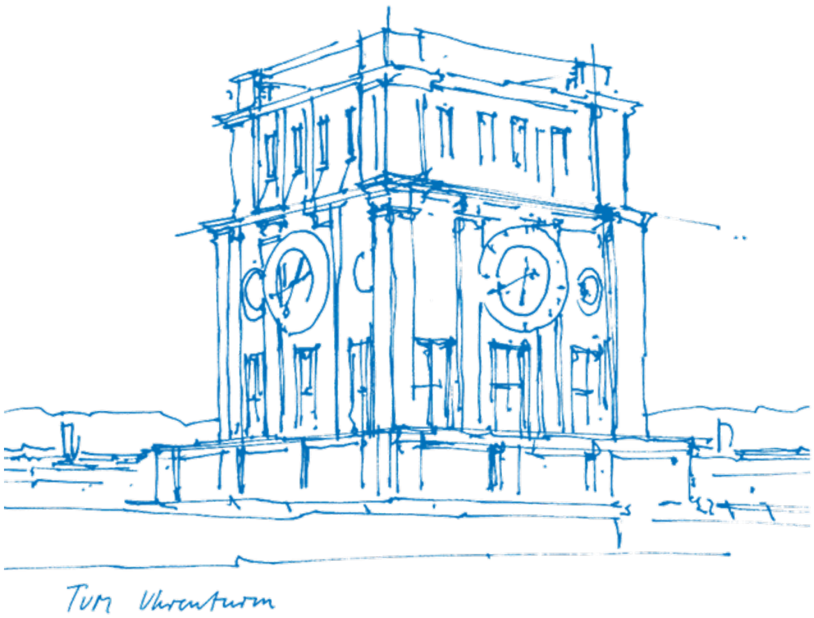
Machine Learning in Crowd Modeling & Simulation
Lecture 1 - Modeling crowd dynamics

Felix Dietrich



Outline

Modeling crowds

- General overview, state of the art, relation to machine learning
- Modeling approaches
- In detail: Cellular automata

Machine Learning in Crowd Modeling & Simulation

Motivation - Modeling and Simulation

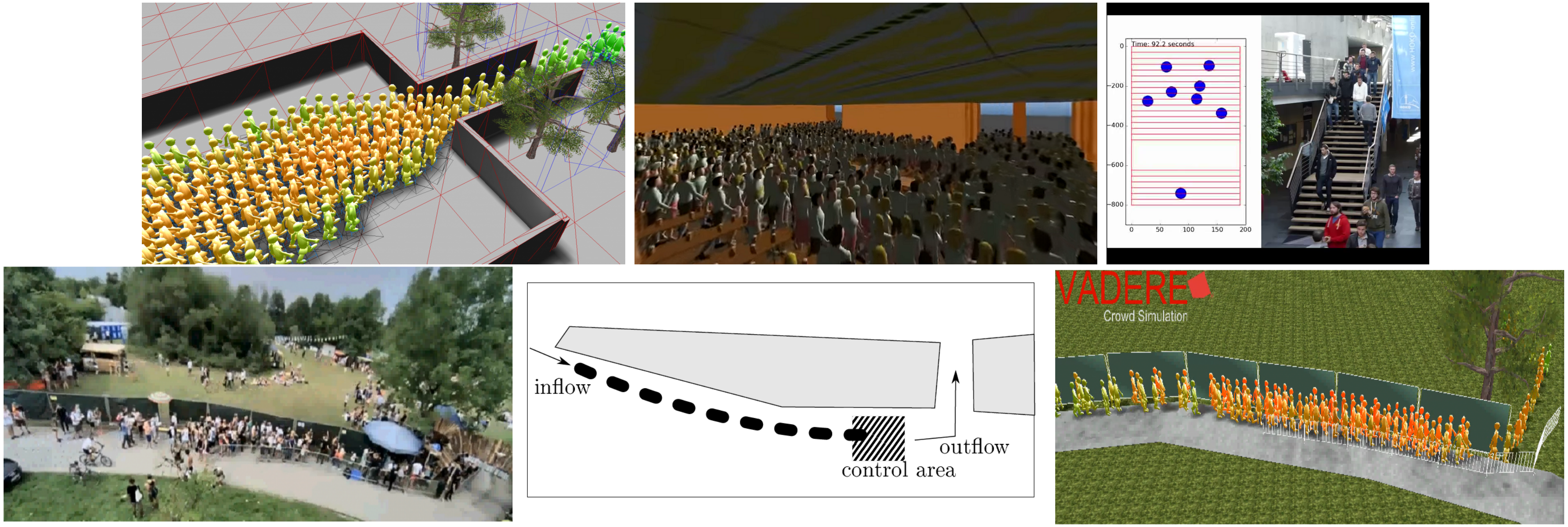


Figure: Left: Simulated pedestrians (colored by density) evacuate a room through a bottleneck. Center: Simulation and 3D visualization of an evacuation of a beer tent. Right: Experiment with students to evaluate dynamics on stairs. Bottom: Comparison of real video footage from an entrance to a music festival (left) with a simulation (right). The scenario setup of the simulation is shown in the center.

