Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P3300-cd

Virtual Planning and Support for Large-Scale Events

Antonio Kruse, Ruth Meyer, Fabian Müller, Corinna Köpke, Jörg Finger

Fraunhofer Institute for High-Speed Dynamics, Ernst-Mach-Institut, EMI, Freiburg, Germany.
E-mail: ruth.meyer@emi.fraunhofer.de, corinna.koepke@emi.fraunhofer.de, joerg.finger@emi.fraunhofer.de

André Viergutz, Thomas Golda

Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, IOSB, Karlsruhe, Germany.
E-mail: andre.viergutz@iosb.fraunhofer.de, thomas.golda@iosb.fraunhofer.de

In this paper we present a modular simulation toolkit designed to assist in the planning and management of large-scale outdoor events like festivals or Christmas markets. The toolkit integrates four key modules: 3D scene reconstruction, visibility computation, agent-based modeling (ABM) of pedestrian flows, and smart video analysis. Data for the framework are obtained from various sources, including drone LiDAR scans, drone videos, ground photos, and fixed video cameras. Our methodology follows a structured process chain: starting with 3D reconstruction to create a detailed virtual environment, followed by visibility computation to assess the perceptibility of information. The ABM module simulates the behavior of event participants, allowing for dynamic interaction of individual agents with and within the environment in different scenarios. Finally, simulation results are validated using insights from smart video analysis, which delivers person counts and density estimates. This ensures the accuracy and reliability of the outcomes. We demonstrate the use and flexibility of our modular simulation toolkit with its application to *Juicy Beats*, a popular electronic music festival held annually in Germany.

Keywords: Large-scale events, 3D scene reconstruction, visibility computation, agent-based modeling, simulation, video analysis.

1. Introduction

Large-scale outdoor events such as music festivals or Christmas markets attract large numbers of people, creating highly complex environments where safety and reliability must be carefully considered. These environments can be influenced by various factors, including human behavior, technical malfunctions, or natural disasters, all of which can potentially lead to dangerous situations. Therefore, it is essential to thoroughly assess the safety infrastructure, identify potential hazards, and develop effective countermeasures.

Simulations have proven to be powerful tools in this context, as they enable the modeling and evaluation of different scenarios to support informed decision making. However, modular simulation toolkits that can adapt to the needs of various event organizers are scarce. Developers usually provide tools for a specific task, e.g., for pedestrian simulations (Senanayake et al. (2024)) or viewshed analyses (McElhinney (2024)). These tools are not intended to be linked, and knowledge about interdependent effects cannot be gained.

In the following, we present a modular simulation toolkit, which integrates four key modules: 3D scene reconstruction, visibility computation, agent-based modeling (ABM) of pedestrian flows, and smart video analysis. We illustrate how this integrated, linked approach enables detailed and reliable insights into safety-relevant aspects of large-scale events.

2. Methodology

2.1. *3D Scene Reconstruction*

A detailed 3D model of the event area forms the basis for several of the methods described below. While 3D models are available for many cities or well-known areas, it is necessary to reconstruct a detailed 3D model for unknown sites and especially for temporary events, additional objects such as stages, drinks stalls or barriers are usually missing in existing models.

