Crowd Formal Modelling and Simulation: The Sa'yee Ritual

Ilias Sakellariou*, Omar Kurdi†, Marian Gheorghe†, Daniela Romano†, Petros Kefalas‡,
Florentin Ipate§ and Ionut Niculescu§
*Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece
Email:iliass@uom.gr
† Department of Computer Science, University of Sheffield, Sheffield, UK
Email: acp11oak@sheffield.ac.uk, m.gheorghe@sheffield.ac.uk, d.romano@sheffield.ac.uk
‡ Department of Computer Science, The University of Sheffield International Faculty,
CITY College, Thessaloniki, Greece
Email:kefalas@city.academic.gr
§ Department of Computer Science, University of Bucharest, Bucharest, Romania
Email: florentin.ipate@ifsoft.ro, ionutmihainiculescu@gmail.com

***Abstract*—There is an increasing interest in modelling of agents interacting as crowd and a simulation of such scenarios that map to real-life situations. This paper presents a generic state-based abstract model for crowd behaviour that can be mapped onto different agent-based systems. In particular, the abstract model is mapped into the simulation framework NetLogo. We have used the model to simulate a real-life case study of high density diverse crowd such as the Hajj ritual at the mosque in Mecca (Makkah). The computational model is based on real data extracted from videos of the ritual. We also present a methodology for extracting significant data, parameters, and patterns of behaviour from real-world videos that has been used as an early stage validation to demonstrate that the obtained simulations are realistic.**

I. Introduction

The interest in modelling complex systems involving numerous individuals interacting in various ways is of increasing importance in many real-life situations, like passengers in an airport, patients in a hospital unit, pilgrims visiting a holy place, but also in safety engineering, digital entertainment and other areas. Numerous accidents involving large crowds have been recorded, and many overcrowded places are being monitored continuously. Many psychologists, sociologists, physicists and computer scientists aim to understand, control and predict the behaviour of such crowds.

These systems are in general highly complex and thus mathematical or analytic models are neither intuitive nor adequate. Agent-based approaches represent an alternative way to model, simulate and analyse crowd behaviour. In connections with various problems, multi-agent systems are widely employed with different modelling paradigms and implemented by using different software and hardware platforms. A comprehensive survey of various aspects regarding the use of the agent-based methods in modelling such systems and the technologies utilised is provided in [1]. The survey identifies three behavioural factors, namely physical, sociological and psychological. *Physical factors* refer to characteristics such as position, speed, appearance, gesture etc. *Social factors* include culture, social norms, leadership, etc. *Psychological factors* include aspects such as emotions, personality traits, appraisal etc.

The first set of factors is the most utilised in agent-based approaches and modelled in various ways. Cellular automata was among the primary models used referring to spatial behaviour [1]. Later on, several other models were introduced among which the extended state machines, such as *X-machine*. These were used to model biologically inspired systems, for instance foraging processes in bees and ant colonies, [2] but have been also extended to deal with psychological factors. for instance modelling the effect of emotions in emergency evacuation cases [3].

The X-machine model has been intensively studied in connection with agent-based models in two directions. Firstly, the state model has been considered as a general abstract specification language for the NetLogo framework [4], [5]. Secondly, the FLAME framework was developed [6] which in its latest stages, it has been upgraded to work on high-performance hardware platforms [7].

The *main contribution* of our paper is a coherent and integrative approach for the specification, simulation and analysis of crowd behaviour, consisting of: (i) a general modelling framework based on X-machines; (ii) flexible and robust agent-based simulation environments implementing the model derived from the X-machine specification; (iii) a rigorous method of extracting data, parameters and patterns of behaviour from videos; and (iv) some validation and analysis procedures for the simulation results. This approach is illustrated for a real-life case, the Hajj, the pilgrimage to Mecca (Makkah), which shows the behaviour of people in an overcrowded area. We have selected a certain part of the pilgrimage, which refers to the crowd behaviour within a corridor where pilgrims move back and forth following a certain religious ritual, called Sa'yee. The simulation is performed using a NetLogo [8] implementation obtained from the generic X-machine model. We have chosen NetLogo as it provides an easy way to rapidly prototype a system and realise a graphical interface

978–1–4799–5538–1/14/$31.00 ©

for conducting experiments.

This paper is structured as follows: the case study describing the behaviour of the pilgrims performing the Sa'yee ritual is presented in Section II; the X-machine formal model and its usage to specify the case study appear in Section III; in Section IV the NetLogo implementation of the case study that uses the X-machine model specification is described; Section V briefly presents the methodology for extracting data, parameters and patterns of behaviour from videos; the experiments performed, results collected and an initial validation approach are presented in Section VI; finally, conclusions are drawn in Section VII.