Machine Learning in Crowd Modeling & Simulation

Motivation - Complex systems

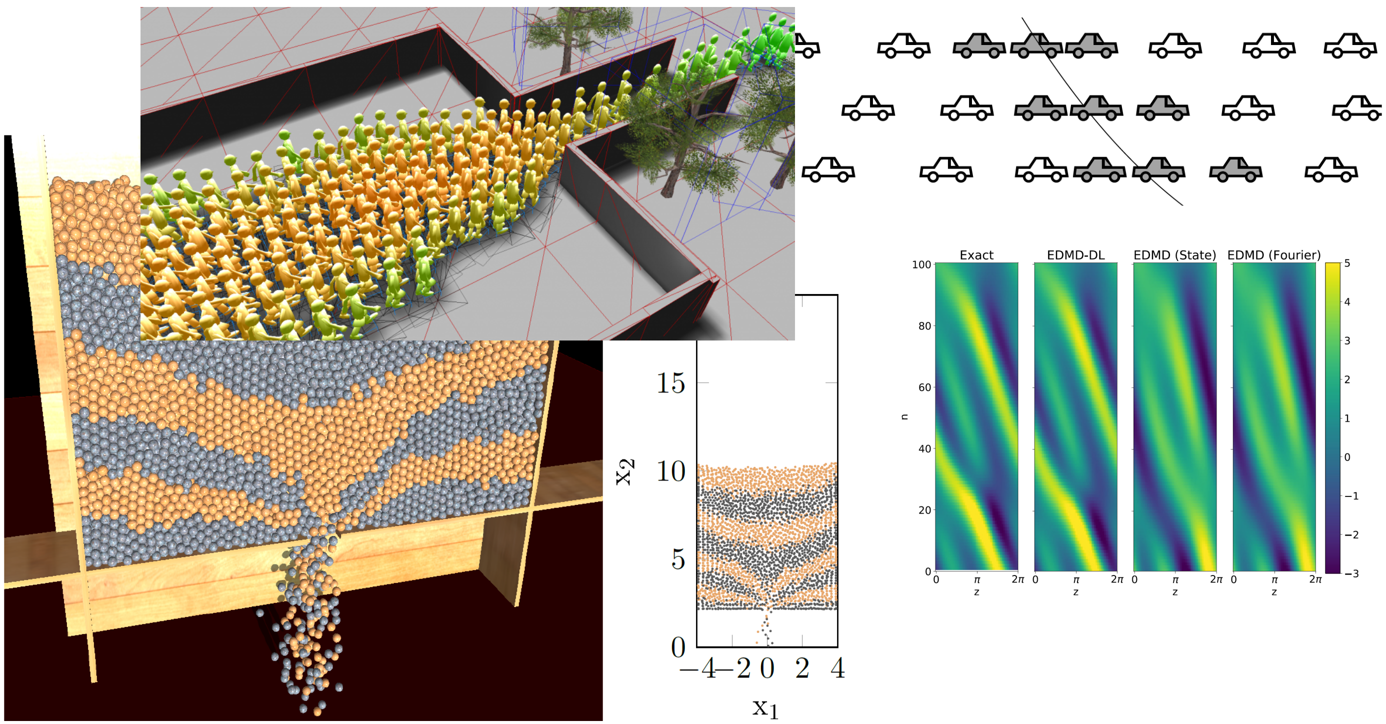


Figure: Applications with complex systems: Crowd modeling and simulation, car traffic, granular flow, combustion / fluid flow, ...

