LARGE-SCALE AND MICROSCOPIC: A FAST SIMULATION APPROACH FOR URBAN AREAS

Gregor Lämmel (corresponding author)
Forschungszentrum Jülich GmbH
Institute for Advanced Simulation
Jülich Supercomputing
52425 Jülich
Germany
Telephone: +49 2461 61-9314
FAX: +49 2461 61-2810
g.laemmel@fz-juelich.de
www.fz-juelich.de/ias/jsc/cst

Armin Seyfried
Forschungszentrum Jülich GmbH
Institute for Advanced Simulation
Jülich Supercomputing Centre
52425 Jülich
Germany
Telephone: +49 2461 61-3437
FAX: +49 2461 61-2810
a.seyfried@fz-juelich.de
www.fz-juelich.de/ias/jsc/cst

Bernhard Steffen
Forschungszentrum Jülich GmbH
Institute for Advanced Simulation
Jülich Supercomputing Centre
52425 Jülich
Germany
Telephone: +49 2461 61-3437
FAX: +49 2461 61-2810
b.steffen@fz-juelich.de
www.fz-juelich.de/ias/jsc/cst

Submitted: August 1, 2013
Revised: November 15, 2013

5481 words + 5 figures + 1 table = 6981 words

Abstract

Agent based pedestrian simulation models can be distinguished by their granularity. Models that consider the simulation environment as a two dimensional continuous space and perform the simulation in small time steps are usually called microscopic, while models that still represent individual persons but rely on a coarser abstraction of the real world or often called mesoscopic. Macroscopic models only use densities or groups of persons. In many situations a coarse representation is to favor over a finer one because (i) less data has to be collected and processed in order to setup a simulation scenario and (ii) a coarser simulation model usually is less consuming in terms of computational costs compared to a coarser model. Nevertheless, there are still situations where a microscopic simulation is needed and wanted. Examples are crossing pedestrian streams, bidirectional flows at high densities, and the simulation of pedestrians with multiple destinations (e.g. pedestrian movement in shopping malls). One approach that takes advantage of both kinds of models is a hybrid combination in which a microscopic model is applied where needed and a mesoscopic model where plausible. When coupling different models one requirement is that dynamic properties like flow, density and speed are conserved over the models' boundaries. This work focuses on the hybrid combination of a mesoscopic queuing model and a microscopic model that is based on considering obstacles in velocity space. The main contribution of this work is a method for a hybrid coupling that guaranties dynamic properties like flow, density and speed are conserved over the models' boundaries. Furthermore, an efficient way to represent the simulation environment and retrieve dynamic information is discussed. The performance of the proposed model is shown based on a hypothetical large-scale scenario.

INTRODUCTION

Agent based pedestrian simulation models can be distinguished by their granularity. Models that consider the simulation environment as a two dimensional continuous space and perform the simulation in small time steps are usually called microscopic, while models that still represent individual persons but rely on a coarser abstraction of the real world or often called mesoscopic. There are also models that replace individual persons by densities or flows, those models are usually called macroscopic. Microscopic models are usually applied to pedestrian simulation scenarios in particular in the context of buildings evacuations. Microscopic state of the art models (e.g. velocity obstacle models *(*1*)*) can simulate scenarios in the dimension of a few 10,000 agents in real time *(*2*)*. For bigger scenarios, however, those models are still to consuming not only in terms of computational costs but also in terms of effort needed to setup those scenarios.

One way to deal with those large-scale scenarios is to reduce the physical complexity or the model's fidelity *(*3*)*. This can reduce the run time since there are less complex interactions between agents. What is more, the effort to setup scenarios is also reduced because with less complexity also less data about the environment has to be collected and processed. One model that makes use of this approach is the so-called queue model *(*4*)*. With the queue model it is possible to simulate scenarios comprising of several million agents faster than real-time *(*5*)*. Typically the queue model is applied to large-scale scenarios of vehicular traffic. However, in recent times the model has been adapted to deal with large-scale pedestrian simulations as well *(*6, 7*)*. On the downside, there is no direct physical interaction between agents in the queue model. This is in particular problematic when it comes to bidirectional flows since oncoming agents do not interact at all. This makes the queue model for pedestrians suitable only for low density conditions where few such interaction happen or for evacuation scenarios where bidirectional flows are unexpected.

However, in complex situations (like crossing pedestrian streams) it is apparent that the queue model is too coarse. In that case a microscopic or "fine-grained" model like the reciprocal velocity obstacle model *(*1, 8*)* is needed.

The concept of hybrid model coupling is nothing new in the field of transportation research. In general hybrid models can be divided into two groups. (i) There is a large body of research dealing with the coupling of macroscopic and microscopic models (see, e.g. *(*9, 10, 11*)*). (ii) What is more relevant for this work, in recent years there has been a some work regarding the coupling of mesoscopic and microscopic models (see, e.g. *(*12, 13*)*). Furthermore, there is at least one work that combines the whole range from macro- over meso- to microscopic simulations in one hybrid model *(*3*)*.

Most of the existing models are applicable for motorized traffic only even though first steps for a hybrid coupling of different pedestrian simulation models have been undertaken (e.g. *(*14, 15*)*). Independent of whether one wants to model pedestrians or motorized traffic with a hybrid approach the underling requirements for such a coupling seem to be the same. According to *(*13*)* these are:

- Consistency in route choice and network representation
- Consistency of traffic dynamics at mesoscopic-microscopic boundaries
- Consistency in traffic performance and transparent communication and data exchange

The third requirement fulfilled by definition in our example because both models are integrated in the same framework (MATSim toolbox[1]). When working with different frameworks, this needs special care.

This contribution focuses on a new approach coupling pedestrian models of different granularity to a hybrid model. This is realized in a way that ensures that fundamental dynamic properties like the relation between flow, density and speed is conserved over the boundaries of the models involved.

The remainder of this work is organized as follows. The two agent based pedestrian simulation models are introduced and optimization strategies regarding the data representation and information retrieval are discussed. An approach how these models can be coupled to a hybrid model is presented in detail. The performance of this approach is shown based on a hypothetical large-scale scenario. This work concludes with a discussion of open problems and gives an outlook on future work as well.

