Advances in Engineering Software 60-61 (2013) 98–103

Contents lists available at SciVerse ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Parallel real time computation of large scale pedestrian evacuations

Armel Ulrich Kemloh Wagoum [a,*], Bernhard Steffen [a], Armin Seyfried [a,b], Mohcine Chraibi [a,b]

[a] Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, 52428 Jülich, Germany
[b] Computer Simulation for Fire Safety and Pedestrian Traffic, University of Wuppertal, Pauluskirchstraße 7, 42285 Wuppertal, Germany

ARTICLE INFO

Article history:
Available online 9 November 2012

Keywords:
Evacuation
Simulation
Real time computation
Force-based model
Parallelization
Visualization

ABSTRACT

Usually, modeling of the evacuations is done during the planning and authorizing process of office buildings or large scale facilities, where computing time is not an issue at all. The collaborative Hermes project [1] aims at improving the safety of mass events by constructing an evacuation assistant, a decision support system for heads of operation in an actual evacuation. For this, the status (occupancy and available egress routes) of a facility is constantly monitored with automatic person counters, door sensors, smoke sensors, and manual input from security staff. Starting from this status, egress is simulated faster than real time, and the result visualized in a suitable fashion to show what is likely to happen in the next 15 min. The test case for this evacuation assistant is the clearing of the ESPRIT Arena in Düsseldorf which holds 50,000–65,000 persons depending on the event type. The on site prediction requires the ability to simulate the egress in ≈2 min, a task that requires the combination of a fast algorithm and a parallel computer. The paper will describe the details of the evacuation problem, the architecture of the evacuation assistant, the pedestrian motion model employed and the optimization and parallelization of the code.

1. Introduction

A number of accidents in recent mass events (Tembisa, South Africa, July 6, 2010 with 10 severely injured, Love Parade in Duisburg, Germany, July 24, 2010 with 21 dead, and Bremen, Germany, September 27, 2010 with 20 injured, 1 severe) have again made it clear that the high density of pedestrians occurring in large events poses a risk even without any aggressive behavior of the participants. If need arises to clear a large facility rapidly – because of a fire, a bomb thread or another reason – the danger is much more imminent than in normal operation, so proper guidance to safe (non jamming) egress routes is highly important. However, security staff often lacks the necessary information. In all three incidents mentioned above there are claims that police action actually contributed to the accident.

To contribute to the improvement of the situation, the Hermes project was started in 2008 in order to evaluate the potential of crowd movement simulations for predicting dangerous situations. An evacuation assistant system will give security staff and police a sound base for decisions to open or close some pathways or guide the crowd in some direction. The first step to this goal is a monitoring of the present situation – automatic counting of people as they enter or leave areas and checking the status of doors and pathways. At present, estimation of crowd numbers is usually done by sight, and numbers differ widely depending on who does the estimate. This information then enters the central part of the evacuation assistant, a program running on dedicated hardware and calculating constantly a 15 min prediction of an evacuation starting from the present situation. To be useful, this calculation must not take more than a few minutes, otherwise it would often come too late for security action. Finally, the results of the status monitoring and especially of the simulations have to be visualized in a suitable way so that decision makers can rapidly grasp the important aspects of the situation and are not flooded by irrelevant details. Obviously, simply visualizing the process of the evacuation in real time as is often done in the planning of a facility is not fast enough.

The time requirements for the simulation are a severe problem, so a moderately parallel computer is employed and much effort will go into making the calculation fast enough. Details will make up the main part of the paper.

2. The architecture of the evacuation assistant

The concept of the evacuation assistant will be tested in summer 2011 in the ESPRIT arena in Dsseldorf. This is a multi-purpose facility for sports (mostly football), concerts and other events. The grandstand of the arena has 51,500 seats (Fig. 1), about half of them in the lower and in the upper part. In events that do not need the soccer field for stage space, up to 14,000 people can be placed on the field. The lower grandstand has 26 portholes and four large exits in the corners, the upper grandstand has 32 portholes and a

* Corresponding author. Tel.: +49 2461 614193; fax: +49 2461 616656.
E-mail addresses: u.kemloh@fz-juelich.de (A.U. Kemloh Wagoum), b.steffen@fz-juelich.de (B. Steffen), a.seyfried@fz-juelich.de (A. Seyfried), m.chraibi@fz-juelich.de (M. Chraibi).