Since modeling by hand would be very tedious

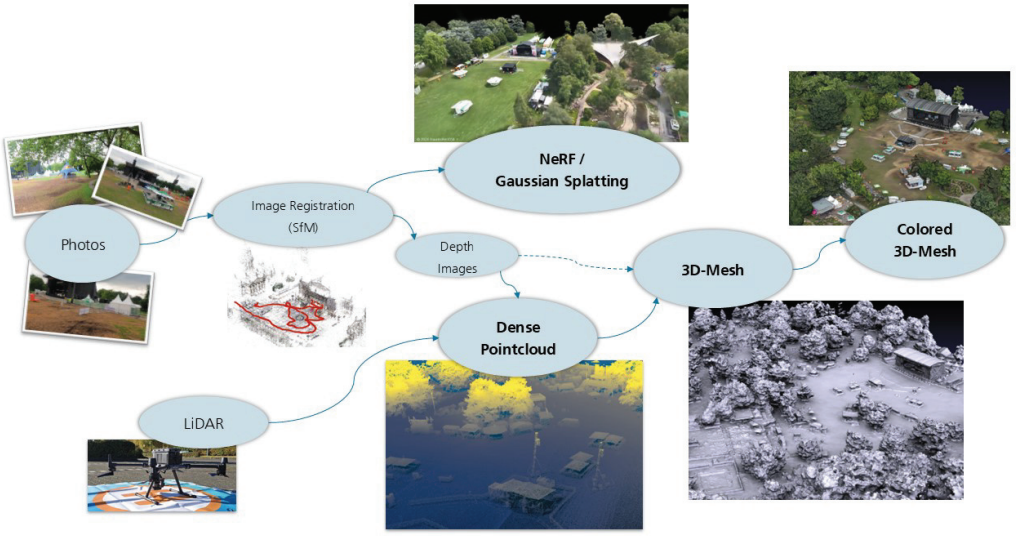

Fig. 1. Overview of our 3D reconstruction pipeline.

and time-consuming, we apply modern recording and reconstruction technologies (see Fig. 1). Aerial images as well as images from a first-person view cover a wide range of the site modalities.

Recording a continuous video of the area enables us to determine the camera poses of the recorded single images using structure-from-motion (SfM). Multiple View Stereo techniques (MVS) are used to extract depth maps, which are subsequently fused to a dense (colored) point cloud. We use COLMAP, an open-source 3D reconstruction toolbox by Schönberger and Frahm (2016), to accomplish these steps.

Another way to obtain a point cloud is to capture real 3D points directly via LiDAR technology. While the laser scanner itself only captures intensity values, color information can be obtained from additional RGB images registered to the LiDAR sensor. We will compare both point clouds in Section 3.

A GPS module on the drone or camera rack allows us to embed the 3D model into a global reference coordinate system, making it easy to orient and scale the model appropriately as well as combine several inputs into the same virtual environment. If there is no GPS data available, geo-referencing can also be conducted manually in a post-processing step.

In order to create a coherent triangular mesh from the point cloud, the individual points are connected to form triangles. Poisson surface reconstruction (Kazhdan et al. (2006) and following work) is well suited for this purpose. In very fine-grained meshes, the RGB information can be saved directly with the vertices of the mesh but to decouple the perceived color fidelity from mesh resolution, a texture atlas is generated and filled with colors looked up and blended from the captured views (e.g., Waechter et al. (2014)).

In addition to triangle meshes, we have also used the results of the SfM to calculate a 3D Gaussian Splatting model (Kerbl et al. (2023)) in order to virtually walk around the area, showing even small optical details that are often swallowed up with photogrammetric reconstruction.

2.2. *Visibility Computation*

The calculation of the total area or volume visible from a vantage point and the evaluation of the visibility of certain facilities, such as emergency exit signs or video walls on stages, rely on ray-

tracing algorithms that implement the concepts of visibility graph (Turner et al. (2001)) and isovist field (Batty (2001)). The fundamental requirement for it to work is a detailed 3D model of the scenery in the form of a triangle mesh stored in STL (Standard Triangle Language) file format.

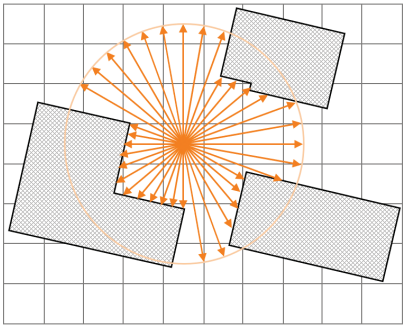

Fig. 2. Construction of an isovist: rays are cast from a cell center with a constant angle increment.