SIMULATION ENVIRONMENT

The environment is the surrounding of the pedestrians, called agents, where they move along, interact and navigate to get from one location to another. The environment can be divided into two layers, namely the mental environment and the physical environment. The mental environment is the entity where the agents plan their routes through the physical environment, i.e. this is the entity where the high-level behavior is performed. The physical environment can be any spatial extent like buildings, parks, airports, or even large urban areas. The construction and representation of the mental and physical environment is discussed in detail in the following.

Mental Environment

In the simulation, pedestrians are moving through the two-dimensional area given by the environment. Usually the environment is given by architectural drawings or maps and constitutes a geometric model of the reality. Such a representation of the spatial information makes it difficult for computational agents to find appropriate paths through the environment. Therefore it is desirable to have the spatial relation of the environment and its entities captured in a graph based structure. There are efficient methods to make complex route planning on a graph, since for graphs one can rely on simple and fast shortest path algorithms like Dijkstra's algorithm *(*16*)* and it's variations (e.g. A^*, see, e.g., *(*17*)*) for route computation.

The navigation graph needs to guarantee that all entities are reachable and that no edge of the graph intersects any wall or other obstacle in the environment. Furthermore, it must be guaranteed that a path exist in the navigation graph between any two entities.

One class of graphs that fulfill the requirements are the so called visibility graphs *(*18*)*, which connect all mutually visible locations by edges. Visibility graphs have been applied to agent-based pedestrian simulations (see, e.g. *(*19, 20*)*). For the construction of this graph the environment is seen as set of polygonal geometries representing obstacles; the visibility graph is generated by connecting all mutually visible vertices of the polygonal geometries.

The visibility graph can be very complex even for environments with only a few obstacles: If the geometries representing the obstacles have altogether n vertices, then the corresponding visibility graph can have n^2 edges (all vertices are mutually visible). In most situations, the complexity

[1]MATSim stands for Multi-Agent Transport Simulation and is an opensource software for large-scale transport simulations (see, www.matsim.org)

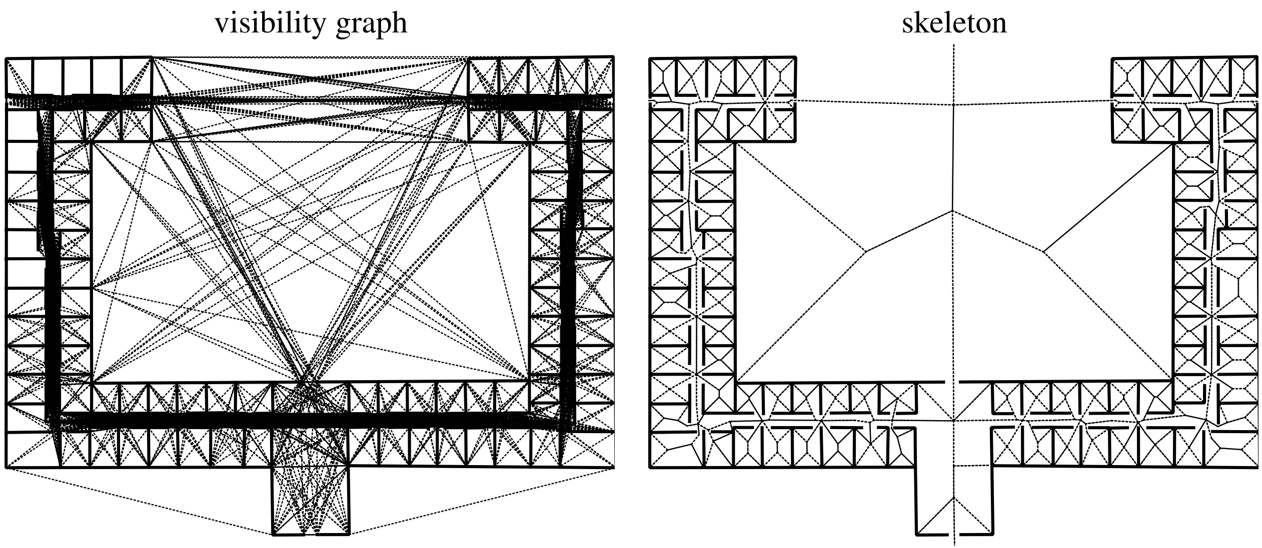


FIGURE 1 Comparison of the visibility graph approach and the skeleton approach. The visibility graph shown on the left side consists of almost 8000 edges, whereas the skeleton approach on the right side only needs about 900 edges.

will be much less than that; still, even simple environments lead to complex visibility graphs as it is illustrated in Figure 1. Another type of graph that is suited for representing the spatial relations in two-dimensional environments is called skeleton or medial axis. Skeletons have been used for navigation in virtual environments before *(*21*)*. The skeleton forms a tree that can be seen as a thin version of the corresponding shape. An intuitive model for constructing a skeleton of a shape is given by *(*22*)*. In that article the analogy of a grass fire propagating as wave fronts is used to construct the skeleton. The algorithm introduced by *(*22*)* is rather complex in terms of computational and implementation costs. An approximate solution in which the two-dimensional space is discretized into a grid like structure is given by *(*21*)*.