II. CASE STUDY: THE SA'YEE RITUAL

To demonstrate the generality of our agent-based-approach we have considered the problem of the Hajj, the religious pilgrimage to Mecca (Makkah), in which a high density crowd of people from all over the world comes to follow a ritual. The whole Hajj requires the pilgrims to perform several steps. At each step a specific activity has to be accomplished. For the purpose of this work we have focused on the *Sa'yee ritual*. This takes place within a corridor linking two hills: the *Safa* (or *Al-Safa*) and the *Marwah* (or *Al-Marwah*). Each pilgrim is required to move between the two hills seven times starting from Safa and ending at Marwah. The corridor is 395.5 metres long, of which the main straight area is about 300 metres, while at each end approximately 45 metres are used for turning. The straight part of the corridor has different lanes: for non-disabled and disabled people. The total width of the corridor is 40 metres of which lanes of 17 metres width are for non-disabled and lanes of 2.5 meters are dedicated for disabled people and their accompanying groups. In addition some non-disabled people use the lane if there are not many disabled people in the corridor in order to offload the numbers in the other lanes. The two priority lanes for disabled people, one per each direction, are situated in the most internal part of the corridor. At each end the pilgrims need to change the lane in the turning area. The pilgrims walking in each lane can be either single individuals or can come in groups. They move mostly in a straight line, unless they need to overtake others. There are a few obstacles, like pillars on both sides of each lane and at the end of the corridor, and the pedestrians need to avoid them. A floor map of the corridor is presented in Figure 1.

Crowd control is of outmost importance in the case of Sa'yee ritual. Failure to control this phenomenon might cause several incidents and the loss of lives. According to [9], *"crowd control has become a major problem with many pilgrims dying in accidents and stampedes in congested places during the Hajj"*. In order to improve emergency management of such situations it is crucial to understand their causes and analyse various alternative scenarios. Rigorous simulation approaches can help significantly in this respect showing the area of potential overcrowding.

As such it is important to demonstrate that simulations created are able to reproduce reality, and that simulation results are validated against the real data obtained for example from videos.

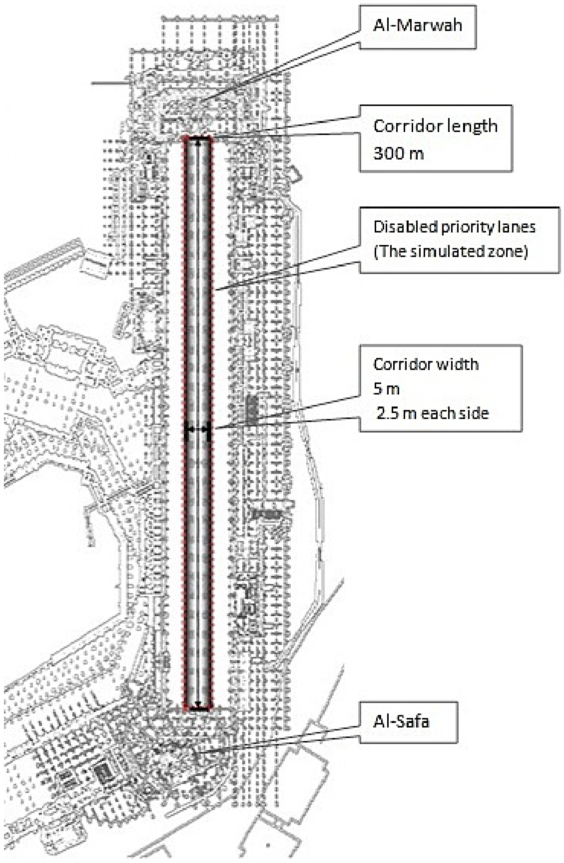


Fig. 1. A floor map of the corridor on the Sa'yee ritual

In the following sections we present the modelling of agents in the above scenario, the simulation as well as the methodology of extracting the relevant information needed to build the simulation model.

III. X-MACHINE MODEL

This work employs X-Machines to model agent behaviour. X-Machines are state machines augmented with a *memory structure* and *functions* that guard state transitions and have a strong legacy to both theory and practice [10].

A *stream X-Machine* [10] is an 8-tuple

$$\mathcal{X} = (\Sigma,\, \Gamma,\, Q,\, M,\, \Phi,\, F,\, q_0,\, m_0)$$

where:

- Σ and Γ are the *input* and *output* alphabets, respectively.
- Q is the finite set of *states*.
- M is the (possibly) infinite set called *memory*.
- Φ is a set of *partial functions* φ; each such function maps an input and a memory value to an output and a possibly different memory value, $\varphi : \Sigma \times M \rightarrow \Gamma \times M$.
- F is the *next state partial function*, $F : Q \times \Phi \rightarrow Q$, which given a state and a function from the type Φ

determines the next state. F is often described as a *state transition diagram*.

- q_0 and m_0 are the *initial state* and *initial memory*, respectively.

