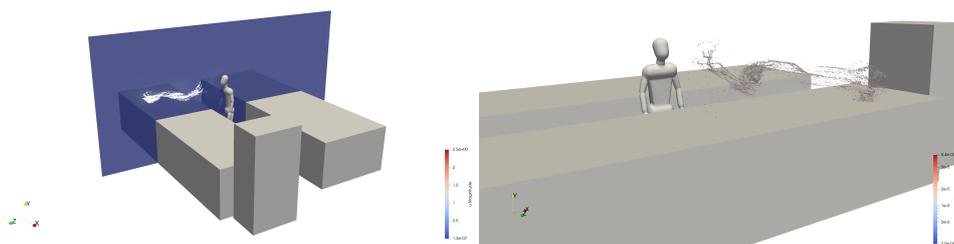


Figure 6: Aerosol transport simulation results for an agent standing at the cash registers. The left image visualizes the velocity and the particle distribution 2.5s after a cough event. The right image shows a different viewpoint after 5s with particles colored by diameter, showing that large particles drop down fairly close to the agent.

original geometry clipped to the situation bounding box, not the surface samples (compare with Fig. 3).

6 CONCLUSION AND FUTURE WORK

Previous work on SARS-CoV-2 transmission focused either on CFD, or on ABM alone (Shao et al., 2021; Farthing and Lanzas, 2021). Thus, we presented a novel system that combines ABM with CFD simulations by intelligent clustering and selection of relevant situations for a given scenario. A typical supermarket scenario is used as a first test case. The combination of the sheer number of agents, the length of the produced trajectories and the size of the scenario would lead to huge computational effort on the CFD side, if simulated as a whole. By selecting relevant situations, the computational complexity for the CFD simulation can be greatly reduced, but at the cost of limited accuracy. Situations are simulated independently from each other and only for selected cluster representatives. A thorough investigation of the impact on accuracy of our approach compared to a CFD simulation of the entire scenario is planned for future work (with ventilation added). Furthermore, we plan to investigate situations with more than two participating agents, which seems to be a non-trivial extension. Also we plan to add more variety into the agent’s movement patterns (e.g., social forces), more realistic behavior (such as sneezing or adding groups of agents), and safety equipment (e.g., different face masks).

While we showed preliminary results for the supermarket scenario, we are still in the process of stabilizing our CFD code to produce airflow simulation results. Also, this paper has presented necessary groundwork to be applied to real scenarios, but risk assessment has yet to be carried out. Both the physics of relevant processes (e.g., advection, evaporation, sedimentation) and the workflow are appropriate to substantially assess infection risks. This can be

done with realistic spatial distributions of infectious particles and inhalation rates (by changing respective physical activity states).

In turn, spatially refined and agent-dependent infection risks are key to infer individual safety precautions. Within a simulation scenario the relative effect of a single or combined safety precautions are to be evaluated. The most effective and least invasive combination of precautions, which conforms with processes specific to the scenario, must be favored. The precautions can be applied locally, i.e., within spatial sub-domains with high infection risk, or globally for the entire scenario.

ACKNOWLEDGEMENTS

This research has received funding from the Fraunhofer Anti-Corona Project “AVATOR – Anti-Virus-Aerosol: Testing, Operation, Reduction”. We would like to thank Stefanie Samtleben for valuable feedback and discussions.

REFERENCES

- Bhagat, R. K., Davies Wykes, M. S., Dalziel, S. B., and Linden, P. F. (2020). Effects of ventilation on the indoor spread of COVID-19. *Journal of Fluid Mechanics*.
- Botea, A., Müller, M., and Schaeffer, J. (2004). Near optimal hierarchical path-finding. *J. Game Dev.*, 1(1).
- Chen, J., Wang, R., Liu, L., and Song, J. (2011). Clustering of trajectories based on Hausdorff distance. In *2011 International Conference on Electronics, Communications and Control (ICECC)*, pages 1940–1944.
- Chen, L., Özsu, M. T., and Oria, V. (2005). Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, page 491–502.
- Chen, W., Zhang, N., Wei, J., Yen, H.-L., and Li, Y. (2020). Short-range airborne route dominates exposure of respiratory infection during close contact. *Building and Environment*, 176:106859.