0965-9978/$ - see front matter
http://dx.doi.org/10.1016/j.advengsoft.2012.10.001

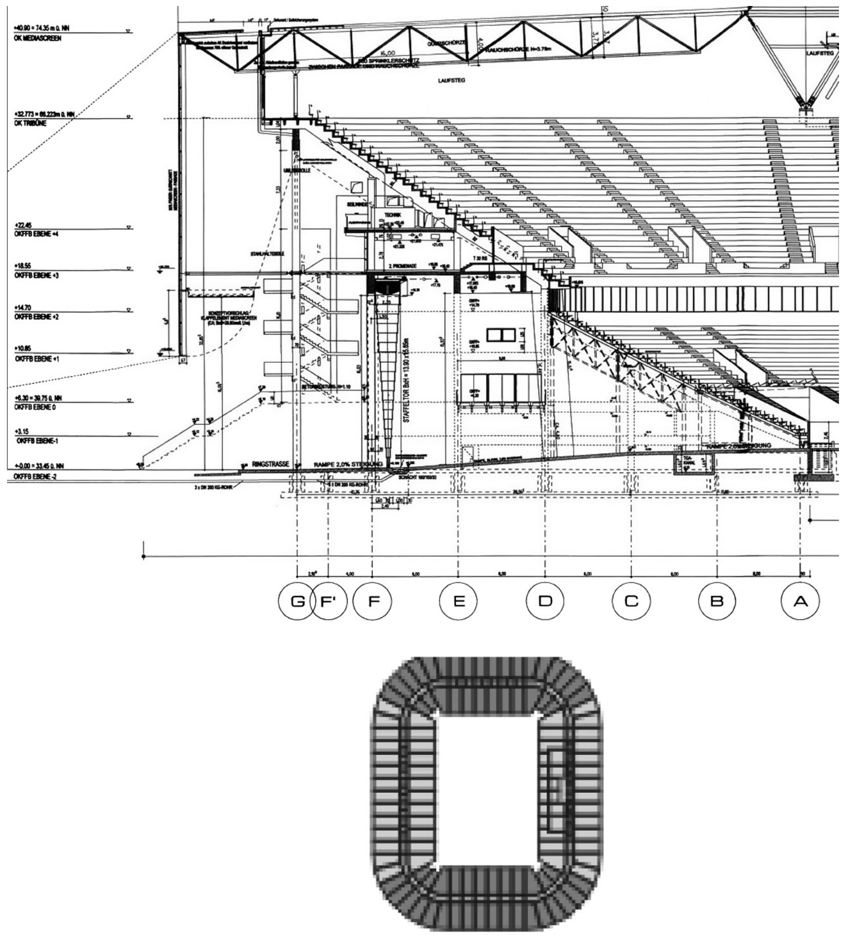

Fig. 1. Cut of part and schematic top view of the ESPRIT arena.

number of exits at the top that are not used in normal operation and will probably not be used by many persons in emergency unless they are forcefully directed there. The central area has three very wide exits at the sides which lead directly outside through wide corridors. The portholes from the grandstands lead to a ring shaped hall (lower $\approx$20 m wide, upper $\approx$10 m wide, top $\approx$5 m). The lower hall has doors directly to the outside area, the upper ones to outside staircases. These halls have rolling gates that separate radial sections. In the halls are plenty of fixed and movable service stands which are obstacles to free movement, so the way from the portholes to the outer door is usually not straight and streams from different portholes may merge and partially cross each other. Streams for upper and lower grandstand do not mix inside the building, so they can be considered independently. Once people have reached the outside area, there are wide passways leading down a slope away from the building, and the slope itself is not too steep to be used by persons with at least normal mobility. Therefore everybody in the outside area is considered to be in safety during an evacuation. The area for the test of the evacuation assistant is about half of the lower grandstand and part of the upper grandstand. The limits are placed such that mixing of the pedestrian stream from the test area with those from the other area can be expected to be minimal. Of course, a testing of the entire facility would be much better, but the budget does not allow a full instrumentation. If the evacuation assistant proves helpful enough, further installations will follow.