Thus, adopting the X-Machine formalism requires modelling agents using the elements presented in the above Definition. Since the most basic definition of agent is that of an entity, situated in an environment that decides its next action based on its current percepts and state, a rather natural, clear and straightforward modelling of agents to X-Machine concepts involves mapping of agent percepts to the input alphabet Σ, its state to memory elements M and agent behaviour to the set of functions Φ and obviously the transition diagram F. Finally, agent actions are mapped to the output Γ.

In the generic model of pedestrians, three types of individuals, or agents, are simulated, as in [11], as we believe that with these three simple types it is possible to simulate the full diversity of individuals in a real-world crowd scenario as considered in this paper:

- *Individuals*, i.e. agents that perform the ritual alone. The population of individuals has a varying maximum speed, but otherwise an identical behaviour. Individuals also include people in wheelchairs.
- *Leaders*, that are agents that guide a group of individuals in the corridor. Leaders ensure that the members of their group are near by waiting for members that are left behind, but otherwise adopt the same behaviour as individuals.
- *Followers* are members of a group that follow a leader and have a very simple strategy to always simply move closer to their leader. If they loose contact with the latter, they adopt a similar behaviour to an individual until either they complete the ritual or find their group again.

It should be noted that all three types of agents are modelled using the same X-Machine, sharing parts of the behaviour that are common, while differentiating their behaviour when necessary. The following sections describe the X-Machine model used in the simulation scenario.

A. X-machine Input and Agent Percepts

Input Σ to the X-Machine represents the agent percepts P and consists of a number of symbolic descriptions of events in the environment. For instance, the $(Safah, exitReached)$ percept is part of the machine input Σ when the agent has reached the end of the corridor in the Safa hill side. Similar percepts exist for reaching other areas of the corridor, as the latter is described in section IV. Probably the most important percepts involve reporting of a possible next free location in the environment the agent can move to. Such locations are annotated as points of interest they are closer to, for instance percept $(p, closer\text{-}to\text{-}exit)$ indicates a free position p that is closer to the exit of the corridor. Finally, *followers* and *leaders* receive percepts that concern the status of the group, as for example the percept $follower\text{-}far$ that a leader agent receives, indicating that a member of its group has been left behind.

B. X-machine Memory and Agent Attributes and Beliefs

A number of memory variables allow modelling a diverse population, using a single X-Machine model. The list of memory variables includes for instance $speed$, holding the current speed of the agent, allowing to model a population of different age, $turns$, holding the number of turns the agent has performed thus far, etc. Special variables indicate the type of agent, for instance a $leader$ indicates whether the agent is a leader, and $followers$ is a variable holding the set of agents that belong to the group. Other variables hold more "dynamic" information regarding the state of the agent: $position$ holds the current position of the agent and is updated for every move that the agent performs. These variables are initialised at the time of agent creation, and updated through X-Machine functions.

C. X-Machine Functions, Agent Behaviour and State Diagram

X-Machine functions operate on memory M and input Σ (percepts P) and produce memory updates and output Γ. Each function acts as a guard to a state transition, thus each function has a number of conditions that evaluated to true provide the action (output). Consider the function $moveToExit$ that allows the agent to move closer to the corridor exit, thus implementing its "walking" strategy:

$$\begin{aligned} &moveToExit : (P, M) \mapsto (O, M') \\ &\quad \mathit{if} \quad (p, closer\text{-}to\text{-}exit) \in P \wedge corridorReached \in P \\ &\mathit{where}\ O = move\text{-}to(p), M' = M \mp (position, p) \end{aligned} \tag{1}$$

The operator $\mp$ stands for a memory position update, while the output of the function is the action $move\text{-}to(p)$, that moves the agent to the position p. The update in memory M' reflects the belief of the agent being in a new position.

D. X-Machine and Agent States

Functions, apart from producing outputs (agent actions), also guard states. The state diagram of the agents is presented in Figure 2, where "Entering" is the initial state of the agents and "Exiting" the final, i.e. in that state agents are considered to have left the simulation.

The state transition diagram of Figure 2 encodes this agent behaviour. Informally, agents enter the simulation environment in the entrance area, heading then to the turning area in front and walking towards the end of the corridor. Upon entering the corridor, they move towards the exit area neighbouring the specific corridor. When the exit area is reached, agents record the fact in their memory, change their state and head towards the turning area, close to that of the exit the agent is currently located. When reaching the turning area, the agent once more heads towards the corridor. This plan of moves is executed seven times according to the ritual.

