Pattern Recognition Letters 44 (2014) 30–38

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

An agent-based crowd behaviour model for real time crowd behaviour simulation ☆

Vassilios Kountouriotis [a,*], Stelios C.A. Thomopoulos [a], Yiannis Papelis [b]

[a] Integrated Systems Laboratory, Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", Agia Paraskevi Attikis, PO Box 60228, 15310 Athens, Greece
[b] Virginia Modeling, Analysis and Simulation Center, Old Dominion University, 5155 Hampton Boulevard Norfolk, VA 23529, United States

ARTICLE INFO

Article history:
Available online 1 November 2013

Communicated by Rita Cucchiara

Keywords:
Simulation
Crowd behaviour
Social forces
Flow fields
Real-time

ABSTRACT

Crowd behaviour models are divided into agent-based, flow-based and particle-based in terms of whether the behaviour emerges from simulating all people (agents) individually (Koh and Zhou, 2011; Braun et al., 2005; Luo et al., 2008; Pan et al., 2007; Shendarkar et al., 2006; Narain et al., 2009), is programmatically defined a priori using fluid dynamics models (Hughes, 2002, 2003; He et al., 2011), or employ a particle system governed by physical laws (Helbing et al., 2000; Bouvier et al., 1997; Treuille et al., 2006; Cucker and Smale, 2007). In agent-based models, computationally intense problems, such as global navigation, hinder the efficient real-time modelling of thousands of agents. In this paper we present a novel approach to crowd behaviour modelling which couples the agent-based paradigm of allowing high level of individual parametrization (group behaviour between friends, leader/follower individuals) with an efficient approach to computationally intensive problems encountered in very large number of agents thus enabling the simulation of thousands of agents in real time using a simple desktop PC.

1. Introduction

Modelling the behaviour of a large number of individuals is a task associated with high complexity, not only because of the sheer number of individuals often comprising crowds but also because their interactions, movements and general behaviour are the result of highly complex and psychological interdependencies, not suitable for direct one-to-one simulation. The effort, therefore, shifts to emulating the general behaviour of a crowd using basic principles and heuristics that closely resemble reality.

In this respect, there have been mainly two directly opposite poles of attraction for model designers and architects. One possibility is to model each individual (agent) in the crowd separately and expect the crowd's behaviour to emerge as all agents interact (Koh and Zhou, 2011; Braun et al., 2005; Luo et al., 2008; Pan et al., 2007; Shendarkar et al., 2006; Narain et al., 2009; Bouvier et al., 1997; Treuille et al., 2006; Cucker and Smale, 2007) and the other to regard the whole crowd as a continuous and homogeneous fluid governed by the laws of nature as defined and studied by the science of fluid dynamics (Hughes, 2003, 2002; He et al., 2011).

Recent works (Narain et al., 2009) have attempted to combine the much more complex and refined emergent behaviour of agent-based models with the efficiency of flow-based models by proposing hybrid models where agents still exist but from certain pressure/density levels upwards, their behaviour more or less mimics particles in fluids.

Agent-based models are better suited to situations where a high level of realism is desired. Unfortunately, most research is focused on the individual leaving groups dynamics such as people being friends or family members relatively untouched: Helbing and Molnár (1998) have proposed attractive forces which fade over time (agents lose interest) but attractive forces are not always fading such as the attractive forces between members of a family. Furthermore, inside these groups we can usually differentiate leaders from follower individuals. These dynamics can have a considerable effect on the overall emergent behaviour of the crowd.

In this paper, we present our own approach to crowd behaviour modelling with support for group dynamics and different agent personality traits. Our model allows for high parametrization of the individual: each individual has her own physical and psychological traits, ranging from mass, top speed and acceleration to leader or follower personality and has the ability to be part of tightly or loosely bound groups such as family or friends. Furthermore, our model is efficient enough to simulate thousands of agents in real-time on a single CPU.

☆ This work is partially funded by the European Commission under the research project "Total Airport Security System: TASS-FP7-SEC-2010-241905."

* Corresponding author. Tel.: +30 2106503128.
E-mail addresses: b.kountouriotis@iit.demokritos.gr (V. Kountouriotis), scat@iit.demokritos.gr (S.C.A. Thomopoulos), ypapelis@odu.edu (Y. Papelis).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.10.024

1.1. Paper structure

The rest of the paper is organized as follows: in Section 2 we present work which closely relates to ours and in many ways laid the foundations for our model development; in Section 3 we present our model in detail; Section 4 is dedicated to our model's performance; Section 5 discusses the model's qualitative aspects and compares its behaviour to real-life experiments where available and finally Section 6 briefly discusses our conclusions and future work we would like to partake.