The first step for the evacuation assistant (Fig. 2) is the monitoring of the present status. For this, all doors in the test areas as well as those leading into it are equipped with infrared based person counters, such that the number of persons in any area is known at any time. Absolute accuracy is neither possible nor necessary, first tests show an error rate about 3%, which will be improved by tuning. In some events, especially pop music concerts, counting persons in the area optically is not possible because the spectator area is dark and vision may be further hindered by smoke from the stage. A second set of sensors monitors the opening and closing of the doors. Further, the smoke sensors that are now connected to a central display will also be connected to the evacuation assistant; an area filled with smoke will not be considered a possible egress route unless this is specified manually by the operation management. Finally, the security staff distributed in the facility can report any unusual conditions to the operation management which is able to enter information manually.

All this information is collected in a front-end computer of the evacuation assistant and processed for display. It is further used to define an egress route for every person which is based on choosing the nearest visible exit. A macroscopic network model is used to get a quick overview and improve the routing. As main task, a microscopic egress simulation is being run constantly to predict how the egress would work starting from the present situation. This simulation is run on a parallel cluster with 25 Nehalem nodes, each having two processors with six cores. At present, this calculation takes about 4 min with potential for further optimization. The goal is 2 min. The results of the simulation are transported back to the front-end, where they are processed for various modes of display. There will be a global display giving time averages of densities, fluxes, and speeds which is supposed to be active constantly. From this, the operation management can zoom in to

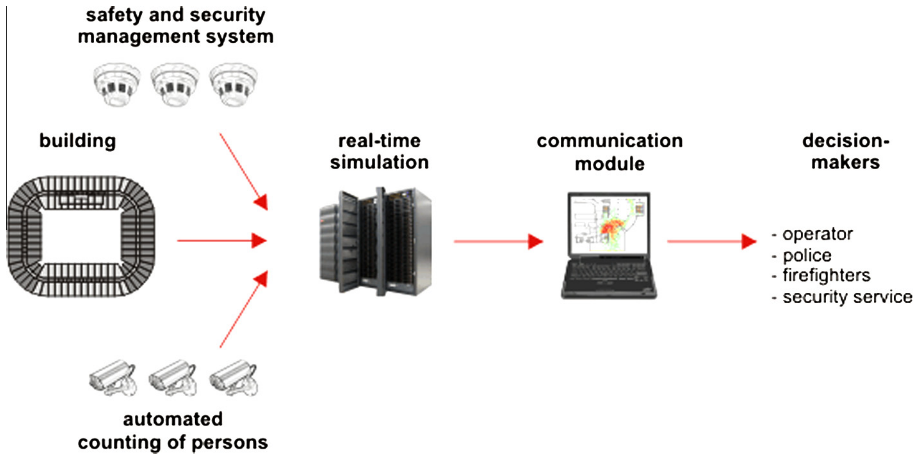

Fig. 2. Components of the evacuation assistant *courtesy S. Holl.*

examine critical areas in space and time and change the mode of display for these areas. The present rules say that after an incident, the management has got 10 min to decide on whether an evacuation is needed and how it is performed, so it would be possible to run two egress simulation, e.g. the rolling gates opened or closed, before deciding what to do. The management is connected to the staff in the facility by radio link and can at any time order to open or close doors or to direct people in a specified direction.

Once the egress has started, the conditions – especially the occupation of rooms – will change rapidly. At the end of every simulation, the measured conditions will be compared with the predicted conditions for some adjustments of parameters and the simulation will be started again with present conditions and improved parameters. This way during egress the simulation will be closer to the actual process than an *a priori* simulation could be.