Machine Learning in Crowd Modeling & Simulation

Motivation - Machine Learning

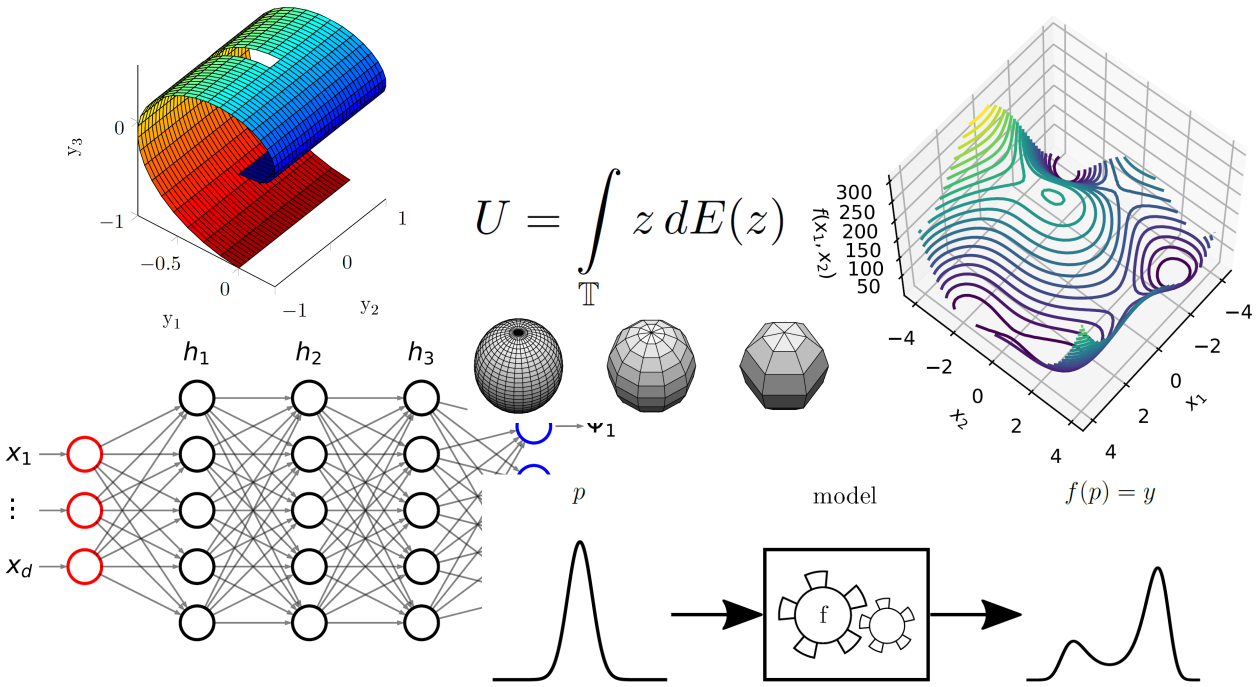
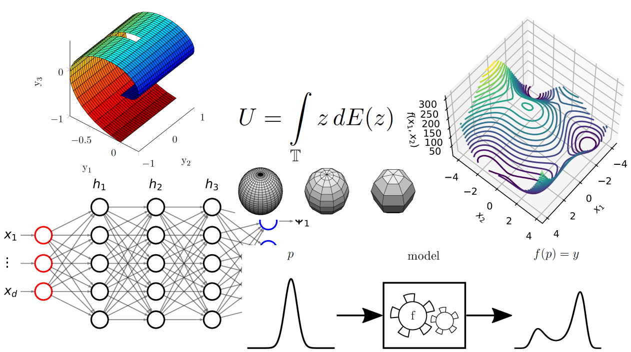


Figure: Zoo of methods in machine learning: Manifold learning, operator theory, model order reduction, optimization, function representation, uncertainty quantification, ...

Machine Learning in Crowd Modeling & Simulation

Motivation - Machine Learning for Complex Systems

Goal of this course: bridge the gap!



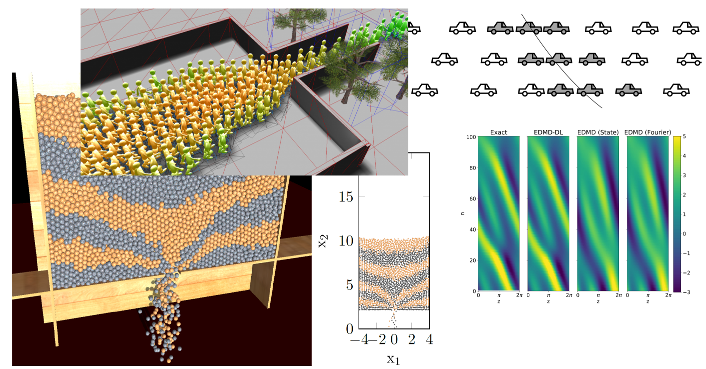


Figure: Left: Zoo of methods in machine learning. Right: Applications with complex systems.

Modeling crowds

General overview: objectives

Figure: Two-dimensional visualization of a crowd simulation result in Vadere [1], three-dimensional VR visualization of a beer tent evacuation [2], and experiment with real humans in red/black shirts (Hermes project [3]).

Different objectives:

- Engineer: does it work?
- Scientist: is it (close to) the truth?
- Game developer: does it look convincing?