Physical Environment

As stated before, the environment where the pedestrians are acting is usually given by architectural drawings or maps. Those drawings represent the obstacles (i.e. walls,) that the pedestrians cannot penetrate. The microscopic simulation model, which is discussed later, makes sure that the pedestrians do not collide with the obstacle. Typically, the obstacles in the environment are given by a set of polygons. When an agent takes care of potential collisions with obstacles in the environment or with other agents, the model has to perform a visibility check first. Obstacles that are not visible from the agent's location do not influence her movement. Even though there are algorithms (i.e. ray casting based algorithms) that can be used to compute visibilities with reasonable effort it is still time consuming when it comes to large-scale scenarios. In this work we propose a simple approach of partitioning the environment in a way that visibility computation can be performed efficiently. The approach is based on the following: If an agent is located inside a convex polygon and the shell of the polygon represents the obstacles, then all of the obstacles are visible from the agent's location by definition. Thus no additional visibility check is necessary. Obviously the environment cannot be represented by a simple convex polygon and usually it is given by a set of polygons with holes. However, any set of polygons with holes can be decomposed into a set of convex polygons. While an exact decomposition is hard to compute *(*23*)* there are algorithms that perform an approximately convex decomposition. In this contribution an algorithm based on *(*24*)* has been chose to perform this task. The algorithm relies on the measurement of the polygon's concavity. The concavity can be see as the depth of the polygon's deepest notch. If this depth exceeds a threshold, then the polygon will be resolved into two less concave polygons. The algorithm is applied recursively on each of the resulting polygons until their concavity is below the threshold. The difference to the original work is that the algorithm has been extended to keep track of newly generated edges and mark them as "opening". This information is needed, since every edge that has the attribute "opening" does not represent an obstacle. An illustration of the result is given in Figure 2. The figure shows a cutout of a building's floor plan. The solid lines represent walls and the dashed lines represent openings connecting neighboring polygons. The polygons "roughly" represent rooms in the building. The proposed approach also helps to determine (mutual) visible agents. Like real pedestrians, a simulated pedestrian (i.e. agent) will only react on visible or perceived agents. By definition perceived agents are those agents that are within a given distance and visible by the agent in question. Thus, when determining these perceived agents one has to make a visibility check. Since the environment is represented by a set of (approximately) convex polygons the visibility check is straightforward and efficiently to perform. Figure 2 gives an illustration.

For performance reason the number of perceived agents might be truncated to n nearest neighbors. In order to perform an efficient nearest neighbor search agents are stored in KD-Trees (see, e.g. *(*18*)*). Each of the sections hold its own KD-Tree, where each of the KD-Trees stores only those agents that are located in its associated section. By using KD-Trees as data structure agents can make spatial queries to efficiently limit potential neighbors.

PHYSICAL SIMULATION

The physical simulation implement the agents' movement. As discussed in the introduction there are different simulation models that can be classified based on their spatio-temporal resolution. A simulation model with a high spatio-temporal resolution is capable of adequately simulating complex situations while a model with a coarser spatio-temporal resolution can be computational more efficient and easier to setup for large-scale scenarios. In this contribution a hybrid approach is proposed that combines the advantage of both models. For such a hybrid model approach it is important that dynamic properties of both systems like flow, density, and speed are conserved over the boundaries [2]. The approach is implemented as an extension to the MATSim toolbox. The following section discusses the simulation model and the transition of agents between these models as well.

Coarse-grained model

The coarse-grained model in this work is a well-established queuing model for transport simulations *(*4, 25*)*. This queuing model is standard model for traffic flow simulation in MATSim. The model was developed to deal with motorized traffic but has meanwhile been adapted so that it can deal with pedestrians as well *(*6, 26*)*. For motorized traffic each street segment (link) is represented as a FIFO (first-in first-out) queue with three restrictions. First, each agent has to remain for a certain time on the link, corresponding to the free speed travel time. Second, a link flow capacity is

[2]Speed and density need to be conserved only approximately, flow exactly.

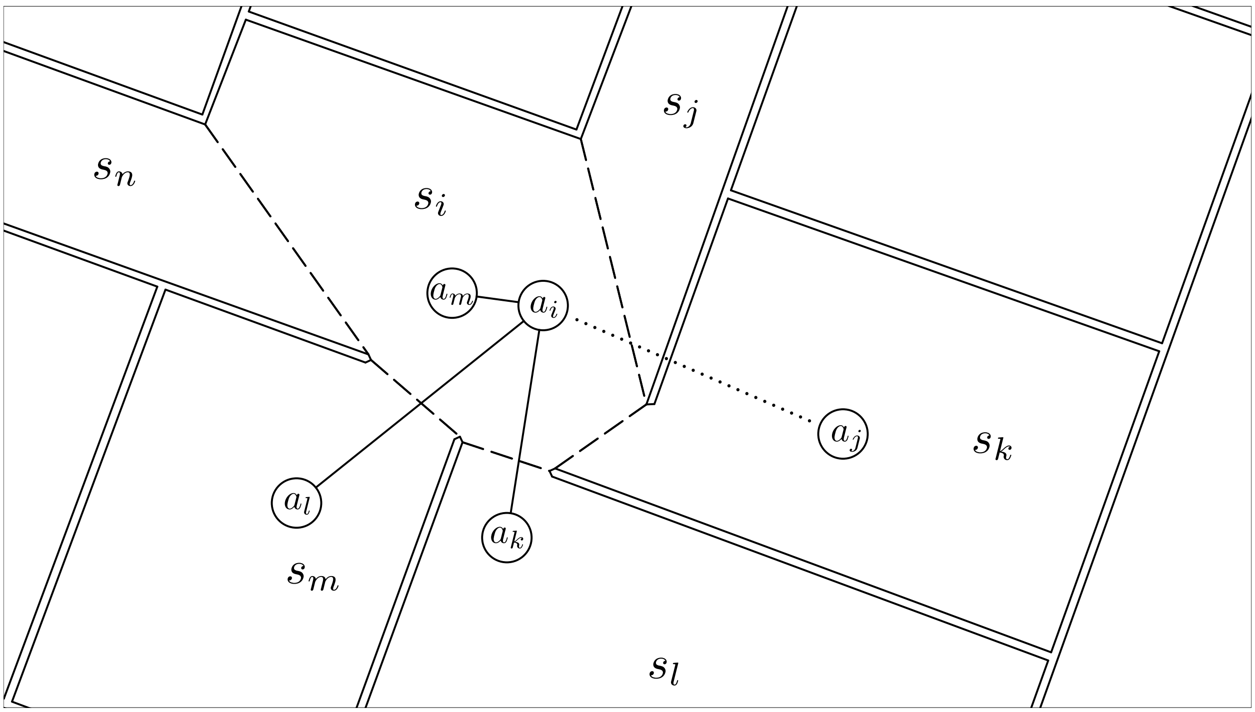


FIGURE 2 Illustration of the approximately convex decomposition and neighborhood computation. The walkable area (e.g. a building's floorplan) is decomposed into approximately convex polygons s. The shells of the polygons define either obstacles (solid lines) or openings connecting neighboring polygons (dashed lines). The sketch shows agent a_i's neighbors. If sections are convex, then agents from same sections are visible by definition (i.e. a_m is visible for a_i). Agents in two neigboring sections are only (mutual) visible if and only if the beeline intersects an opening that both sections have in common. Here, agents a_k and a_l are visible for agent a_i, while agent a_j is not.