3. The simulation kernel

The most critical part of the evacuation assistant is the rapid calculation of the microscopic egress simulation. The requirement that the egress of the ≈20,000 persons in the testing area (and 65,000 for a full installation of the assistant) must be modeled within 2 min requires optimal use of the available computer resources. These resources have to be on site, as events requiring evacuations may have the potential to break communication lines.

There are a number of commercial software tools for microscopic egress simulation. In a test case, all tools tested could predict evacuation times of a high rise building accurately enough [2] and give further useful information. However, all had noticeable weaknesses for specific situations and were not correct down to the degree of detail they provide. Furthermore, they are routinely used in planning where computing time is not an issue and therefore are not fast enough. This means that a custom build software tool has to be constructed.

There are three different types of models for pedestrian dynamic: cellular automata models [3,4], rule based models[5,6], and force based models [7,8]. Cellular automata models are the fastest by a good margin, but they are based on a fixed – usually quadratic – mesh and therefore have problems modeling curved geometries and persons moving at oblique angles as is prevailing in the halls. Rule based models are better in this respect, but some tests showed that the models at hand had problems reproducing some experimental data of a crucial feature of the arenas geometry, so a rule based model would have to be modified at least. For simulation we will use the generalized centrifugal force model (GCFM) from [9,10].

3.1. The generalized centrifugal force model

Force-based models take Newtons second law of dynamics as a guiding principle. Given a pedestrian i with coordinates $\overrightarrow{R_i}$ we define the set of all pedestrians that influence pedestrian i at a certain moment as

$$\mathcal{N}_i := \{j : \|\overrightarrow{R_j} - \overrightarrow{R_i}\| \leqslant r_c \wedge i \text{ “feels” } j\} \tag{1}$$

where r_c is a cutoff radius. We say pedestrian i “feels” pedestrian j if the line joining their centers of mass does not intersect any obstacle. In a similar way we define the set of walls or borders that act on pedestrian i as

$$\mathcal{W}_i := \{w : \|\overrightarrow{R_{w_i}} - \overrightarrow{R_i}\| \leqslant r_c\} \tag{2}$$

where $w_i \in w$ is the nearest point on the wall w to the pedestrian i. Thus the equation of movement is:

$$m_i \ddot{\overrightarrow{R_i}} = \overrightarrow{F_i} = \overrightarrow{F_i^{\rm drv}} + \sum_{j \in \mathcal{N}_i} \overrightarrow{F_{ij}^{\rm rep}} + \sum_{w \in \mathcal{W}_i} \overrightarrow{F_{iw}^{\rm rep}}, \tag{3}$$

where $\overrightarrow{F_{ij}^{\rm rep}}$ denotes the repulsive force from pedestrian j acting on pedestrian i, $\overrightarrow{F_{iw}^{\rm rep}}$ is the repulsive force emerging from the obstacle w and $\overrightarrow{F_i^{\rm drv}}$ is a driving force. m_i is the mass of pedestrian i.

The repulsive forces models the collision-avoidance performed by pedestrians and should guarantee a certain volume exclusion for each pedestrian. The driving force, on the other hand, models the intention of a pedestrian to move to some destination and walk with a certain desired speed. The set of Eq. (3) for all pedestrians results in a high-dimensional system of second order ordinary differential equations. The time evolution of the positions and velocities of all pedestrians is obtained by numerical integration with sufficient accuracy. The choice of integrating method and time step are discussed below.

Below we introduce the forces as defined in the GCFM [10]. The mathematical expression for the driving force is given by

$$\overrightarrow{F_i^{\rm drv}} = m_i \frac{\overrightarrow{v_i^0} - \overrightarrow{v_i}}{\tau}, \tag{4}$$

with a time constant τ.

Given the direction connecting the positions of pedestrians i and j

$$\overrightarrow{R_{ij}} = \overrightarrow{R_j} - \overrightarrow{R_i}, \qquad \overrightarrow{e_{ij}} = \frac{\overrightarrow{R_{ij}}}{R_{ij}} \tag{5}$$

The repulsive force reads

$$\overrightarrow{F_{ij}^{\mathrm{rep}}} = -m_i k_{ij} \frac{(\eta v_i^0 + v_{ij})^2}{\mathrm{dist}_{ij}} \overrightarrow{e_{ij}}, \tag{6}$$

with the distance dist_{ij} between two ellipses which is defined as the distance between the borders of the ellipses, along a line connecting their centers.