---

1 `https://www.vadere.org`

2 `https://www.youtube.com/watch?v=5UxGIsptL5g`

3 `https://www.fz-juelich.de/ias/ias-7/EN/AboutUs/Projects/Hermes/_node.html`

Modeling crowds

General overview: publications

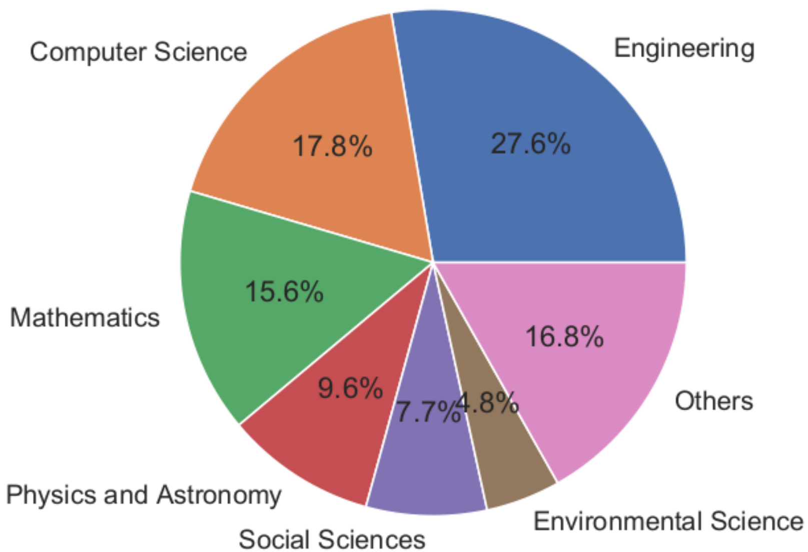
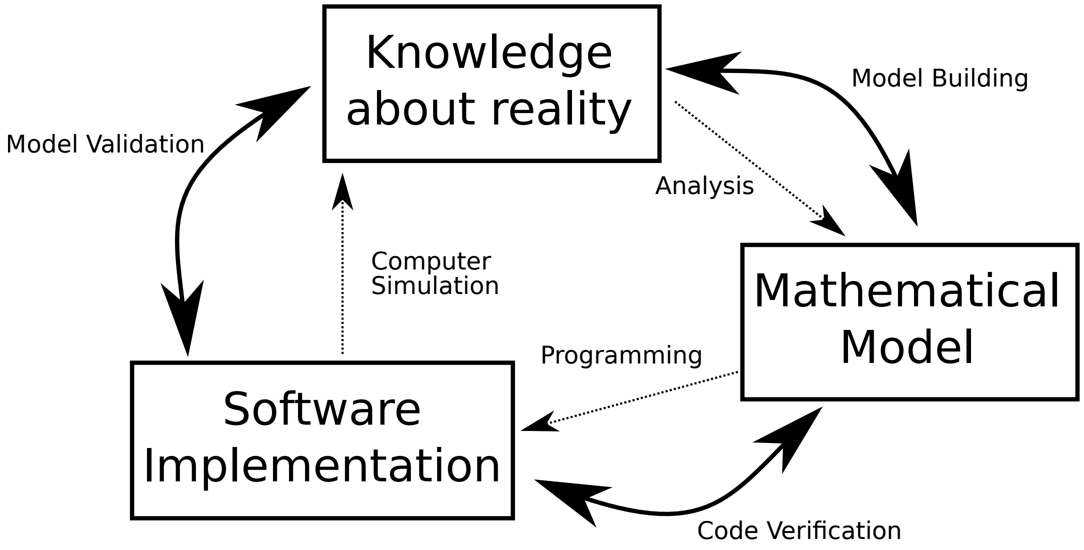


Figure: Scopus search result for the term “pedestrian dynamics” on 2019-06-05. Overall, 2422 documents are listed, see [Kleinmeier et al., 2019] for details.

Modeling crowds

General overview: verification and validation

- Validation: did we build the right system?
- Verification: did we build the system right?