IV. Simulation Implementation in NetLogo

The simulation of the the above mentioned case study was implemented using NetLogo [8]. During the last decade we have invested considerable time to develop executable models for various types of agent architectures and multi-agent

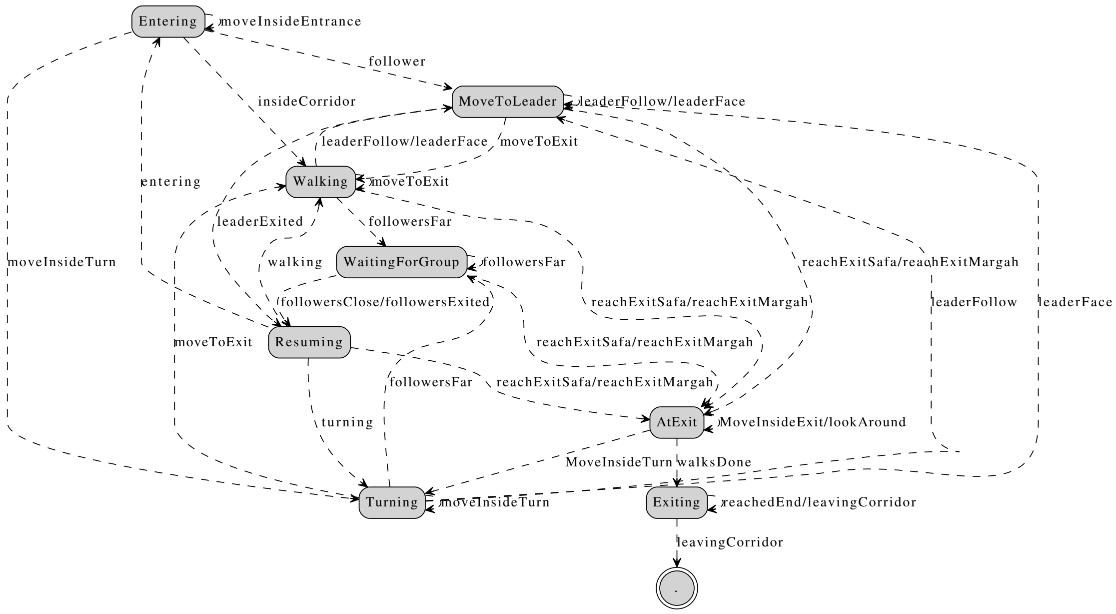


Fig. 2. The X-Machine State Transition Diagram

systems by building libraries to facilitate those. NetLogo was simple enough to demonstrate the validity of formal models we created and visualise properties that we could not formally verify due to the well-known state explosion problem of the model checking tools.

In order to represent the area under study in the simulator, a common assumption was adopted, according to which each individual occupies an area of 40 by 40 cm. Each such cell is modelled by a NetLogo patch. Based on this assumption an *initial corridor* was modelled in NetLogo (Figure 3), that has a smaller length (approx. 85m), but retains the width of the corridors and the size of the "turning" areas, i.e. the areas near Safa and Marwah.

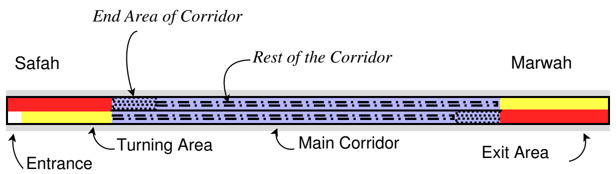


Fig. 3. The Corridor under Simulation in NetLogo.

There are four areas in the simulation environment, as depicted in Figure 3:

- the *entrance* area which presents the point at which agents enter the simulation,
- the *turning* area which is the area where agents make turns in order to walk back to the corridor,
- the *main corridor* area where agents walk, and
- the *exit* area that represents the exit of each corridor.

Each agent can occupy a single cell at a given time point and can move from one cell to the other, forward, left or right, overtaking other agents and avoiding obstacles.

```
to-report state-def-of-persons
 report (list
   state "Entering"
   # x-func "follower?" goto "MoveToLeader"
   # x-func "moveInsideTurn" goto "Turning"
   ...
   end-state

   state "Walking"
   # x-func "reachExitSafa" goto "AtExit"
   # x-func "reachExitMarwah" goto "AtExit"
   # x-func "leaderFar" goto "MoveToLeader"
   # x-func "leaderFarTurn" goto "MoveToLeader"
   # x-func "followerFarMove" goto "WaitingForGroup"
   # x-func "followersFar" goto "WaitingForGroup"
   # x-func "moveToExit" goto "Walking"
   # otherwise do "nothing" goto "Walking"
   end-state

   (... more states)

   state "Exiting"
   # x-func "reachedEnd" goto "Exiting"
   # x-func "leavingCorridor" goto "Exiting"
   end-state
    )
end
```

Fig. 4. X-machine diagram in NetLogo

The set of all available next positions the agent can move to with respect to its current speed S, is the neighbourhood of free (i.e. no other agents there) cells it can reach in S jumps. This is computed recursively from its current position, to reflect the fact that a valid path must exist to cells belonging to the set of cells reachable from the current cell of the agent. Given the above, behaviour such as overtaking, slowing down or even stopping due to congestion in the corridor, etc. present emergent properties of the simulation model.