This definition of the repulsive force in the GCFM reflects several aspects. First, the force between two pedestrians decreases with increasing distance. In the GCFM it is inversely proportional to their distance R_{ij}. Furthermore, the repulsive force takes into account the relative velocity v_{ij} between pedestrian i and pedestrian j. The following special definition provides that slower pedestrians are not affected by the presence of faster pedestrians in front of them:

$$\begin{aligned} v_{ij} &= \frac{1}{2}[(\overrightarrow{v_i} - \overrightarrow{v_j}) \cdot \overrightarrow{e_{ij}} + |(\overrightarrow{v_i} - \overrightarrow{v_j}) \cdot \overrightarrow{e_{ij}}|] \\ &= \begin{cases} (\overrightarrow{v_i} - \overrightarrow{v_j}) \cdot \overrightarrow{e_{ij}} & \text{if } (\overrightarrow{v_i} - \overrightarrow{v_j}) \cdot \overrightarrow{e_{ij}} > 0 \\ 0 & \text{otherwise}. \end{cases} \end{aligned} \tag{7}$$

As in general pedestrians react only to obstacles and pedestrians that are within their perception, the reaction field of the repulsive force is reduced to the angle of vision (180°) of each pedestrian, by introducing the coefficient

$$k_{ij} = \frac{1}{2} \frac{\overrightarrow{v_i} \cdot \overrightarrow{e_{ij}} + |\overrightarrow{v_i} \cdot \overrightarrow{e_{ij}}|}{v_i} = \begin{cases} (\overrightarrow{v_i} \cdot \overrightarrow{e_{ij}})/v_i & \text{if } \overrightarrow{v_i} \cdot \overrightarrow{e_{ij}} > 0 \quad \& \quad v_i \neq 0 \\ 0 & \text{otherwise}. \end{cases} \tag{8}$$

The coefficient k_{ij} is maximal when pedestrian j is in the direction of movement of pedestrian i and minimal when the angle between j and i is bigger than 90°. Thus the strength of the repulsive force depends on the angle.

3.2. Optimization of the algorithm

The first step in making the computation fast is finding suitable algorithms for solving the equations of motion. There are a number of packages for systems of ODEs, some equipped with automatic control of step size and approximation order. Tests with these solvers gave very unsatisfactory results. This can be traced to the singularity in the repulsive force term. On one hand, in a large dense crowd there are always some persons at close distance, which means large gradients of the repulsive forces, which in turn gives extremely small step sizes. On the other hand, the error estimates used for step size control are overly pessimistic in our case – an error in a step at close encounters will be partially compensated in the next time step, and accuracy requirements in pedestrian dynamics are moderate anyhow. We therefore tried a number of simple ODE solvers – Euler, backward Euler, Velocity-Verlet, Runge–Kutta, leapfrog – with different and constant time steps. With Euler, a time step $\leqslant$0.001 s was needed. With larger time steps, close encounters may give weird and unrealistic movements. Backward Euler is more stable, but the necessary iteration for solving the nonlinear equation of a step makes it inefficient. The most efficient schemes were leapfrog with a time step of 0.01 s and Runge–Kutta with a time step of 0.05 s. Unfortunately this bigger step size is also bounded to more computational power spent into the calculation of the forces.

The calculation of the forces has to be optimized in various ways. A naive implementation of the repulsive force is order O(n^2), with the test $j \in \mathcal{N}_i$ the crucial component. But $\mathcal{N}_i$ is small

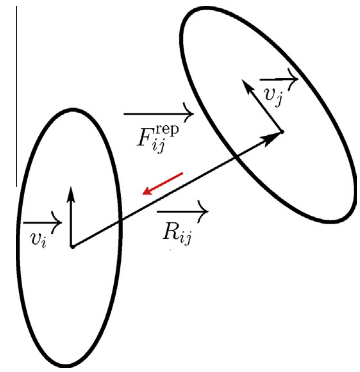

Fig. 3. Direction of the repulsive force. Pedestrians are represented as velocity-dependent ellipses.