defined, which limits the outflow from the link. If, in any given time step, that capacity is used up, no more agents can leave the link. Finally, a link storage capacity is defined that limits the number of agents on the link. If it is filled up, no more agents can enter this link. In principle pedestrian traffic flow rely on the same parameters as vehicular traffic flow. For instance for both one can define a free flow speed, a storage capacity for links and a flow capacity of bottlenecks. The links of the network could derived from the links of the navigation graph that is discussed in section 2.1 or from other data sources like openstreetmap (see, www.openstreetmap.org).

The flow related parameters of a link are the length l, the free flow travel time τ, the flow capacity q, and the storage capacity c. The storage capacity for pedestrians usually is given in person per area. According to *(*27*)* a pedestrian flow comes to a stand still at a density of $\rho^{max} = 5.4\,/m^2$. The storage capacity is determined by the area A that corresponds to the link. What area belongs to which link might be unapparent. Traditionally, the queue model has been applied to vehicular traffic where each link corresponds to a street segment between two intersections. In that case the area of the link is the area of the corresponding street segment. In the current work, however, the queue model is applied to pedestrian scenarios with complex geometries where this mapping might be ambiguous. And, as mentioned earlier, there is no physical interaction between oncoming pedestrian flows. Thus, the model might be suitable for bidirectional flows only at very low densities. There is further research necessary to clarify this. For the time being the queue model is only applied to situations where this area-link mapping is conclusive (e.g. side walks, straight hallways) and bidirectional flows are less important. The storage capacity of a link is:

$$c = A * \rho^{max}. \tag{1}$$

In the queue model the free flow speed v^0 is a link specific parameter that is subject to the nature of the link (e.g. plane, stair, ramp). As a link specific parameter v^0 is the same for all agents on that link. According to *(*27*)* the free speed is $1.34\ m/s$ in the plane. The free speed travel time of link is:

$$\tau = l/v^0. \tag{2}$$

The flow capacity q of link with minimum width w (i.e. the width of the bottleneck) is set to:

$$q = w * 1.2 \frac{1}{m * s}. \tag{3}$$

This reflects a general accepted value (see, e.g., *(*28*)*).

Fine-grained model

When simulating pedestrians in complex situations (e.g. crossing streams or bi-directional flows) microscopically it is important to model the collision avoiding behavior observable with real pedestrians. The computation of collision free trajectories in complex situations is a well-known problem in robotics. For static environments (i.e. situations with only one moving entity and an arbitrary number of static obstacles) collision free trajectories can be computed by approaches that rely on the concept of configuration space obstacles (see, e.g. *(*18*)*). An extension of the configuration space obstacle approach for dynamic environments is the so-called velocity obstacle approach *(29)*. A velocity obstacle is a set of velocities that would lead to a collision at some future time under the assumption that none of the moving entities change their velocity. Later work focuses on the application of the velocity obstacles concept to pedestrian simulations *(*1*,* 8*)*. The work result in a model called ORCA (optimal reciprocal collision avoidance). One problem with the model is that agents can walk with high speeds even under high density conditions. Recently, an extension of the ORCA model has been proposed that correctly reproduce fundamental diagrams observed in pedestrian experiments *(*30*)*. For this work the ORCA model has been reimplemented in MATSim. During the simulation the ORCA model has to chose a collision free velocity for each agent in every time step. In the current work agents follow a pre-computed path along the navigation graph of the environment. Details are discussed in *(*31*)*. The only difference in this work is that agents computed a desired velocity directly, instead of computing a force that pushes them along the navigation graph. Implementation details are not discussed here. The interested reader is referred to the cited works.

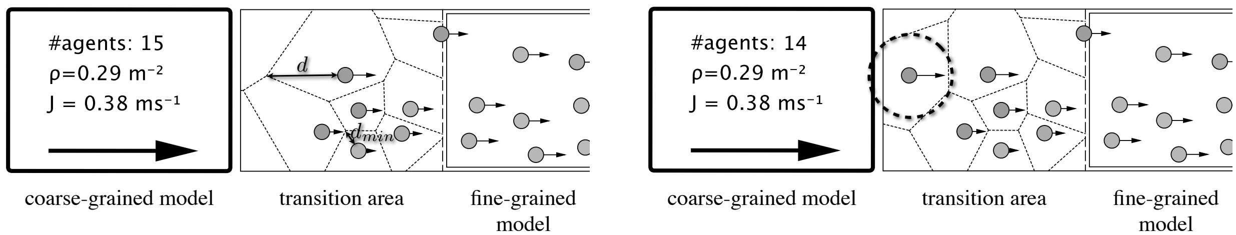
Hybrid model

In the hybrid approach the models are operating independently of each other but have to provide interfaces where a transition of pedestrians between the models is possible. Any spatio-temporal region can have its own simulation model. However, a temporal assignment of a simulation model would make it necessary to switch model assignments while the simulation is running, which is much more challenging than a static spatial model assignment. The dynamic model assignment switching is subject to future research and for the time being we use a static assignment.

One problem when combining models of different granularity is the discontinuity when an entity (i.e. agent) transfers from one model to another. The reason for this discontinuity is the change in temporal and spatial resolution.