A screenshot of the NetLogo interface for this system is provided at [12].

```
to-report moveToExit
 ifelse
    has-percept "corridorReached" and has-percept-type "closer-to-exit"
   [report x-true
     #< x-action task [person-move perception-list-value "closer-to-exit"] >#
     #< x-mem-set "position"  perception-list-value "closer-to-exit" >#         ]
   [report x-false ]
end
```

Fig. 5. An example X-Machine function, encoded in TXStates.

In order to directly support X-Machine execution in NetLogo, the TXStates DSL has been used [3]. The language supports easy encoding of states, transitions and functions, in a simple, intuitive syntax, thus allowing rapid development of X-Machine models. A meta-interpreter executes agent specifications respecting the X-machine semantics and thus allows simulations to be built easily. For example, part of the state diagram of Figure 2, encoded in TXStates is presented in Figure 4.

TXStates allows encoding X-machine functions as NetLogo reporters, that return outputs and memory updates. In this case, the meta-interpreter will directly execute the outputs, since the latter is an agent action and updates the memory values automatically without the user intervention. For instance, the implementation of the $moveToExit$ function (see Section III-C) in NetLogo using TXStates is shown in Figure 5.

Note. More details regarding the case study, including videos, agent behaviour tracking, NetLogo code, interface screenshots and various simulations are provided at [12].

V. Model Instantiation

One of the important steps in agent-based simulation is the model instantiation and validation. The method used for this simulation is to analyse real video recordings of high density crowd in different situations in order to extract patterns of behaviour of crowd movement, and implement them in the abstract model presented in this paper. In particular, we are interested in identifying the speed of movement of the different types of pedestrians in the Sa'yee such as men, women, elderly and disabled. The Institute of the Two Holly Mosques provided us with two hours CCTV video footages of the pilgrims performing the ritual during one day of the holy month of Ramadan in the disable priority lanes of the corridor. This data has been used to extract characteristic behaviour of crowd and real-world parameters to be used in the simulations.

In [13] pedestrians speed is measured by tracking their heads and applying a Voronoi diagram to divide the environment to a number of cells and then to calculate the density of people in each region. In [14], a semi-automatic method for calculating the trajectory and densities of pedestrians is suggested, by tracking their heads from video recording with an arbitrary observation angle, such as CCTV where the authors applied the method to video recording from experiments in which pedestrian flow was generated under controlled conditions. In [15], the relationships between speed and density are studied, and compared with the data available in the literature; however, the experiments and measurements were made again under laboratory settings. The approach presented here utilises real-world videos, with arbitrary observation angle, from a free hand camera, as well as fixed-point cameras such as for example in CCTV footage. We have utilised the video footage to extract pilgrims' trajectories and speed. The following limitations need to be reported: (i) the camera angle tracks the pedestrian for no more than 10 meters and (ii) we are not able to track pedestrians when the density is higher than two people per square metre – see Figure 6.

Fig. 6. Trajectory tracing

In particular, we have collected video recordings of different locations representing real-world pedestrians' activities in the high-density crowd situations such as the Sa'yee. Each video clip was observed carefully to extract generalizable behaviour, as described in Section III, and extract accurate parameters such as walking speed within groups for the different types of people in the scene.

TABLE I. Speed Values (In km/h)

Men	Women	Disabled
5.74	5.18	4.97
5.38	4.83	5.38

Relevant interesting clips have been identified within each of the video footages considered, representing about 10 seconds of crowd movement, and reduced to frames as in [15]. The frames have been loaded into the Petrack software [16] to extract the trajectory of each individual. In addition, important frames, such as the beginning and the end frame of the sequence, have been numerically analysed to extract other parameters such as the distance travelled and the walking speed. Having obtained the distance travelled and counting the number of frames per second in the sequence we are able to identify the speed of movement of the pedestrian in the footage considered. We have utilised the CCTV footage from Al-Haram corridor. Elements in the environment acted as a guide, for instance the columns distance is 10m and the width of the corridor is 5m. We have also considered the average speed for individuals as described in literature. In Table I are reported the average speed values for three types of individuals as measured from videos – first raw – and as reported in [17] – second row.

VI. Experiments, Results and Validation

We consider in the simulations below different populations of agents, either single agents or groups walking together through the corridor, three types of agents, men, women and disabled persons. These will be differentiated by their walking speed. These agents travel through the corridor and at the end they turn into the opposite lane to travel back, seven times, according to the Sa'yee ritual rules. We have started with agents walking at a speed close to the average values provided in Table I, first row. It has been soon observed that measuring the average speed when 400 and 500 agents have been walking in the corridor, due to the need to slow down when congestions appear, the average is below the values provided in Table I. Hence, it has been decided to allow agents walking at higher speed values. The issue is how these values might be chosen and how these are distributed across the three categories of agents. This calibration problem will be addressed in the next part of the paper together with experiments that aim to confirm some behaviour observed in videos like: the density of the agent population is higher towards the two ends of the corridors then in the rest, the agents that appear in groups tend to stick with the group and its leader, the agents walking at a higher speed overtake those walking at a lower speed.