for all i, on average it may contain only 10 indices, and slowly varying. With a suitable sorting of people by groups, the test $j \in \mathcal{N}_i$ will have to be done only for the members of a few groups, and the number of tests will be independent of the number of persons in the simulation. For this sorting neighborhood list schemes like the Linked-Cell algorithm [13] or the Verlet-List algorithm [14] are used. Using the linked-cells the neighborhood search complexity can be reduced to O(n) [18]. This reduces the computation times considerably relative to a naive approach, but further tuning is needed. Further the force calculation itself has to be efficient. At present, the distance between ellipses is calculated which needs trigonometric functions, see Fig. 3. For the purpose at hand, using short tables for these functions gives a faster approximation with sufficient accuracy.

A standard topic of optimization is improving the locality of the data. On modern CPUs, arithmetic operations are very much faster than memory access, for this reason cache misses should be reduced. This means that interacting people should be close in numbering, which requires a careful ordering and a reordering in regular intervals. In a 2D arrangement, it is impossible to have all neighbors carrying close-by numbers. A fractal ordering (Hilbert [11] or Morton [12] ordering) has proved efficient in matrix calculations and in molecular dynamics. It was tried here, too, but gave an improvement of less than 1%, which does not justify the added program complexity. The reason for this seems to be on the one hand that the force calculations are quite complex, so the delay by a cache miss is partially hidden by computations, on the other hand a good ordering at the start will for a long time stay fairly good because neighbors will stay close in most cases, so the sophisticated ordering is less needed than in molecular dynamics.

3.3. Parallelization

To achieve the real time requirement, the optimization of the serial code is performed at three levels. The Message Passing Interface (MPI) is used to run the simulation across the computing nodes. The simulation code on each node is run in a multi-threaded environment using OpenMP. Finally special neighborhood list algorithms are used to consider the short range character of the repulsive forces (see previous section). In this case the linked cells is used with a cutoff radius of 2 m.

The parallelization across nodes is implemented using a domain decomposition of the facility that tries to cut the area of simulations into weakly interacting parts (preferably rooms, though this is not always feasible) and exchange boundary data via MPI after every time step. One major issue is the choice of the domain decomposition technique, which has to be at doors where possible or parallel to the main walking direction otherwise to minimize the communication volume. Two main reasons speak in favor of a static domain decomposition on the geometry of the simulated area presented in Fig. 1. The first reason is the relative low communication requirements and also due to the natural partitions given by the persons counting system. This partitioning was done to achieve as little communication as possible between the domains. This is obtained by choosing the boundaries between the domains, which are also the counting lines for the system, as small as possible, thus mainly at doors. We also assume an initial almost uniform distribution of the pedestrians in the simulation area. The ghost areas which hold the data shared by the different nodes are computed with the help of the linked cells. The second reason is the route choice algorithm used by the pedestrians. The algorithm needs information about the pedestrian distribution in a specific decision area (in this case a room) at each simulation step. The route choice algorithm is based on a graph structure. Pedestrians are routed using a quickest path approach based on the observation of the environment presented in [17]. With a static domain decomposition in rooms, all this information is always available on the same processor. In addition, the simulation of pedestrians as particles differs from the simulation of the general N-Body systems. One difference is that in pedestrian dynamics the pedestrian stream and direction of movement are predictable. Predictable in the sense that at a certain time in the simulation, they will gather at exits. Therefore the concept of splitting the simulation area into static areas is well suited. Another reason is the production machine dedicated to the application. 15 nodes are available for the space continuous model in the evacuation assistant. In addition the results of the simulation, i.e. the trajectories of the pedestrians are written with respect to those areas. Thus the nodes can independently perform IO operations without any need of synchronization. In the case of the evacuation assistant, the initial number of pedestrians is received via sensors of an automatic persons counting system. The received pedestrians are then homogeneously distributed in their respective areas. All processes on the nodes read the same geometry file and logically have the same constraints on that geometry. The constraints are for example the doors states (closed/open) and the block states (smoked/smoke-free). Thereafter the master shares the works, depending on the number of computing nodes available. Each node receives $\left\lfloor \frac{\#sections}{\#nodes} \right\rfloor$. The last node in the list might receive more. When a node receives more than one section to operate on, it is guaranteed that the sections are contiguous. This also limits the data exchange between the processes.