To calculate overall visibility, the algorithm casts rays from every point within the area of interest to the surroundings. Since there is an infinite number of points within a closed area, a discretization is necessary. The algorithm spreads a rectangular grid of equally sized square cells over the entire area. Cells outside the area boundaries or within an obstacle are removed. Cell centers serve as origins for sets of horizontal rays, which are sent out circularly with a certain angle increment. If a ray does not hit any obstacle, it will end after a pre-defined length, which is the maximum visibility range (Fig. 2). From the sum of the squares of the ray lengths, the total visible area from a vantage point can be approximated. The area calculated in this way converges to the so-called isovist (Benedikt (1979)) of the point if the angle increment approaches zero.

In an analogous way, it is also possible to approximate the visible volume. To this end, the rays spread dome-like from each cell center with given azimuth and elevation angle increments. The summation is then performed using volumes (proportional to the radius cubed) instead of areas.

In order to evaluate the visibility of facilities, the algorithm sends rays from the cells associated with those facilities to all other cells. For rays that are not blocked by obstacles, the visibility depends on the distance between the target cell and the facility cell as well as the angle between the ray and the facility's normal (if the facility is directional).

In general, the ground of a (real) scenery is not level. To account for changing elevations, the algorithm adjusts the vertical position of the cells to the terrain using a digital elevation model (DEM). The DEM must be given in xyz format, where x and y represent geographic coordinates and z are relative or absolute elevations. The grid cells are then positioned with a constant vertical distance to the ground, adopting the terrain shape. The distance corresponds to the view height of the virtual observer. A DEM can be obtained either from 3D surface reconstruction by automated elimination of all vegetation and buildings or from the stock of the communal land registry.

The described algorithms are implemented in C++ using OpenMP 3.0 shared memory parallelization for execution on CPUs, and Nvidia CUDA 12.1 programming for massively parallel execution on GPUs. The latter enables near real-time computation of the overall visibility. A framework of Python wrapper scripts facilitates the preparation and organization of input files as well as the creation of outputs in the desired formats (e.g., PNG, GeoJSON, GeoTIFF). It can easily be extended to process other formats such as Open Geospatial Consortium (OGC) standards.

2.3. *Agent-Based Simulation*

The agent-based simulation module models pedestrian flows within a spatial environment. Each pedestrian is represented by an agent with specific characteristics and behavior. The characteristics like age, gender, preferred speed, or fitness level can either be taken from empirical data or assigned randomly using predefined distributions. Each agent's behavior consists of physics-based movement in continuous 2.5D space, a needs-driven action selection mechanism, and information processing. We model agent needs such as hunger, thirst, entertainment, or rest as reservoirs, which deplete over time and can be refilled

by certain actions or objects within the environment. Agents determine what to do next by finding the action that best satisfies their current most pressing need (Meyer et al. (2024)). In a festival context, e.g., a thirsty agent will choose the nearest drink stall to go to, whereas an agent who primarily wants to be entertained will go join the crowd in front of a stage.

Movement is determined on two levels. At the macro-level, agents use the A* algorithm (Hart et al. (1968)) to find the shortest route from their current position to a chosen target. At the micro-level, motion and collision avoidance along this route are modeled using a force-based approach, combining the well-known social force model (Helbing and Molnar (1995)) with fluid dynamics and potential fields. The force $\vec{F}_i$ acting on agent i within a time step dt is composed of:

(1) A locomotive force towards the target, taken from the social force model;
(2) Repulsive forces from other agents, varying with crowd density;
(3) Repulsive forces from nearby obstacles.

For low density cases, we use the social force model, whereas for high densities – where a crowd can plausibly be compared to behaving like a fluid – we apply the Smoothed Particle Hydrodynamics approach (van Toll et al. (2021)). For densities in between, we interpolate between both regimes. To enhance computational efficiency, collision avoidance with static obstacles in the environment is achieved by deriving the repelling forces from a pre-calculated potential field (Lu et al. (2016)), and storing them in a geo-referenced grid at model initialization.