The aim of this work is the hybrid combination of the queue model *(*4*)* and an extended version of the ORCA (optimal reciprocal collision avoidance) model *(*8*,* 30*)*. The transition of the agents between the models should have no impact on the overall simulation. But the way an agent transfers from one model to the other defines the boundary conditions of the involved models. In order to reduce the impact of the transition on the overall simulation, agents are transferred gradually. This is done to guaranty that fundamental features like density ρ, flow J, and speed v are retained over the boundaries of the two models. The model transition is possible in both directions (i.e. from the queue model to the ORCA model and vice versa). Both transitions have to be treated separately. An agent is generated in the ORCA model if there is at least one available in the overlap zone of the queue model and the ORCA model is capable of accommodating her in a suitable location. To find these locations, a Voronoi diagram *(*32*)* of the agents in the transition area is computed and the corner points of the Voronoi cells are tested for distance d to the neighbors (a corner normally has three neighbors at identical distance). If more than one location is available, a weighted random selection is done, where the weights are constructed from $(d - d_{min})^2$. The agent is placed at the selected corner and given the average speed of her three neighbors. This gives a fairly smooth start for her in the ORCA model. The transition from the coarse-grained model (queue model) to the fine-grained model (ORCA model) is illustrated in Figure 3.

The density in the ORCA model is this way adjusted automatically, and the transport in the queue model link is matched to the acceptance capacity of the ORCA area following. The width of the ORCA model transition area has to be chosen such that its transport capacity is at least as big as the acceptance capacity of the ORCA area following, and it is advisable to make it not too much bigger, because a strong reduction in capacity causes a bottleneck with a capacity smaller than the minimum of the two joining parts.



(a) Situation with 15 agents are "waiting" for the transition from the coarse-grained model to the fine fine-grained model.

(b) Situation after one additional agent is transfered. After the transition the Voronoi diagram is re-computed.

FIGURE 3 Illustration of the transition from the coarse-grained model to the fine-grained model. Transferring agents are inserted at corner points of Voronoi cells. The corner points are selected based on a weighted random selection depending on the distances from agents already in the transition area and the corner points of the Voronoi cells.

The transition from the fine-grained model to the coarse-grained model is much less challenging. Agents are transferred from the ORCA model onto the down stream queue model link as soon as say cross a virtual finish line. In case there is no space left on the down stream link agents stop and wait at the "finish line" until space becomes available. This approach works fine for the queue model, however, a disappearance of agents that just have crossed the "finish line" would produce artifacts in the ORCA model. The reason is, that if an agent disappears in dense conditions a direct following agent would accelerate since more space in front of her becomes available. This acceleration again would let further agents also accelerate. To avoid these artifacts we propose a simple solution, where agents that have crossed the "finish line" exist in both models for a while. When an agent is transferred to the down stream queue model link then a copy of her still exist in the ORCA model for a couple of simulation steps. These copy only dissolves when she crosses a second virtual finish line. The location of this second finish line coincides with the beginning of the transition area (see, Figure 3).

This means those leaving agents also influence agents that are transferring from the queue model to the ORCA model via the corresponding transition area. This makes that approach also suitable for bidirectional flows. For bidirectional traffic, the two concepts are combined in a way that the outgoing ORCA model is a little longer than the incoming one. For the generation of agents in the transition zone, the choice is modified in a way that positions with neighbors moving in the same direction are strongly preferred, and the velocity generated always has a positive component in the desired direction.

EXPERIMENTS

In this section we demonstrate the performance of the proposed hybrid model. The field of application for this kind of model is doubtlessly scenarios with thousands or even hundred of thousands of pedestrians like the simulation of large shopping malls, sports events or music festivals. However, the model is still not complete (e.g. the simulation of pedestrians in queues needs further research). Therefore we are not going to applied the hybrid model to a real world scenario at this point. Instead, we demonstrate the hybrid model performance on a fictional near real world scenario, the simulation of New York City's Grand Central Terminal (GCT). According to Wikipedia (see, en.wikipedia.org/wiki/Grand_Central_Terminal) GCT is the world largest train station. On the GCT's official web page (see, www.grandcentralterminal.com) one reads that on a normal day 750,000 people pass through the station. A microscopic simulation of the entire station with 750,000 agents is still beyond current computational ability.

We created a simulation scenario with the following setup:

- The microscopic ORCA model is applied to GCT's Main Concourse. The floor plan has been roughly reproduced and is not true to scale.
- The accesses to the platforms are simulated by the queue model, the platforms itself are not modeled (i.e. agents are inserted at the accesses to the Main Concourse). The width of the accesses are set to $3.5m$ at will.
- The GCT's surrounding is also simulated by the queue model, where the width of all side walks in the area are set to $4m$ also at will.
- The synthetic population consist of 375,000 agents. Each agent follows a daily plan. Plans are model as activity chains. The activity locations and activity end times are chosen to emulate real world commuter behavior (including morning and evening peak). Every agent has to pass through GCT's Main Concourse twice (once in the morning and once in the evening). Thus, 750,000 person movements are simulated over the whole day.

An overview of the chosen setup is depicted in Figure 4. The testbed for the simulation is a mobile computer with a 2.3 GHz Intel Core i7 CPU, 8 GB DDR3 RAM @ 1600 MHz, and a NVIDIA GT 650M video card. The simulation software (MATSim) is implemented in JAVA. The JAVA version is 1.6.0_51. The JAVA VM was called with arguments "-Xmx5g" and "-XX:+UseCompressedOops".

Three simulation runs of a whole day (6:00 am to 11:00 pm) in the fictional GCT scenario have been performed. When a simulation is running MATSim writes all events that happen (from the coarser agents' activity end or link enter events up to microscopic XY-coordinates for each agent in every time step) steadily in a compressed events file. The creation of this file consumes a considerable amount of the overall run time. Thus, an aggregation while the simulation is running might be useful. To demonstrate this matter two simulation runs have been performed. In *run 1* the individual trajectories in the ORCA model are dropped. *Run 2* is basically the same run but this time individual trajectories in the ORCA model are recorded and what is more, the simulation is visualized on-the-fly. Visualizer snapshots at various zoom levels are depicted in Figure 4. To measure the scalability, an additional run (*run 3*) with 750,000 agents (i.e. 1,500,000 person movements) has been performed. General performance measurements are given in Table 1. A more detailed picture is given in Figure 5(a).