Modeling crowds

Different approaches

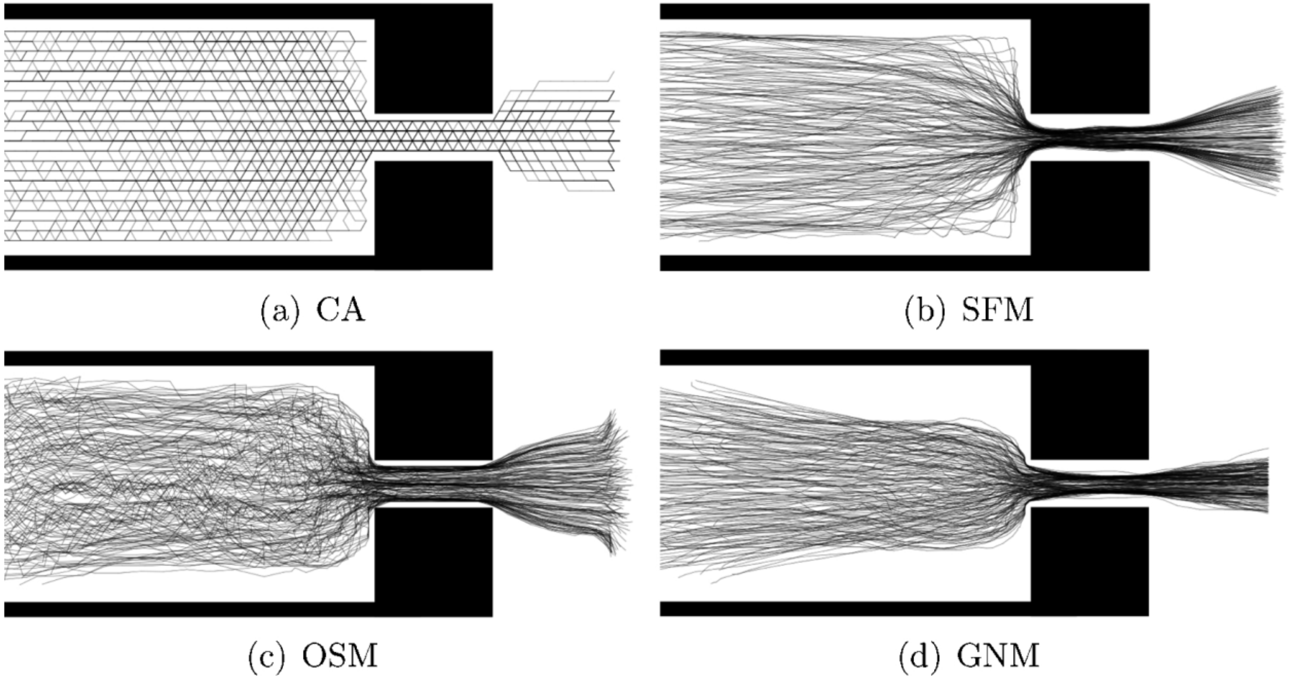


Figure: Trajectories of pedestrians obtained by simulating four different models: (a) Cellular Automaton, (b) Social Force Model, (c) Optimal Steps Model, and (d) Gradient Navigation Model. From: [Dietrich et al., 2014]

Modeling crowds

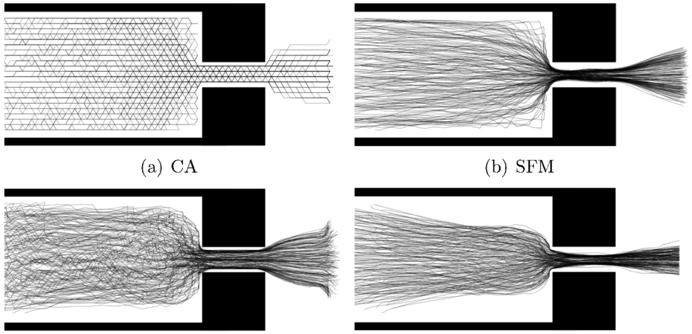
Different approaches

Vadere video.

Modeling crowds

Different approaches

- Reviews and overview: [Dietrich et al., 2014, Seitz et al., 2016b]
- Force based models: [Helbing and Molnár, 1995, Chraibi et al., 2010, Chraibi et al., 2011]
- Velocity based models: [Dietrich and Köster, 2014, Tordeux and Seyfried, 2014]
- Cellular automata: [Gipps and Marksjö, 1985, Blue and Adler, 2001, Burstedde et al., 2001, Kirik et al., 2009, Davidich and Köster, 2013, Was and Lubaś, 2013, Fu et al., 2015]
- Discrete models: [Seitz and Köster, 2012, von Sivers and Köster, 2015]
- Heuristics, decision models, and psychology: [Moussaïd et al., 2011, Seitz et al., 2015, Seitz et al., 2016a, Seitz et al., 2016c, von Sivers et al., 2016]



(a) CA (b) SFM (c) OSM (d) GNM

Modeling crowds

Cellular automaton: general idea

A cellular automaton is defined through the following [Boccara, 2010]:

1. A state space of cells, each with their own possible states.
2. An update rule (or *evolution operator*) to update all cells to the next state.

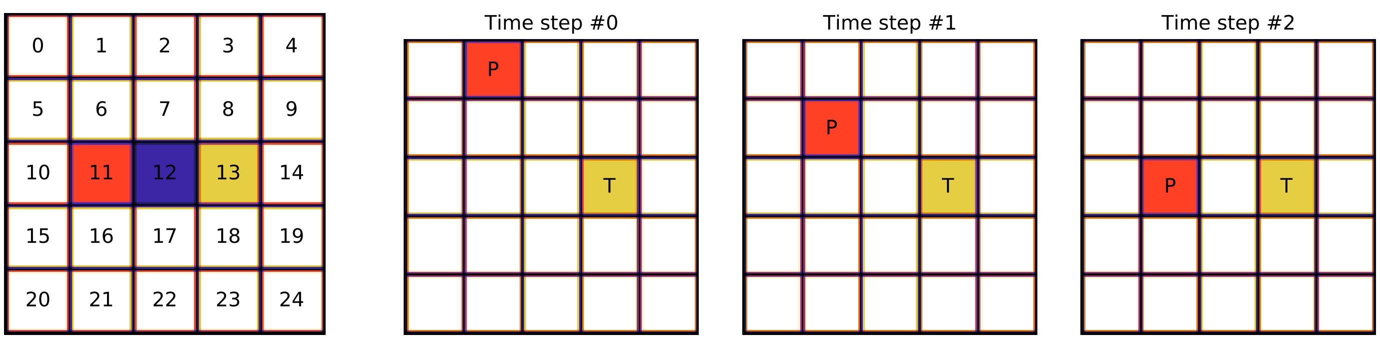


Figure: Left: The state space of a cellular automaton with 25 cells. Right: An update rule, applied two times.

Modeling crowds

Cellular automaton: general idea

In a cellular automaton, the full state of the system is contained in the states of individual cells. The automaton is updated with an update rule, to generate the next state.