To be able to model situations at large-scale events requiring evacuation, agents are equipped with information processing. When they perceive a warning message issued, e.g.,from a stage, video display or loudspeaker because their current position is within the perceptibility range as computed by the visibility module (see Section 2.2), they will try to understand it. Comprehension is influenced by the quality of the warning message, i.e., the higher the informativity score of a message, the higher the probability that an agent understands the message (Schmidt-Colberg et al. (2024)). Lack of understanding will drive an agent to seek additional information by either remaining close to the warning device in order to perceive the message again or finding other agents in the vicinity to talk to.

The environment is represented as a set of areas containing objects of interest for the agents such as stages, booths, entrances and exits, or other features important for navigation such as streets, stairs and obstacles. These are imported as multi-polygons in the GeoJSON format. In contrast to the visibility module, the ABM cannot work directly with a simple 3D-mesh but needs additional semantic information about the existing features in order to allow agents' useful interaction with the modeled scenario.

2.4. *Video Analysis*

For the purpose of crowd monitoring, video data is captured and processed using optical pan-tilt-zoom (PTZ) cameras. The data provides essential information about the number of individuals and the flow of the crowd at specific moments in time. It can, for example, be used to initialize the ABM by specifying the locations and directions of the agents in the model that represent people in the monitored crowd.

Furthermore, extracted information can be considered suitable for the derivation of other significant metrics that contribute to safety concepts and crowd management strategies.

To map pixel data to real-world coordinates, the image domain is projected onto world coordinates through a series of transformations. These transformations include intrinsic camera calibration, which account for internal camera parameters, and extrinsic parameters such as rotation and translation, derived from the camera's orientation and position. For a flat ground plane, the projection is simplified by assuming the height of the plane to be zero. The result is a projection matrix specifically tailored for the ground plane. By inverting this matrix, detected points are georeferenced, which enables the creation of a metered grid. This grid is then used to analyze each cell and determine the number of individuals present.

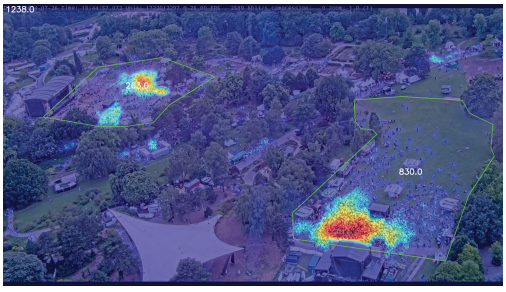

Fig. 3. Overlay of a density-based crowd counting heatmap over the corresponding view of the festival area. The two target areas marked for the video analysis are the areas of the two main stages visible from the camera position.

A convolutional neural network (CNN) is employed to generate crowd density heatmaps. These heatmaps provide an approximate pedestrian distribution for various regions in the image, as depicted in Fig. 3. The density heatmaps are subsequently used to estimate the number of people in each grid cell.

3. Application

To demonstrate the use and flexibility of our modular toolkit introduced in Section 2, we apply it to the *Juicy Beats* festival. This is an annual music festival held in a large public park in Dortmund, Germany. It features a diverse lineup of electronic, hip-hop and indie music artists, attracting a large audience of about 30,000 participants who enjoy live performances in an outdoor setting. The festival organizers cooperated with us, shared their security concept, and allowed us access to the festival grounds for data capture and observation.

3.1. *Data Acquisition*

Prior to the opening of the festival site for visitors, we walked the grounds for nearly 2.5 hours recording the state of the almost fully prepared site with a GoPro HERO 8 and Insta360 X4 camera mounted on a telescopic boom to cover several viewing directions at once. In addition, we conducted several automated flights with our DJI Matrice M350 RTM drone + Zenmuse L2 LiDAR sensor that covered a total area of approximately 0.14 km². This resulted in a large pool of data covering a variety of camera perspectives (different heights, pitch angles, 360° views) as well as types of recorded data (RGB, thermal, 3D points).