- Chen, Y., Li, W., Fan, R., and Liu, X. (2021). GPU optimization for high-quality kinetic fluid simulation. *IEEE Trans. Vis. Comput. Graph.*
- Curtis, S. (2013). *Pedestrian velocity obstacles: Pedestrian simulation through reasoning in velocity space*. PhD thesis, University of North Carolina at Chapel Hill.
- Demyen, D. and Buro, M. (2006). Efficient triangulation-based pathfinding. In *Aaai*, volume 6, pages 942–947.
- Eiter, T. and Mannila, H. (1994). Computing discrete Fréchet distance. Technical report, Technische Universität Wien. CD-TR 94/64.
- Farthing, T. S. and Lanzas, C. (2021). Assessing the efficacy of interventions to control indoor SARS-CoV-2 transmission: an agent-based modeling approach. *medRxiv*.
- Garyfallidis, E., Brett, M., Correia, M., Williams, G., and Nimmo-Smith, I. (2012). Quickbundles, a method for tractography simplification. *Front. Neurosci.*, 6:175.
- Geier, M., Schönherr, M., Pasquali, A., and Krafczyk, M. (2015). The cumulant lattice Boltzmann equation in three dimensions: Theory and validation. *Computers & Mathematics with Applications*, 70(4):507–547.
- Guha, A. (2008). Transport and deposition of particles in turbulent and laminar flow. *Annual Review of Fluid Mechanics*, 40(1):311–341.
- Gupta, J. K., Lin, C.-H., and Chen, Q. (2009). Flow dynamics and characterization of a cough. *Indoor Air*, 19(6):517–525.
- Leclerc, Q. J., Fuller, N. M., Knight, L. E., Funk, S., and G. M. K. (2020). What settings have been linked to SARS-CoV-2 transmission clusters? *Wellcome Open Research*, 5(83).
- Lee, J.-G., Han, J., and Whang, K.-Y. (2007). Trajectory clustering: A partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, page 593–604.
- Lelieveld, J., Helleis, F., Borrmann, S., Cheng, Y., Drewnick, F., Haug, G., Klimach, T., Sciare, J., Su, H., and Pöschl, U. (2020). Model calculations of aerosol transmission and infection risk of COVID-19 in indoor environments. *Int. J. Environ. Res. Public Health*, 17(21).
- Li, W., Chen, Y., Desbrun, M., Zheng, C., and Liu, X. (2020). Fast and scalable turbulent flow simulation with two-way coupling. *ACM Trans. Graph.*, 39(4).
- Li, Y. (2011). *An improved volumetric LBM boundary approach and its extension for sliding mesh simulation*. phdthesis, Iowa State University.
- Lu, J., Gu, J., Li, K.-B., Xu, C., Su, W., Lai, Z., Zhou, D., Yu, C., Xu, B., and Yang, Z. (2020). COVID-19 outbreak associated with air conditioning in restaurant, Guangzhou, China, 2020. *Emerg. Infect. Dis.*, 26.
- Magdy, N., Sakr, M. A., Mostafa, T., and El-Bahnasy, K. (2015). Review on trajectory similarity measures. In *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems*.
- Nanni, M. and Pedreschi, D. (2006). Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, 27:267–289.
- Palma, A. T., Bogorny, V., Kuijpers, B., and Alvares, L. O. (2008). A clustering-based approach for discovering interesting places in trajectories. In *Proceedings of the 2008 ACM Symposium on Applied Computing*.
- Qian, H., Miao, T., Liu, L., Zheng, X., Luo, D., and Li, Y. (2021). Indoor transmission of SARS-CoV-2. *Indoor Air*, 31(3):639–645.
- Shao, S., Zhou, D., He, R., Li, J., Zou, S., Mallery, K., Kumar, S., Yang, S., and Hong, J. (2021). Risk assessment of airborne transmission of COVID-19 by asymptomatic individuals under different practical settings. *Journal of Aerosol Science*, 151:105661.
- Shen, Y., Li, C., Dong, H., Wang, Z., Martinez, L., Sun, Z., Handel, A., Chen, Z., Chen, E., Ebell, M. H., Wang, F., Yi, B., Wang, H., Wang, X., Wang, A., Chen, B., Qi, Y., Liang, L., Li, Y., Ling, F., Chen, J., and Xu, G. (2020). Community Outbreak Investigation of SARS-CoV-2 Transmission Among Bus Riders in Eastern China. *JAMA Internal Medicine*, 180(12):1665–1671.
- Stadnytskyi, V., Bax, C. E., Bax, A., and Anfinrud, P. (2020). The airborne lifetime of small speech droplets and their potential importance in SARS-CoV-2 transmission. *Proceedings of the National Academy of Sciences*, 117(22):11875–11877.
- Stam, J. (1999). Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques, SIGGRAPH '99*. ACM Press.
- Stout, B. (1996). Smart moves: Intelligent pathfinding. *Game developer magazine*, 10:28–35.
- Sun, C. and Zhai, Z. (2020). The efficacy of social distance and ventilation effectiveness in preventing COVID-19 transmission. *Sustain. Cities Soc.*, 62:102390.
- Van den Berg, J., Lin, M., and Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935. IEEE.
- Van Toll, W. G., Cook IV, A. F., and Geraerts, R. (2012). Real-time density-based crowd simulation. *Computer Animation and Virtual Worlds*, 23(1):59–69.
- Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., and Keogh, E. J. (2004). Indexing multidimensional time-series. *The VLDB Journal*, 15:1–20.
- Vuorinen, V., Aarnio, M., Alava, M., Alopaeus, V., et al. (2020). Modelling aerosol transport and virus exposure with numerical simulations in relation to SARS-CoV-2 transmission by inhalation indoors. *Safety Science*, 130.
- Wang, Y., Xu, G., and Huang, Y.-W. (2020). Modeling the load of SARS-CoV-2 virus in human expelled particles during coughing and speaking. *PLoS One*, 15(10).
- Weber, D., Mueller-Roemer, J., Stork, A., and Fellner, D. (2015). A cut-cell geometric multigrid Poisson solver for fluid simulation. *Computer Graphics Forum*, 34(2):481–491.
- Yuksel, C. (2015). Sample elimination for generating Poisson disk sample sets. *Computer Graphics Forum*, 34(2):25–32.