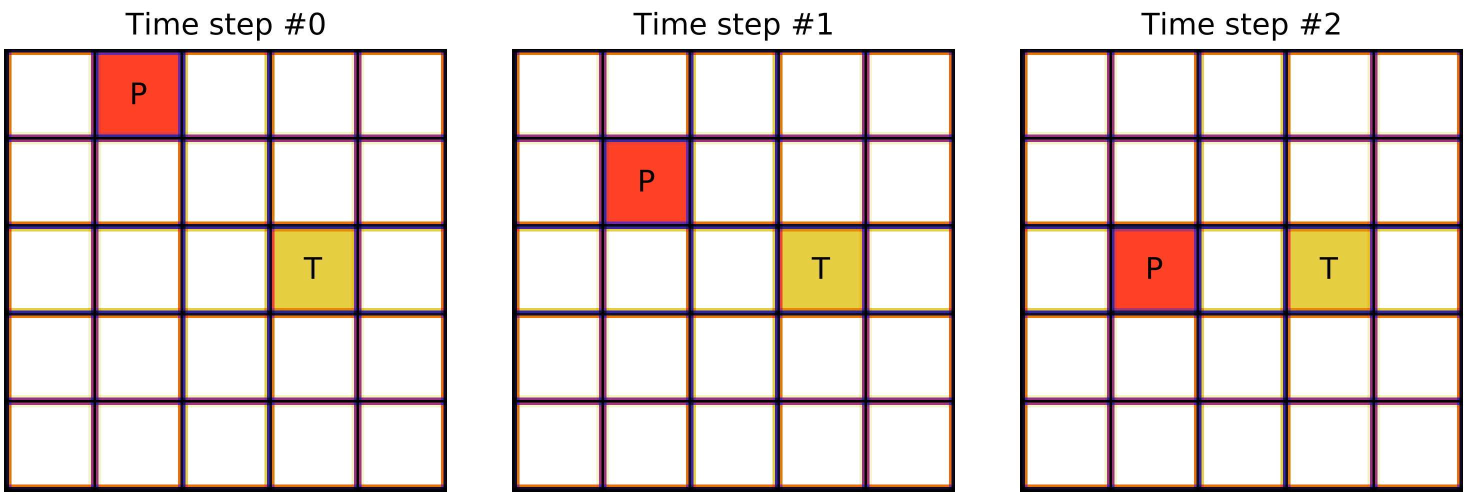


Figure: General idea of a cellular automaton. The state is contained in a collection of cells, and is updated with an update rule. In this example, the state P indicates a pedestrian, T a target.

Modeling crowds

Cellular automaton / state space (1/3)

In a cellular automaton, the full state of the system is contained in the states of individual cells. In a crowd simulation, these cells are typically arranged in a two-dimensional grid.

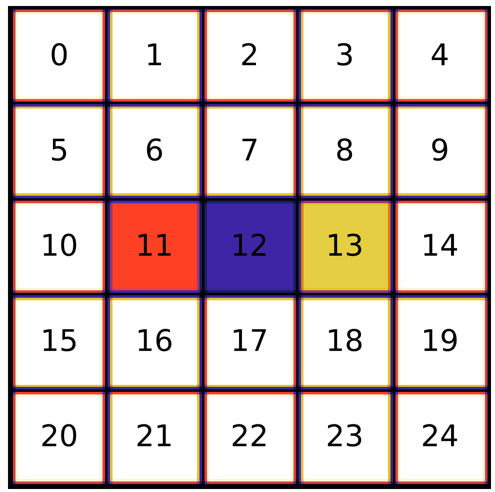


Figure: The state space of a cellular automata with 25 cells. Three cells are marked in different color. The number is the index of the cell, without reference to row or column number.

Modeling crowds

Cellular automaton / state space (2/3)

In cellular automata we will use, the full state of the system is contained in the states of individual cells. In a crowd simulation, these cells are typically arranged in a two-dimensional grid. A possible state space X_i for a single cell i may be

$$X_i := \{E, P, O, T\},$$

where the three possible symbols for a state are interpretable as

1. E: empty cell,
2. P: there is a pedestrian in this cell,
3. O: there is an obstacle in this cell,
4. T: this cell is a target for the pedestrians in the scenario.

If we arrange the cells of the cellular automata in a grid, the state space of the complete system would be $X = \{E, P, O, T\}^{5\times 5}$, i.e. 25 different cells with E, P, O, or T as their current state.

Modeling crowds

Cellular automaton / state space (3/3)

If we arrange the cells of the cellular automata in a grid, the state space of the complete system would be $X = \{E, P, O, T\}^{5\times 5}$, i.e. 25 different cells with E, P, O, or T as their current state.

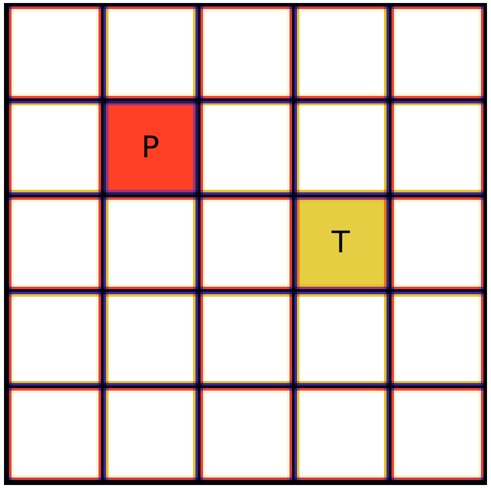


Figure: The state space of a cellular automata, a pedestrian at $(2, 2)$ and a target at $(4, 3)$.