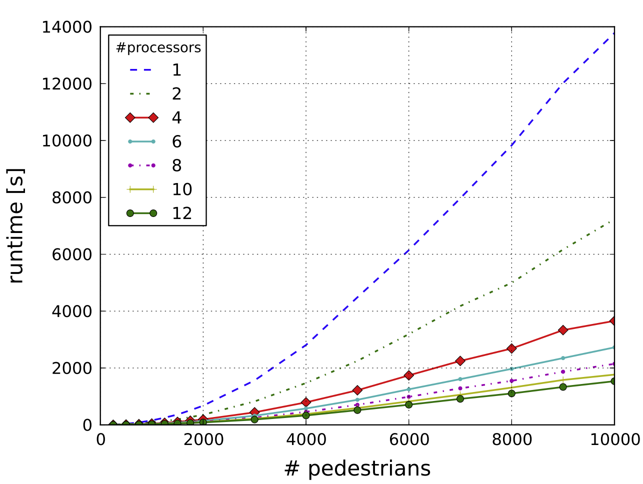

Fig. 4. Runtime using different numbers of processors with the shared memory OpenMP parallelization method.

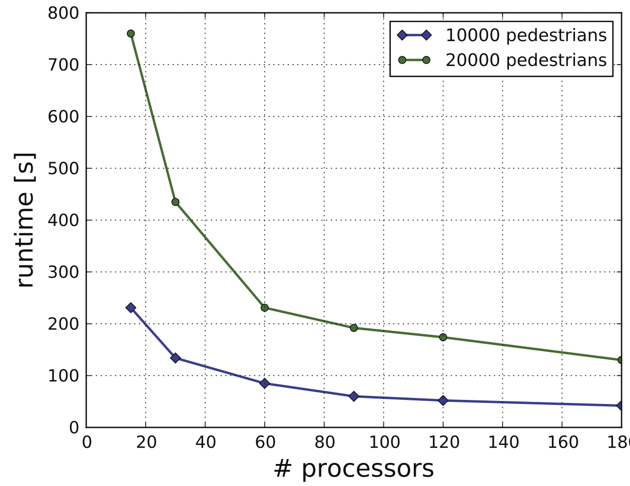

Fig. 5. Runtime using different numbers of processors. The evacuation times are 332 s for 10,000 pedestrians and 580 s for 20,000 pedestrians.

The runtimes using different numbers of processors with OpenMP and the linked cells are given in Fig. 4. The OpenMP parallelization achieves a maximum speedup of 9 over the single core routine for 10,000 pedestrians distributed in a single room and computed on up to 12 processors.

The runtimes for 10,000 and 20,000 pedestrians obtained by combining OpenMP, MPI and the linked cells are shown in Fig. 5. The evacuation times and therefore the simulation runtime are strongly coupled to the route choice strategy used. Much of the optimization of the program is done using the JUROPA computer of the JSC which has an older Version of Nehalem nodes with similar architecture, just only eight cores per node. A strong advantage of using the JUROPA computer is the availability of the leading edge performance analysis tools of the Scalasca toolset [15], while the cluster for the Hermes project has only the software required for production runs.

4. Processing the results

The results of the simulation are moved to the front-end computer, where they have to be presented in a form that allows decision making within a few minutes. This excludes the standard method of visualizing the entire simulation and playing it to the audience. What is needed is a condensed representation of the results which allows identification of critical situations in short time. There are two options planned for this representation, both depending on the calculation of local densities [16]. One is a high speed movie of the densities in the facility, the other is a static representation showing the maximal short time average of the density by color coding. Tests will show which one is more instructive to the safety management. When critical regions in space and time are identified, which will (hopefully) encompass only a small area and limited time, zooming in will be needed to decide on actions. For this, different visualizations can be activated interactively. One is a visualization running moderately faster than real time, others will be displays of passing times and densities for the critical region, or other kinds of display. These (or similar) visualizations are already in use in facility planning systems [19], and details will be decided upon during the testing.