2. Related work

Researchers generally prefer and focus on agent-based modelling since it provides a much more refined framework for describing complex human behaviour. They formally describe each individual agent's behaviour and let the crowd's behaviour emerge; the modelling, therefore, is thorough enough to allow realistic agent behaviour while at the same time trying to retain the computational complexity to an acceptable upper limit. Some models incorporate psychological information and factors in the model for better accuracy (Pelechano et al., 2007; Braun et al., 2003; Sakuma et al., 2005) while others omit psychology altogether and only directly simulate in-view agents, leaving the rest to probabilistic methods (Arikan et al., 2001).

One of the main diversifying issues among agent-based models is the underlying movement model for the agents and specifically the way they avoid collisions with obstacles and other agents and the way they decide on a path to their goal. Collision avoidance can be based on Helbing's *Social forces model* (Helbing and Molnár, 1998), *steering behaviours* (Reynolds, 1999), a combination of the two (Moussaid et al.), or even correction of agents' velocities using aggregate movement dynamics from nearby agents (Narain et al., 2009). Global path planning, on the other hand, is achieved by running a path searching algorithm such as A* or Dijkstra on a carefully prepared graph representation or grid (i.e. quadtree) of the environment (Shao and Terzopoulos, 2007; Shendarkar et al., 2006).

The movement model is usually complimented by a psychological and emotional state model which endows agents with the ability to react to stimuli differently depending on their psychological and emotional state: Modelling here is based on Belief-Desire-Intention (BDI) models (Shendarkar et al., 2006; Rao and Georgeff, 1992), decision networks (Yu and Terzopoulos, 2007), Bayesian networks (Hy et al., 2004; Pearl, 1988; Ball and Breese, 2000) and even fully layered frameworks embodying an agent's situational awareness and consequence foreseeing into the model (Luo et al., 2008).

Finally, many researchers have incorporated environmental perception into their models. Various methods have been proposed for the propagation and perception of auditory (Monzani and Thalmann) and visual (Funge et al., 1999; Noser et al., 1995; Kuffer and Latombe, 1999; Renault et al., 1990) stimuli; others have proposed models for combined audio, visual and haptic sensory inputs (Conde and Thalmann, 2004). Most of them are, generally, computationally expensive.

3. A model for real time crowd behaviour simulation

3.1. Overview

We propose an agent based model with fully autonomous agents using a three-layered approach: at the higher level is the *cognitive layer* which models the emotional state of each agent, using a set of preset emotions and a quantification of the intensity each emotion is being "felt" by the agent. These emotions are affected by the agent's interaction with the virtual world (*"Was that an explosion I heard? Why are these people running?"*) and events in the virtual world have an impact on them. To achieve this, human perception is modelled through perception limitation structures such as visibility graphs and by using specific excitement modifiers communicated between agents, each agent can influence those nearby (those who see him poll him for influence modifiers to be applied to their own states). An agent in panic, for example, will broadcast high values of stress and fear to be added to the states of agents listening who will update their states accordingly. The combined intensities of these emotions affect the lower layer, the *strategic* layer: this is responsible for the decision making process of the agent and is implemented using hierarchical state machines: *"Now that I have completed my check-in, where should I proceed to?"*.

At the bottom of the stack rests the *movement* layer. This is responsible for realizing the actual movement of the agent towards her goal, as set by the strategic layer while avoiding structural obstacles and pitfalls as well as collisions with other agents. Our is a *social forces – flow-field – aggregate dynamics* model (Helbing et al., 2000; Narain et al., 2009; Patil et al., 2011).

3.2. Movement model

At the core of our movement model is the refined social forces model proposed by Helbing et al. (2000) extended to allow flow-field based navigation and aggregate dynamics built-in.

For each agent α and given a specific time $t, \vec{r}_\alpha(t)$ is the agent's absolute position vector on the world (the point of her center of mass), m_α is her mass, $\vec{v}_\alpha(t)$ her actual velocity vector and r_α her body's radius. Furthermore, depending on the state she is in, she has a desired speed $s_\alpha(t)$.