A. Experiments and Results

In our simulation we allow agents to walk at a given maximum speed value which is randomly associated with the agent from a normal distribution with a mean value and a standard deviation. As mentioned above, these are the parameters we are aiming to determine in the calibration experiment. These values are considered for the case when no interactions appear. The values that we aim to identify have to be set to such values that match the real data. However, the real data concern individuals located in the middle section of the corridor where interactions with other agents do affect their average speed but congestion phenomena that occur at the ends do not. In order to calibrate the mean values, several simulations have been conducted that involved pedestrians walking the corridor once, i.e., agents enter the corridor at the Safa end at a constant rate and disappear when reaching the Marwah end. In the calibration process, two different values of the entrance rate were considered, that resulted in having approximately 400 and 500 agents in the corridor respectively. We have measured the average values of the "real" speed of the agents by generating a population of approximately 70000 agents. This process has resulted in speed distributions shown in Table II.

TABLE II. Speed distributions (70000 agents)

	Mean speed (cells/tick)	Deviation	Mean speed (m/sec)	Mean speed (km/h)
Men	4.1	0.4	1.64	5.9
Women	3.7	0.4	1.48	5.33
Disabled	3.5	0.1	1.4	5.04

As described in Section IV, each agent occupies an area (cell) of 40 by 40 cm. In Table II the average speed is expressed in cells/tick and transformed then in m/sec and km/h; for example for men these values are 4.1, 1.64 and 5.9, respectively. For agent type the standard deviation is also provided. So, both parameters of the normal distribution are now fixed for each agent type.

Mean speeds are slightly higher than the means reported in real results (see Table I) in order to get the right average speed that is affected by other pedestrians.

We have first considered a first experiment with 400 agents in a lane and we have run four times with different random seeds (i.e., a number of different experiments). The results, average speed (in km/h) measured for each of the experiments, and using the above mentioned normal distribution parameters are provided in Table III. A total of 70000 agents have been generated and the average speed computed.

TABLE III. Speed Values/400 Agents (In km/h)

Experiment	1	2	3	4
Men	5.76	5.76	5.76	5.76
Women	5.2	5.19	5.19	5.19
Disabled	4.93	4.73	4.88	4.88

In the case of 500 agents in a lane the results are provided in Table IV.

TABLE IV. Speed values/500 agents (in km/h)

Experiment	1	2	3	4
Men	5.67	5.64	5.63	5.63
Women	5.13	5.11	5.11	5.09
Disabled	4.81	4.75	4.74	4.81

One can observe that the values in Table III are closer to the real values from Table I and those in Table IV are lower than values in Table III, which is normal because the pedestrian density is higher in the first case.

B. Validation

Validation of any model against real data is a difficult, laborious and depends on various circumstances. In our case a more robust validation prococess based on some statistical data related to the simulation and real data is out of the question as the only real data collected consist of videos. Hence, we restrict ourselves to patterns observed in videos that are reproducible in simulations in certain controlled conditions. The first aspect that we considered is the density of the agent population at different points of the corridor. From the videos provided, one can observe that the density at each of the two ends is higher than in the rest. This property is a type of emergent aspect of the simulation that we aim to verify.

Multiple sets of experiments have been conducted. In the first class of experiments there have been only pilgrims performing the ritual alone (not groups), entering the corridor at a rate of five people per second. Pilgrims have been entering the corridor through the area marked as "Entrance" (Figure 3) and have been allowed to complete the ritual, i.e., seven cycles in the corridor. The generated population consisted of 5% women, 94% men and 1% disabled. We have conducted experiments with a maximum population between 500 and

1500 people in the corridor, considering increases of 250 people.

In all the experiments the density of people has been measured at the both ends of the corridor, as they are indicated in Figure 3, and in the rest of the corridor. The areas at the end have been approximately 24 m^2 and the results showing how densities evolve during the experiment are shown in Figure 7a. The figure presents the cases of the lowest and highest population in the corridor, i.e., 500 and 1500 agents, respectively.

One can observe that for a population of maximum 500 agents the average density value is almost the same at the ends of the corridor and in the rest of it, and has a value between 1 and 2 when the system is stable, after approx. 100 steps. In the case of 1500 agents, the average density at the ends is between 5 and 6 and in the rest of the corridor is between 4 and 5, after approx. 250 steps.