3.2. *Photogrammetry vs. LiDAR*

Although the photogrammetrically reconstructed mesh from ground photos contains slightly more detail in certain areas and lets us look underneath roofs (see Fig. 4, left), the mesh generated from the LiDAR point cloud (Fig. 4, right) has more appealing properties for our use case. The smoother and more complete point coverage makes it easier for the surface reconstruction algorithm to identify neighboring points. The higher vantage point allows for a better capture of objects, including, e.g., roofs and treetops. Similar results are achievable with airborne photogrammetry, except that the laser beam is able to penetrate vegetation so that we could even capture ground points from the air, and the recording and post-processing times of the LiDAR approach are noticeably shorter, saving hours of point-cloud generation.

3.3. *Visibility Analysis*

Starting from the LiDAR-based 3D model of the relevant festival area, we conducted a visibility analysis. First, the overall visibility (Fig. 5a) gives a comprehensive and objective measure that helps to identify regions of low visibility. Those regions harbor the risk of inadequate dissemination of information in the event of a dangerous situation. This can, e.g., lead to disorganized escape movements of visitors during a panic. At the *Juicy Beats* festival, there are two main stages, each of which opens onto a large lawn. These areas are represented by bright yellow and orange colors in Fig. 5a indicating high visibility. However, the region between the two stages is covered with trees, reducing the visible area to a minimum.

Second, we are interested in the spread of information from the stages (Fig. 5b). On the one hand, this is important in order to estimate where visitors will position themselves so that they can follow the show in the best possible way. On the other hand, in the event of an emergency, the stages can also be used to disseminate instructions on appropriate visitor behavior or to control an evacuation.

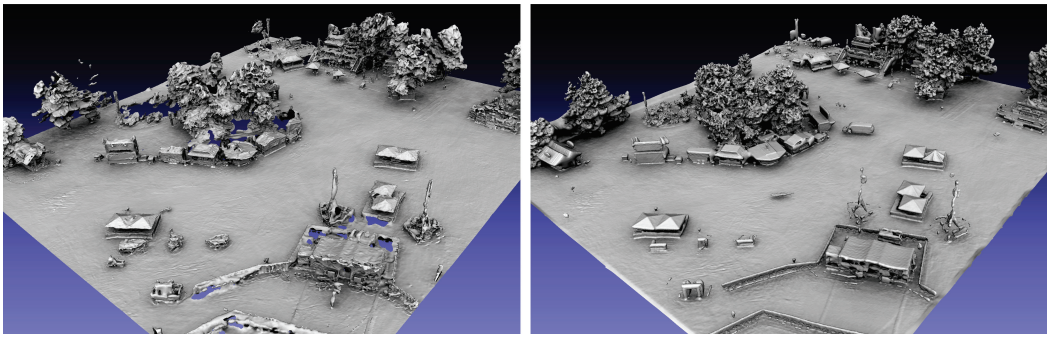

Fig. 4. Left: Mesh obtained from photogrammetric reconstruction of the ground photos. Right: Mesh generated from airborne LiDAR point cloud.