The validity of both the ORCA and queue simulation model has been investigated elsewhere (see, e.g. *(*26*,* 30*,* 31*)*). Thus, we only looked at the interface coupling both simulation models. As stated before, the flow of pedestrians needs to be retained over the model boundaries. Therefore, the flow of both models has been compared (see, Figure 5(b)). For the queue model the flow corresponds to the flow at the cross-section of the corresponding queue link. For the ORCA model the flow is determined by a method based on Voronoi diagrams as introduced in *(*33*)*. The

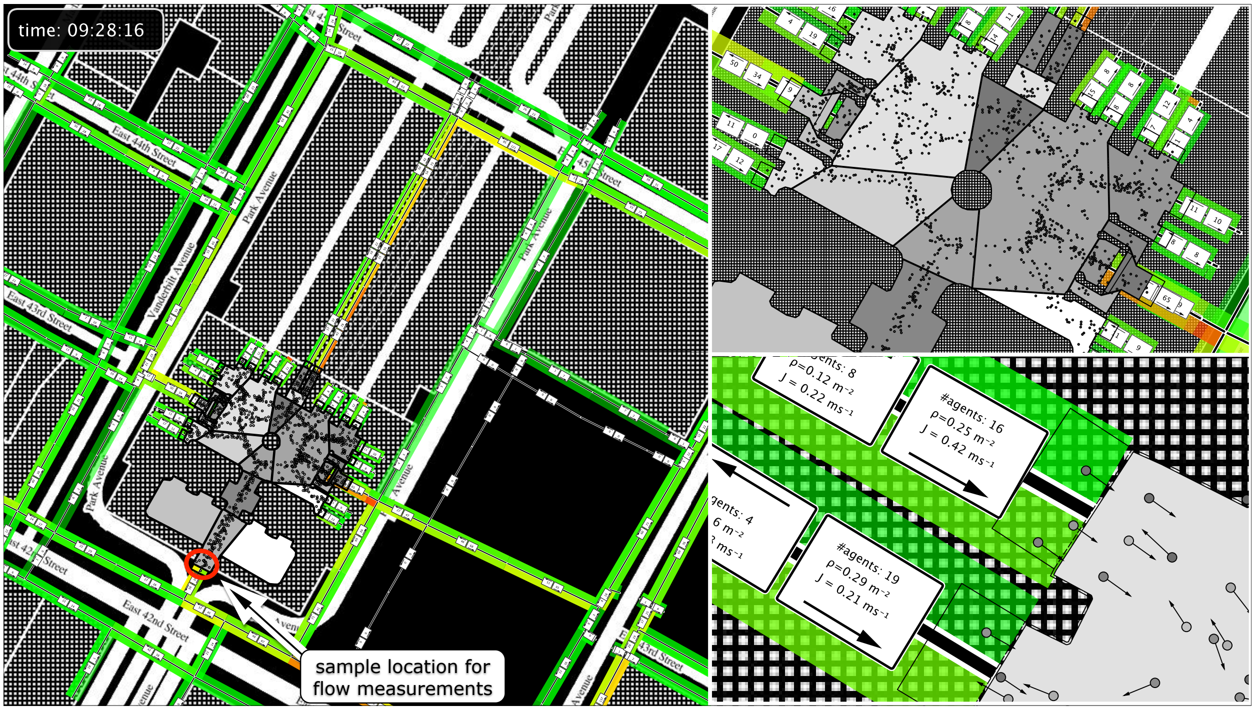
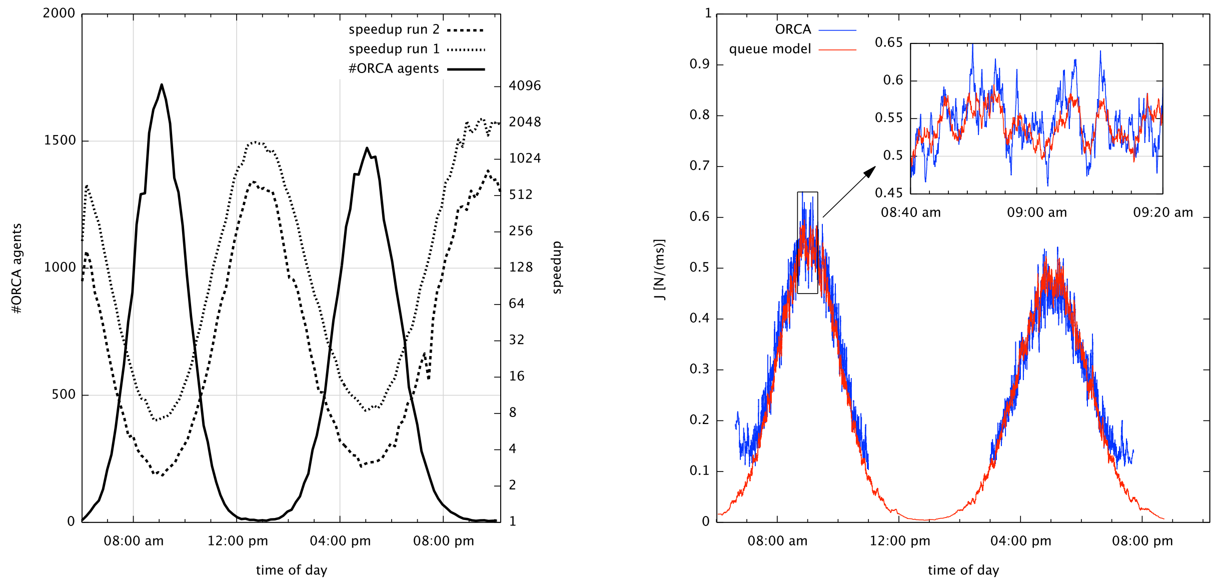


FIGURE 4 Visualizer snapshots at various zoom levels. Link colors correspond to the densities (green low, red high). Dynamic properties like the agents' velocities in the ORCA model and flow characterisitcs (number of agents, density ρ, and flow J) for queue model links are depicted at the highest zoom level (bottom right).

	run 1	*run 2*	*run 3*
#agents	375,00	375,00	750,000
max #ORCA agents	1,745	1,745	4,383
overall runtime	00:36:01	01:50:52	2:43:02
max memory	2.3 GB	2.7 GB	4.3 GB
events file size	292 MB	11 GB	642 MB

TABLE 1 Performance measurements of *run 1*, *run 2*, and *run 3*.

flow in the ORCA model (blue curve) is hard to determine for low densities that's why the blue curve is interrupted.