Modeling crowds

Cellular automaton / update scheme

The simulation of a “crowd”, here, defined as “all pedestrians in the scenario”, can be done in many different ways. For cellular automata, the concept of an update scheme is important. You can read more about update schemes in the literature [Seitz and Köster, 2014], but for this exercise, a simple, discrete-time update scheme with constant time shifts suffices. This update scheme for the cellular automaton can be defined as follows:

1. Let $x^{(n)} \in X$ be the state of the system (the automaton) at time step n.
2. Define $x^{(n+1)} = f(x^{(n)})$ as the next state of the system, with the time step $n+1$ a fixed $\Delta t \in \mathbb{R}$ after time step n and $f : X \to X$ a map between system states (the “evolution operator”).
3. Reset the system state to $x^{(n+1)}$, advance n by one, and continue with step (1).

Modeling crowds

Cellular automaton / evolution operator

The evolution operator f of the cellular automaton can use all the information currently available to advance the system to the next time step. For a scenario with only one cell being in state P (one pedestrian in the scenario), and one cell being in state T (one target), the evolution operator may act like this:

1. For each cell in state P, collect the neighboring cells N_P that are in state E.
2. If any cell N_P is in the state T, return the current state unchanged.
3. Else, compute the distance from all neighbors in N_P to the cell in state T through

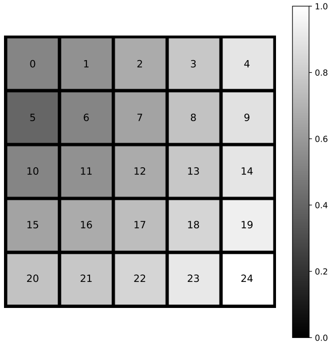
   $$d(c_{ij}, c_{kl}) = \sqrt{(i-k)^2 + (j-l)^2},$$

   that is, the Euclidean distance between the cell indices ij and kl, where i and k are the row indices, and j and l are the column indices of the cells.
4. If any cell in N_P is closer to the target than the current cell, set the state of the cell in state P to the state E, and set the cell with the smallest distance to the target cell from state E to state P. Return this new state space.

Modeling crowds

Cellular automaton / utility functions (1/2)

A more sophisticated (but also more useful) way to update the state is the use of a utility function $u : I \times X \to \mathbb{R}$. This function takes a cell index in the index set I as well as the current state of the cellular automaton, and results in a (real valued) utility at the given index. The evolution operator f then only needs to check the neighboring cells of a given pedestrian cell for the value of u, and move the pedestrian to the cell with the highest utility (which may also be the current cell, i.e. the pedestrian does not move at all). An example for a utility function (or rather, a cost function!) is the distance to the closest target cell.



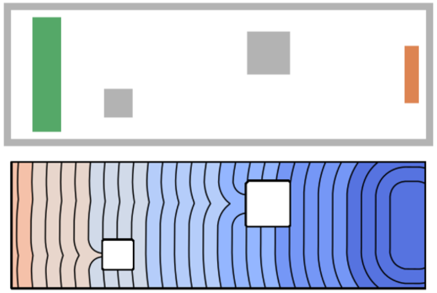


Figure: Left: A utility (here: negative utility, or cost) function modeling the distance to the target (cell 5). Right: More complex scenario (top, from [Kleinmeier et al., 2019]) with starting area, two obstacles, and target (individual cells not shown). The utility function is shown on the bottom, with red lowest and blue highest utility.

Modeling crowds

Cellular automaton / utility functions (2/2)

Interactions of individuals with others or obstacles in the environment is typically modelled through utility or cost functions that depend on the distance to other pedestrians in addition to the distance to the target. The left panel in the figure shows the following cost function for the interaction between two individuals, which can simply be added to a cost function for the target to obtain simple avoidance behavior:

$$c(r) = \begin{cases} \exp\left(\frac{1}{r^2 - r_{\max}^2}\right) & \text{if } r < r_{\max} \\ 0 & \text{else} \end{cases}$$

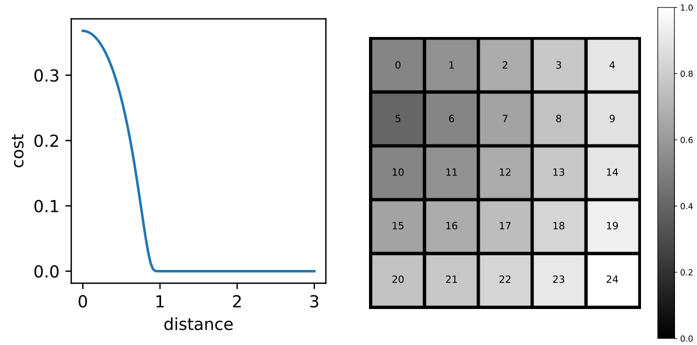


Figure: Left: A typical cost function $c(r)$ modeling the pedestrian-pedestrian or pedestrian-obstacle interaction at distance r. Right: A cost function modeling the distance to the target (cell 5).