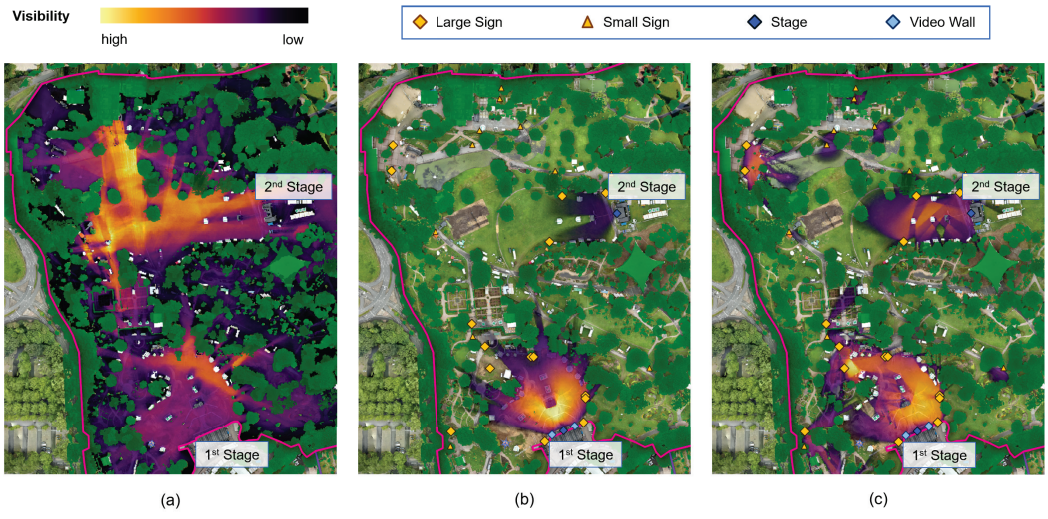

Fig. 5. Results of the visibility analysis: (a) overall visibility, (b) visibility of the two main stages, (c) visibility of emergency exit signs.

Closely related to the last point is the question of the visibility of facilities such as emergency exit signs in general. We therefore located all the emergency exit signs on the site and calculated their (cumulative) visibilities (Fig. 5c). While the stages, as expected, mainly cover the open spaces in front of them, the visibility of the signs extends to larger areas away from the lawns and along the paths. However, there is still a qualitative concentration in the open spaces, leaving the area between the stages without information coverage.

3.4. *Video Analysis*

We utilized the DM-Count model introduced by Wang et al. (2020) exclusively for person counting without capturing additional dynamic information from an input video stream recorded with an Axis P1455 ip camera. We were allowed to mount the camera at a height of 140m on the restaurant platform of the TV tower located inside the park, which is closed to the public throughout the festival. For each 25x25 meter grid cell, the model outputs the total number of individuals present. The results are provided in GeoJSON format using the EPSG:4326 coordinate reference system, enabling

precise georeferencing of the grid.

3.5. *Agent-Based Simulation*

For the agent-based simulation, we focused on the part of the festival site around the two main stages, which attract the majority of the visitors. This area also contains a variety of stalls (merchandise, food, drink, toilets) and the festival organizers were particularly interested in the flow of people between the two stages. The modeled area was manually constructed in QGIS, based on a georeferenced orthophoto generated by the 3D reconstruction module and festival planning data from the organizers. Data collected during the festival at the entrances (number of people/hour) are used to spawn the relevant numbers of agents at the corresponding entrances to the modeled area over the course of the simulation. Agents then move around the simulated festival site as described in Section 2.3. The actual stage program is represented in the model by setting stages as active (a particular act is playing) or inactive (change-over between acts) at the respective time intervals.

In a first scenario, we simulated 5 hours of festival (18:00 to 23:00), with up to 15,000 agents present in the simulation at the same time. We were able to reproduce realistic crowd behavior, with large crowds in front of the stages and queues forming at food and drink stalls and toilets over time. When large numbers of agents move around the area, e.g., at the opening of the festival or to switch between stages at the end of one act, the simulation also shows bottlenecks on the relatively narrow paths of the park (see Fig. 6).

We were able to qualitatively validate the simulation by comparing its results to the data produced by the video analysis (see Fig. 3).

Fig. 6. Snapshot of the ABM showing large crowds in front of the two main stages and smaller crowds forming on the paths. Stages are depicted as green polygons, stalls as orange polygons and obstacles in gray. The ABM layout is overlaid on a satellite picture of the area.

4. Limitations and Future Work

In its current version, our simulation toolkit has limitations with regard to real-time applicability. Firstly, the demonstrated pipeline is not yet fully automated, requiring, e.g., manual generation of the ABM layout and manual placement of facilities for the visibility analysis. Secondly, simulating the realistic behavior of large numbers of festival goers is computationally intensive, thus producing predictions of future situations in real-time is not yet possible.

Future work will not only address these limitations, but will also focus on extending the functionality of the toolkit. For the ABM, we will explore the integration of group behavior, as this influences both movement and communication during evacuations. The planned improvements to the visibility module will include the influence of lighting, other measures in addition to area and volume, and the ability to run GPU-accelerated visibility analyses for facilities.