(a) Speedup (ratio simulated time/wallclock time) and number of agents simulated by the ORCA model over time of day. The number of ORCA agents peak at 1,745. The minimum speedup is about 8 (*run 1*) and 2 (*run 2*) respective. (Note: Speedup is in log scale.)

(b) Comparison of pedestrian flow over time of both simulation models for a sample location at the models' boundaries (see, Figure 4 for location). Both flows are almost the same. The ORCA model shows more variation.

FIGURE 5 Characterization of the hybrid model's behavior.

DISCUSSION

The results presented in the previous section show that the proposed hybrid approach is capable to simulate large-scale microscopic pedestrian scenarios with reasonable computational effort. Nevertheless, there are still some open issues that need to be addressed. Agents in the microscopic ORCA model do not behave as one would expect from real people. Instead, agents with the same destination move in one line, which looks more like ant trails and not like trajectories observable in real situations. The reason for this behavior is the way agents perform their global navigation. As discussed, they update their desired velocity in each time step of the simulation based on a static navigation graph. Thus, two consecutive agents with the same destination also have (almost) the same desired velocity (direction and speed). This behavior leads to situations where groups of agents "clump" despite open space is available in the surroundings. In other words, agents do not take detours to avoid dense groups but force their way through those groups. To solve this issue agents would need to make some kind of density analysis in particular for open areas like GCT's Main Concourse and then create feasible paths to avoid dense regions on-the-fly. In transport science this approach is also known as within-day re-planning. A different approach that could also solve the problem of "clumping" agents is iterative learning (or day-to-day re-planning). In case the navigation graph leaves enough alternatives, agents can learning by trial-end error to find feasible paths to the desired destinations. The result would be a user equilibrium in terms of travel times (see (34) for an approach concering this matter). However, an iterative approach needs to repeat the simulation many times, which would increase the computation time dramatically. In underlying context those local artifacts seem to influence the gloabl outcome only marignaly. However, in other scenarios this might be different. Futher studies are necessary to clearify this. Another issue what we would like to discuss is the simulation of pedestrians with a queue model. The problem that can occur is that bidirectional flows are not simulated correctly. This is because in the queue model oncoming agents do not interact. Still, for unidirectional flows (e.g. evacuation scenarios) and also for bidirectional flows of low densities[3] the queue model gives meaningful results.

CONCLUSION AND OUTLOOK

A new hybrid coupling approach for pedestrian simulation models of different granularity has been presented. The proposed approach conserves dynamic properties of the coupled models like flow, density, and speed over the model boundaries. The performance of the hybrid approach has been demonstrated on a large-scale scenario with 375,000 (750,000 respective) commuting agents. The computing time for a whole day is about 40 minutes (without live visualization) or two hours (with live visualization). In future work it is planed to improve the agents' navigation so that the global behavior becomes more realistic. Furthermore, we want to improve the simulation models to reproduce correctly bidirectional pedestrian flows also under high density conditions. A part of this work will be the model validation with experimental data gathered e.g. in (35). Another interesting research topic that will be addressed in future is the extension of the hybrid model so that a temporal model assignment becomes possible. For the GCT scenario this could mean that the Main Concourse is only simulated by the ORCA model for times where a higher number of agents are in the scene and with the faster queue model else. We also want to couple macroscopic and mesoscopic models. This will be useful for optimizations, where simulations have to be repeated many times for slightly varied scenarios.

ACKNOWLEDGMENTS

This project was funded in part by the German Ministry for Education and Research (BMBF) under grant 13N11382 ("GRIPS") and by the German Research Foundation (DFG) under grant NA682/5-1.

References

1. van den Berg, J., M. Lin, and D. Manocha, Reciprocal Velocity Obstacles for real-time multi-agent navigation. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, pp. 1928–1935.

2. Guy, S., J. Chhugani, C. Kim, N. Satish, M. Lin, D. Manocha, and P. Dubey, ClearPath: highly parallel collision avoidance for multi-agent simulation. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, 2009, SCA '09, pp. 177–187.

3. Yang, Q. and D. Morgan, *A Hybrid Traffic Simulation Model*. Annual Meeting Preprint 06-2582, Transportation Research Board, Washington D.C., 2006.

[3]Taking the results from (35) uni- and bidirectional flows seem to be equal for densities up to $1/m^2$. How to simulate bidirectional flows at higher densities is still an open question.

4. Simon, P., J. Esser, and K. Nagel, Simple queueing model applied to the city of Portland. *International Journal of Modern Physics*, Vol. 10, No. 5, 1999, pp. 941–960.

5. Balmer, M., K. Nagel, and B. Raney, Large scale multi-agent simulations for transportation applications. *J. of Intelligent Transport Systems*, Vol. 8, 2004, pp. 205–223.

6. Lämmel, G., H. Klüpfel, and K. Nagel, The MATSim Network Flow Model for Traffic Simulation Adapted to Large-Scale Emergency Egress and an Application to the Evacuation of the Indonesian City of Padang in Case of a Tsunami Warning. In *Pedestrian Behavior* (H. Timmermans, ed.), Emerald Group Publishing Limited, 2009, chap. 11, pp. 245–265.

7. Lämmel, G. and K. Nagel, *Multi agent based large-scale evacuation simulation*. Annual Meeting Preprint 09-2135, Transportation Research Board, Washington D.C., 2009, also VSP WP 08-13, see www.vsp.tu-berlin.de/publications.

8. van den Berg, J., S. Guy, M. Lin, and D. Manocha, Reciprocal n-body collision avoidance. In *Robotics Research: The 14th International Symposium ISRR*, Springer, Vol. 70, 2011, pp. 3–19.

9. Helbing, D., A. Hennecke, V. Shvetsov, and M. Treiber, Micro- and Macro-Simulation of Freeway Traffic. *Mathematical and Computational Modelling*, Vol. 35, 2002, pp. 517–547.