Modeling crowds

Summary

1. General overview of crowd modeling
2. Objectives of stakeholders
3. Verification and validation of models
4. Different crowd modeling approaches
5. Cellular automata

Literature I

Blue, V. J. and Adler, J. L. (2001).
Cellular automata microsimulation for modeling bi-directional pedestrian walkways.
Transportation Research Part B: Methodological, 35:293–312.

Boccara, N. (2010).
Modeling Complex Systems.
Graduate Texts in Physics. Springer, New York, 2nd edition.

Burstedde, C., Klauck, K., Schadschneider, A., and Zittartz, J. (2001).
Simulation of pedestrian dynamics using a two-dimensional cellular automaton.
Physica A: Statistical Mechanics and its Applications, 295:507–525.

Chraibi, M., Kemloh, U., Schadschneider, A., and Seyfried, A. (2011).
Force-based models of pedestrian dynamics.
Networks and Heterogeneous Media, 6(3):425–442.

Chraibi, M., Seyfried, A., and Schadschneider, A. (2010).
Generalized centrifugal-force model for pedestrian dynamics.
Physical Review E, 82(4):046111.

Davidich, M. and Köster, G. (2013).
Predicting pedestrian flow: A methodology and a proof of concept based on real-life data.
PLoS ONE, 8(12):1–11.

Dietrich, F. and Köster, G. (2014).
Gradient navigation model for pedestrian dynamics.
Physical Review E, 89(6):062801.

Dietrich, F., Köster, G., Seitz, M., and von Sivers, I. (2014).
Bridging the gap: From cellular automata to differential equation models for pedestrian dynamics.
Journal of Computational Science, 5(5):841–846.

Fu, Z., Zhou, X., Chen, Y., Gong, J., Peng, F., Yan, Z., Zhang, T., and Yang, L. (2015).
The influence of random slowdown process and lock-step effect on the fundamental diagram of the nonlinear pedestrian dynamics: An estimating-correction cellular automaton.
Communications in Nonlinear Science and Numerical Simulation, 20(3):832–845.

Gipps, P. and Marksjö, B. (1985).
A micro-simulation model for pedestrian flows.
Mathematics and Computers in Simulation, 27(2–3):95–105.

Helbing, D. and Molnár, P. (1995).
Social Force Model for pedestrian dynamics.
Physical Review E, 51(5):4282–4286.

Kirik, E. S., Yurgelyan, T. B., and Krouglov, D. V. (2009).
The shortest time and/or the shortest path strategies in a CA FF pedestrian dynamics model.
Mathematics and Physics, 2(3):271–278.

Literature II

- Kleinmeier, B., Zönnchen, B., Gödel, M., and Köster, G. (2019).
  Vadere: An open-source simulation framework to promote interdisciplinary understanding.
  arXiv.
- Moussaïd, M., Helbing, D., and Theraulaz, G. (2011).
  How simple rules determine pedestrian behavior and crowd disasters.
  Proceedings of the National Academy of Sciences, 108(17):6884–6888.
- Seitz, M. J., Bode, N. W. F., and Köster, G. (2016a).
  How cognitive heuristics can explain social interactions in spatial movement.
  Journal of the Royal Society Interface, 13(121):20160439.
- Seitz, M. J., Dietrich, F., and Köster, G. (2015).
  The effect of stepping on pedestrian trajectories.
  Physica A: Statistical Mechanics and its Applications, 421:594–604.
- Seitz, M. J., Dietrich, F., Köster, G., and Bungartz, H.-J. (2016b).
  The superposition principle: A conceptual perspective on pedestrian stream simulations.
  Collective Dynamics, 1:A2.
- Seitz, M. J. and Köster, G. (2012).
  Natural discretization of pedestrian movement in continuous space.
  Physical Review E, 86(4):046108.
- Seitz, M. J. and Köster, G. (2014).
  How update schemes influence crowd simulations.
  Journal of Statistical Mechanics: Theory and Experiment, 2014(7):P07002.
- Seitz, M. J., Templeton, A., Drury, J., Köster, G., and Philippides, A. (2016c).
  Parsimony versus reductionism: How can crowd psychology be introduced into computer simulation?
  Review of General Psychology.
- Tordeux, A. and Seyfried, A. (2014).
  Collision-free nonuniform dynamics within continuous optimal velocity models.
  Physical Review E, 90:042812.
- von Sivers, I. and Köster, G. (2015).
  Dynamic stride length adaptation according to utility and personal space.
  Transportation Research Part B: Methodological, 74:104 – 117.
- von Sivers, I., Templeton, A., Künzner, F., Köster, G., Drury, J., Philippides, A., Neckel, T., and Bungartz, H.-J. (2016).
  Modelling social identification and helping in evacuation simulation.
  Safety Science, 89:288–300.
- Was, J. and Lubaś, R. (2013).
  Adapting social distances model for mass evacuation simulation.
  Journal of Cellular Automata, 8:395–405.
  Journal of Cellular Automata, Old City Publishing.