5. Conclusion

In this paper, we demonstrated the operability of an integrated simulation toolkit for the planning and management of large outdoor events. As shown in Section 3, the work necessary to evaluate certain scenarios related to the safety and security of a festival is considerable and requires the cooperation and integration of several disciplines. By its modular design, our toolkit facilitates the interconnection between these otherwise solitary

insights. We have established a pipeline from data acquisition using drones, via 3D reconstruction, visibility computation to agent-based simulation and video analysis. This lays the groundwork for an easier investigation of further scenarios for the *Juicy Beats* festival and a subsequent extension to other outdoor events.

The illustrated approach not only enhances decision-making for event organizers but also promotes safety and efficiency in crowd management, paving the way for more effective large-scale event planning.

Acknowledgement

We would like to thank the organizers of the *Juicy Beats* festival, Carsten Helmich, Mark Scholtz, Dirk Jöhle and others, for their cooperation, and our colleague Dennis Böhmer for providing laser scanning and photogrammetry data of the festival area. The work described in this paper was carried out within the Fraunhofer Center for the Safety and Security of Socio-Technical Systems (SIRIOS).

References

Batty, M. (2001). Exploring isovist fields: Space and shape in architectural and urban morphology. *Environment and Planning B: Planning and Design 28*(1), 123–150.

Benedikt, M. L. (1979). To take hold of space: Isovists and isovist fields. *Environment and Planning B: Planning and Design 6*(1), 47–65.

Hart, P. E., N. J. Nilsson, and B. Raphael (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics 4*(2), 100–107.

Helbing, D. and P. Molnar (1995). A social force model for pedestrian dynamics. *Phys. Rev. E 51*, 4284–4286.

Kazhdan, M., M. Bolitho, and H. Hoppe (2006). Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, Goslar, DEU, pp. 61–70. Eurographics Association.

Kerbl, B., G. Kopanas, T. Leimkuehler, and G. Drettakis (2023). 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph. 42*, 1–14.

Lu, G., L. Chen, and W. Luo (2016). Real-time crowd simulation integrating potential fields and agent method. *ACM Transactions on Modeling and Computer Simulation 26*(4).

McElhinney, S. (2024, February). Isovist app userguide v1-7.

Meyer, R., A. Schmidt-Colberg, A. Kruse, D. Eberhardt, and C. Köpke (2024). Towards a specification of behaviour models for crowds. In C. Elsenbroich and H. Verhagen (Eds.), *Advances in Social Simulation. ESSA 2023. Springer Proceedings in Complexity*, pp. 495–502. Springer.

Schmidt-Colberg, A., S. Teigler, R. Meyer, and A. Kruse (2024). Towards Modelling Human Behaviour and Warning Message Informativity in Large-Scale Event Evacuation. In *Proceedings of the 23rd International Conference on Modelling and Applied Simulation (MAS 2024)*.

Schönberger, J. L. and J.-M. Frahm (2016). Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Senanayake, G. P., M. Kieu, Y. Zou, and K. Dirks (2024). Agent-based simulation for pedestrian evacuation: A systematic literature review. *International Journal of Disaster Risk Reduction 111*, 104705.

Turner, A., M. Doxa, D. O'Sullivan, and A. Penn (2001). From isovists to visibility graphs: A methodology for the analysis of architectural space. *Environment and Planning B: Planning and Design 28*(1), 103–121.

van Toll, W., T. Chatagnon, C. Braga, B. Solenthaler, and J. Pettré (2021). SPH crowds: Agent-based crowd simulation up to extreme densities using fluid dynamics. *Computers and Graphics 98*, 306–321.

Waechter, M., N. Moehrle, and M. Goesele (2014). Let there be color! — Large-scale texturing of 3D reconstructions. In *Proceedings of the European Conference on Computer Vision*. Springer.

Wang, B., H. Liu, D. Samaras, and M. Hoai (2020). Distribution matching for crowd counting.