10. Bourr, E. and J.-B. Lesort, Mixing Microscopic Representations of Traffic Flow: Hybrid Model Based on Lighthill-Whitham-Richards Theory. *Transportation Research Record*, Vol. 1852, 2003, pp. 193–200.

11. Espié, S., D. Gattuso, and F. Galante, *A HYBRID TRAFFIC MODEL COUPLING MACRO AND BEHAVIOURAL MICRO SIMULATION*. Annual Meeting Preprint 06-2013, Transportation Research Board, Washington D.C., 2006.

12. Burghout, W., H. Koutsopoulos, and I. Andréasson, Hybrid Mesoscopic-Microscopic Traffic Simulation. *Transportation Research Record*, Vol. 1934, 2005, pp. 218–225.

13. Burghout, W. and J. Wahlstedt, Hybrid Traffic Simulation with Adaptive Signal Control. *Transportation Research Record*, Vol. 1999, 2007, pp. 191–197.

14. Chooramun, N., P. Lawrence, and E. Galea, Implementing a Hybrid Space Discretisation within an Agent Based Evacuation Model. In *Pedestrian and Evacuation Dynamics 2010* (R. Peacock, E. Kuligowski, and J. Averill, eds.), Springer, Berlin, Heidelberg, New York, 2011, pp. 449–458.

15. Anh, N., Z. Daniel, N. Du, A. Drogoul, and V. An, A Hybrid Macro-Micro Pedestrians Evacuation Model to Speed Up Simulation in Road Networks. In *AAMAS 2011 Workshops* (F. Dechesne, ed.), Springer, LNAI 7068, 2012, pp. 371–383.

16. Dijkstra, E., A note on two problems in connexion with graphs. *Numerische Mathematik*, Vol. 1, 1959, pp. 269–271.

17. Russel, S. and P. Norvig, *Artificial intelligence: A modern approach*. Series in Artificial Intelligence, Prentice Hall, 1995.

18. de Berg, M., M. van Kreveld, M. Overmars, and M. Schwarzkopf, *Computational Geometry*. Spinger, 2000.

19. Gloor, C., P. Stucki, and K. Nagel, Hybrid techniques for pedestrian simulations. In *Cellular automata, Proceedings* (P. Sloot, B. Chopard, and A. Hoekstra, eds.), Springer, No. 3305 in Lecture Notes in Computer Science, 2004, pp. 581–590.

20. Nishinari, K., A. Kirchner, A. Nazami, and A. Schadschneider, Extended floor field CA model for evacuation dynamics. *IEICE Transactions on Information and Systems*, Vol. E87-D, No. 3, 2004, pp. 726–732.

21. Chaudhuri, P., R. Khandekar, D. Sethi, and P. Kalra, An Efficient Central Path Algorithm for Virtual Navigation. In *Proceedings of the Computer Graphics International*, IEEE Computer Society, Washington, DC, USA, 2004, pp. 188–195.

22. Blum, H., A Transformation for Extracting New Descriptors of Shape. *Models for the Perception of Speech and Visual Form*, 1967, pp. 362–380.

23. Lingas, A., The power of non-rectilinear holes. In *Automata, Languages and Programming* (M. Nielsen and E. Schmidt, eds.), Springer Berlin Heidelberg, Vol. 140 of *Lecture Notes in Computer Science*, 1982, pp. 369–383.

24. Lien, J.-M. and N. Amato, Approximate convex decomposition of polygons. *Computational Geometry*, Vol. 35, 2006, pp. 100–123.

25. Gawron, C., An Iterative Algorithm to Determine the Dynamic User Equilibrium in a Traffic Simulation Model. *International Journal of Modern Physics C*, Vol. 9, No. 3, 1998, pp. 393–407.

26. Lämmel, G., *Escaping the Tsunami: Evacuation Strategies for Large Urban Areas. Concepts and Implementation of a Multi-Agent Based Approach*. Ph.D. thesis, TU Berlin, 2011.

27. Weidmann, U., *Transporttechnik der Fussgänger*. Literature research, Institut für Verkehrsplanung und Transportsysteme, ETH Zürich, ETH-Hönggerberg, CH-8093 Zürich, 1993, in German.

28. Predtetschenski, W. and A. Milinski, *Planning for Foot Traffic in Buildings*. Amerind Publishing Co. Pvt. Ltd., New Delhi, 1978.

29. Fiorini, P. and Z. Shiller, Motion Planning in Dynamic Environments Using Velocity Obstacles. *The International Journal of Robotics Research*, Vol. 17, No. 7, 1998, pp. 760–772.

30. Curtis, S. and D. Manocha, Pedestrian Simulation using Geometric Reasoning in Velocity Space. In *Pedestrian and Evacuation Dynamics 2012* (U. Weidmann, U. Kirsch, and M. Schreckenberg, eds.), Springer, in press.

31. Lämmel, G. and M. Plaue, Getting out of the way: collision avoiding pedestrian models compared to the real world. In *Pedestrian and Evacuation Dynamics 2012* (U. Weidmann, U. Kirsch, and M. Schreckenberg, eds.), Springer, in press.

32. Voronoi, G., Nouvelles applications des paramètres continus à la théorie des formes quadratiques. *Journal für reine und angewande Mathematik*, Vol. 133, 1908, pp. 97–178.

33. Steffen, B. and A. Seyfried, Methods for measuring pedestrian density, flow, speed and direction with minimal scatter. *Physica A: Statistical Mechanics and its Applications*, Vol. 389, No. 9, 2010, pp. 1902–1910.

34. Kretz, T., K. Lehmann, and T. Friderich, Selected Applications of a Dynamic Assignment Method for Microscopic Simulation of Pedestrians. In *European Transport Conference 2013*, 2013, http://abstracts.aetransport.org/paper/index/id/125/confid/1.

35. Zhang, J., W. Klingsch, A. Schadschneider, and A. Seyfried, Ordering in bidirectional pedestrian flows and its influence on the fundamental diagram. *Journal of Statistical Mechanics: Theory and Experiment*, Vol. 2012, No. 02, 2012, p. P02002.