5. Conclusion

Using an integrated system which incorporates the information from person counting, door sensors, smoke detectors, and possibly safety staff to feed an evacuation simulation running on a cluster system it is possible to get predictions of the performance of an evacuation fast enough to allow safety operations to take measures against critical situations a few minutes before those happen. How much difference this will make for the safety of events will become visible when the test system is actually set up.

6. Additional material

Further information including experiments and various dataset for benchmarking will be made available at the following permanent addresses:

- http://www.fz-juelich.de/ias/jsc/cst/.
- http://www.asim.uni-wuppertal.de/.

Acknowledgements

The Hermes project is funded by the German Federal Ministry of Education and Research (BMBF), the computation time on JUROPA is provided by the John von Neumann-Institut for Computing (NIC).

References

[1] Kemloh Wagoum AU, Chraibi M, Mehlich J, Seyfried A, Schadschneider A. Efficient and validated simulation of crowds for an evacuation assistant. Comput Animat Virt Worlds 2012;23(1):3–15.

[2] Rogsch C, Klingsch W, Seyfried A. Prediction accuracy of evacuation times for high-rise buildings and simple geometries by using different software-tools. In: Traffic and Granular Flow '07. Berlin Heidelberg: Springer; 2009. p. 395–400.

[3] Blue V, Adler J. Cellular automata microsimulation for modeling bidirectional pedestrian walkways. Transport Res Part B 2001;35:293–312.

[4] Kirchner A, Schadschneider A. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. Physica A 2002;312:260–76.

[5] Thompson PA. Developing new techniques for modelling crowd movement. PhD Thesis, University of Edinburgh; 1994.

[6] Galea ER. BuildingEXODUS V4.0 User guide and technical manual; 2004.

[7] Helbing D, Molnar P. Social force model for pedestrian dynamics. Phys Rev E 1995;51:4282–6.

[8] Yu WJ, Chen R, Dong LY, Dai SQ. Centrifugal force model for pedestrian dynamics. Phys Rev E 2005;72/2:026112.

[9] Chraibi M, Seyfried A, Schadschneider A, Mackens W. Quantitative verification of a force-based model for pedestrian dynamics. Traffic Granul Flow 2009.

[10] Chraibi M, Seyfried A, Schadschneider A. Generalized centrifugal-force model for pedestrian dynamics. Phys Rev E 2010;82:046111.

[11] Hilbert D. Über die stetige Abbildung einer Linie auf ein Flächenstück. Math Ann 1891;38:459–60.

[12] Morton GM. A computer oriented geodetic data base and a new technique in file sequencing. Technical report, Ottawa, Canada: IBM Ltd.; 1966.

[13] Griebel M, Knapek S, Zumbusch G. Numerical simulation in molecular dynamics: numerics, algorithms, parallelization, applications. 1st ed. Springer Publishing Company, Incorporated; 2007.

[14] Verlet L. Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. Phys Rev 1967;159(1):98103.

[15] Geimer M et al. The Scalasca performance toolset architecture. Concurr Comput: Pract Exper 2010;22(6):702–19.

[16] Steffen B, Seyfried A. Reliability issues in the microscopic modeling of pedestrian movement. Physica A 2010;389:1902–10.

[17] Kemloh Wagoum AU, Seyfried A, Holl S. Modelling dynamic route choice of pedestrians to assess the criticality of building evacuation. Adv Complex Sys 2012;15(3).

[18] Seyfried A, Chraibi M, Kemloh U, Mehlich J, Schadschneider A. Runtime optimization of force based models within the Hermes project. In: Pedestrian and Evacuation Dynamics 2010. US: Springer; 2011. p. 363373.

[19] http://www.ptvag.com/software/transportation-planning-traffic-engineering/software-system-solutions/vissim/pedestrian-simulation/.