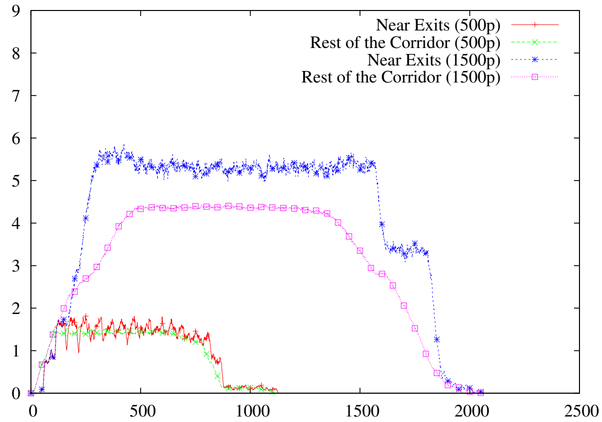
The second class of experiments involved groups of pilgrims. Allocation of pilgrims to groups followed a distribution that has been measured using real data collected. The data indicate that 46% of the pilgrims are individuals, 36% are couples, 6% form groups of three, 6% form groups of four and finally 6% are in groups of five people. Given the previous group characteristics, the simulation has involved again the generation of five pilgrims per second, either in groups or individuals, and the maximum number of people has been in the same range with the previous class of experiments. Results showing densities are indicated in Figure 7b.

From these two experiments one can assert that irrespective of the structure of the population, either only individual agents or a mixture of individuals and groups, the density distribution at the ends and inside the corridor is relatively the same and at the two ends is greater than in the corridor.

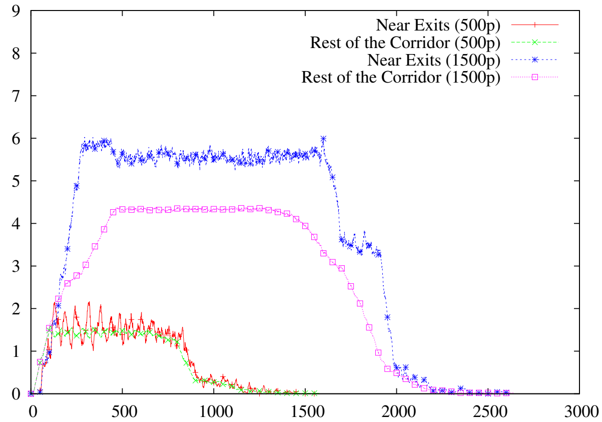
In both experiments, densities start increasing at the start of the simulation, which is expected as more pilgrims enter the simulation area. After a while, it is clearly observed in both classes of experiments that given a large enough number of agents, i.e., 1500 people in the corridor in order to reach a critical mass to create congestion, the densities at the ends of the corridor are higher than those in the centre of the corridor. When there is a small number of agents, congestion phenomena occur only temporarily, i.e., there are some time points in the simulation when the density at the ends is higher than that in the rest of the corridor, but this is due to the fact that a large number of pilgrims happen to arrive at the ends of the corridor simultaneously.

We have also run experiments with an initial population of agents in the corridor and the behaviour with respect to densities is the same. The results of these experiments and other experiments are presented in [12].

One other aspect of crowd simulation is believability of agents behaviour in simulation when compared to the real behaviour. The simulations performed shows that other expected patterns of behaviour, w.r.t. to videos, are overtakings and association of some agents to groups, which means they stick with their teammates as much as possible. These simulations and videos are available from [12]. Figures 8a and 8b present agents overtaking other agents in the context of a video and



(a) Density values with single agents



(b) Density values with groups

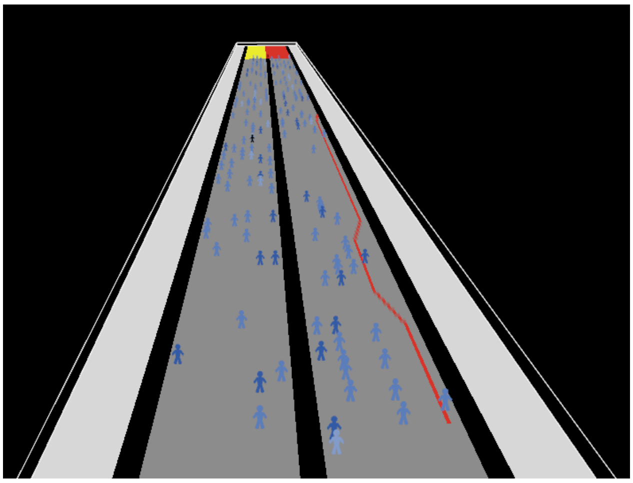
Fig. 7. Experimental results regarding the evolution of pedestrian densities in the different areas of the corridor.

the simulation, respectively. The behaviour of the agents in both cases is considered equivalent. However, the overtaking traces in Figure 8b is made out of linear trajectory due to the manner in which the navigation path is generated in NetLogo.

VII. Conclusions

In this paper we have presented a multifaceted approach to using agent-based systems for the simulation of crowd behaviour. The novelty of the approach consists in combining various methods and models, and in its flexibility to easily expand the population or the agents' behaviour. The case study is extracted from a real-life situation, but it is restricted to describing fluid movement of controlled crowd, including avoiding obstacles, accelerating or slowing down actions, group cohesion and gaps formation amongst groups. A formal model has been also presented, which allows a systematic way of developing the NetLogo code. The model, based on the X-machine formalism, is general enough allowing various further developments. One of them is the translation of the model into other software environments. We are currently working on translating it to FLAME, a powerful agent-based platform relying on X-machines, which is utilised for the simulation of large systems and provides support for different hardware platforms. This will allow us to expand the simulations to a

(a) Overtaking (video trace)



(b) Overtaking (NetLogo simulation trace)

Fig. 8. Videos comparing overtaking of pedestrians between real video footage and simulation.

even bigger number of agents, due to the well-know potential of FLAME in dealing with complex systems.

A medium term plan of further developments includes the use of other case studies involving crowd behaviour in shopping malls, such as Meadowhall. This will allow us to improve our approach regarding the adjustment of various parameters of the model and to consider other patterns of behaviour.

As we have already started investigating various social aspects [11] and psychological elements, like emotions [3], in agent-based systems, we aim to extend the approach presented in this paper to these topics and to integrate them with systems similar to that presented here.

ACKNOWLEDGMENT

Ionut Niculescu, Florentin Ipate and Marian Gheorghe are supported by a grant of the Romanian National Authority for Scientific Research, CNCS-UEFISCDI (project number: PN-II-ID-PCE-2011-3-0688). Marian Gheorghe acknowledges the support provided for the synthetic biology research by EPSRC ROADBLOCK (project number: EP/I031812/1). Daniela Romano research has been supported by a MoD Competition of Ideas grant, SPICE. Omar Kurdi is supported by a PhD studentship provided by the Saudi Arabia Government.

REFERENCES

[1] S. Zhou, D. Chen, W. Cai, L. Luo, M. Y. H. Low, F. Tian, V. S.-H. Tay, D. W. S. Ong, and B. D. Hamilton, "Crowd modeling and simulation technologies," *ACM Trans. Model. Comput. Simul.*, vol. 20, no. 4, pp. 20:1–20:35, Nov. 2010.

[2] M. Gheorghe, M. Holcombe, and P. Kefalas, "Computational models of collective foraging," *BioSystems*, vol. 61, pp. 133–141, 2001.

[3] I. Sakellariou, P. Kefalas, and I. Stamatopoulou, "Evacuation simulation through formal emotional agent based modelling," in *Proceedings of the 6th International Conference on Agents and Artificial Intelligence (ICAART 2014)*. SciTePress, 2014, pp. 193–200.

[4] I. Sakellariou, "Agent based modelling and simulation using state machines," in *2nd International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2012)*, 2012, pp. 270–279.

[5] ——, "Turtles as state machines - agent programming in netlogo using state machines," in *4th International Conference on Agents and Artificial Intelligence (ICAART 2012*, 2012, pp. 235–378.

[6] "FLAME: Flexible Large-scale Agent Modeling Environment (available at `http://www.flame.ac.uk/`)."

[7] S. Coakley, M. Gheorghe, M. Holcombe, S. Chin, D. Worth, and C. Greenough, "Exploitation of high performance computing in the flame agent-based simulation framework," in *IEEE High Performance Computing and Communication (HPCC-2012)*, June 2012, pp. 538–545.

[8] U. Wilensky, *NetLogo*, Center for Connected Learning and Computer-Based Modeling, Northwestern Univ., Evanston, IL., 1999, http://ccl.northwestern.edu/netlogo/.

[9] F. Abdessemed, "An integrated system for tracking and control pilgrims shuttle buses," in *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 384–389.

[10] M. Holcombe and F. Ipate, *Correct Systems – Building a Business Process Solution*. Springer, 1998.

[11] D. Romano, M. Kiran, and T. Karmakharm, "Multi-massive believable heterogeneous social agent-based pedestrians on the gpu," *IEEE Transactions on Computational Intelligence and AI in Games*, submitted.

[12] "Sa'yee webpage with experiments and videos (available at `http://muvet.ifsoft.ro/sayee/`)."

[13] B. Steffen 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, pp. 1902–1910, 2010.

[14] M. Plaue, M. Chen, G. Bärwolff, and H. Schwandt, "Trajectory extraction and density analysis of intersecting pedestrian flows from video recordings," in *Photogrammetric Image Analysis*, 2011, pp. 285–296.

[15] A. Seyfried, B. Steffen, and T. Lippert, "Basics of modelling the pedestrian flow," *Physica A: Statistical Mechanics and its Applications*, vol. 368, no. 1, pp. 232–238, 2006.

[16] M. Boltes, A. Seyfried, B. Steffen, and A. Schadschneider, "Automatic extraction of pedestrian trajectories from video recordings," in *Pedestrian and Evacuation Dynamics*, 2010, pp. 43–54.

[17] M. Carey, "Establishing pedestrian walking speeds," in *Project report*, 2005.