3.2.1. Self-force

The self-force represents the force an agent applies on herself in order to reach her destination, while travelling at her desired speed $s_\alpha(t)$. This force can be broken down into two components:

(i) the *goal self-force* which is a force in the direction of the shortest path to the agent's destination and
(ii) the *aggregate self-force* which is a force exerted on the agent because of the general movement of a dense crowd she is in.

3.2.1.1. Goal self-force. Agents in our model are pre-endowed with complete information on the shortest path to their next waypoint within a virtual room. This is exploited by agents applying a force on themselves pushing them forward in the direction of the shortest path to their destination, called the *goal self-force*. The intensity of the force depends on the difference between their current velocity $\vec{v}_a(t)$ and their desired velocity $s_\alpha(t)\vec{e}_G(\vec{r}_\alpha(t))$:

$$\vec{f}_\alpha^{goal}(t) = \frac{m_\alpha}{\tau_\alpha}(\vec{v}_\alpha(t) - s_\alpha(t)\vec{e}_G(\vec{r}_\alpha(t))) \quad (1)$$

where τ_α is a *relaxation time* controlling the agent's acceleration and $\vec{e}_G(\vec{r}_\alpha(t))$ is the direction (vector) of the shortest path from point $\vec{r}_\alpha(t)$ to point G, where G is α's goal.

3.2.1.2. Rapid Flow-Field Computation algorithm. Since the shortest path computations are considerably expensive and cannot be re-calculated continuously for every agent separately we have developed a Rapid Flow-Field Computation (RFFC) algorithm for fast calculation of the directions of the shortest paths from any given point to any goal. Specifically, a grid is overlaid on the simulated area and for each cell and each possible goal, we pre-compute the direction agents should follow for shortest access to said goal.

The direction of each cell is computed before starting the simulation and is stored in memory. When the simulation begins, this information is loaded and agents have direct access to it. In the following, $\vec{e}_G(C)$ refers to the direction of the field for the goal G in the cell C, $T_G(C)$ refers to the cost (with respect to G) of the cell C. Furthermore, given a cell $C = (i,j)$ and its Von Neumann neighbourhood N, we will refer to cells $P = (i_P, j_P), Q = (i_Q, j_Q) \in N$ as *paired Von Neumann neighbours* of C if and only if: $P \neq Q, i_P \neq i_Q$ and $j_P \neq j_Q$. To better illustrate this, refer to Fig. 1: there are 4 sets of paired neighbours of C : $\{M,N\}, \{N,Q\}, \{Q,P\}$ and $\{P,M\}$. The computation of the flowfield is described below.

Algorithm 1. Rapid Flow-Field Computation (RFFC)

function LERP(C, M, N, T_G)
  $\alpha_0 \leftarrow \min_\alpha \left\{ (1-\alpha)T_G(M) + \alpha T_G(N) + \sqrt{\alpha^2 + (1-\alpha)^2} \right\}$
  $0 \leqslant \alpha \leqslant 1$
  return α_0
procedure RAPIDFLOWFIELDCOMPUTATION(G)
  $T(G) \leftarrow 0$
  $Q \leftarrow$ Von Neumann neighbours of G
  while Q is not empty **do**
    $C \leftarrow$ POPQUEUE(Q)
    for all K: not computed Von Neumann neighbour of C **do**
      PUSHQUEUE(Q, K)
    if C has only 1 computed Von Neumann neighbour, K **then**
      $\vec{e}_G(C) \leftarrow \text{dir}(\overrightarrow{CK})$
      $T_G(C) \leftarrow T_G(K) + 1$
    if C has 2 or more computed Von Neumann neighbours **then**
      $\alpha_0 \leftarrow \min\{\text{Lerp}C, M, N, T_G | M, N$ paired neighbours of $C\}$
      $\vec{e}_G(C) \leftarrow \text{dir}(\overrightarrow{CX})$
      $T_G(C) \leftarrow \alpha_0$

A brief description of the logic of the RFFC algorithm follows:

(i) LERP
Assume that M, N are paired Von Neumann neighbours of C and we want to calculate the direction $\overrightarrow{CX}$ which points to the shortest path to the goal G based on the costs of M and N and their directions, $\vec{e}_G(M)$ and $\vec{e}_G(N)$ (see Fig. 1). The cost of C is the cost to move from C to X which is the distance $\|\overrightarrow{CX}\|$ plus the cost at X which can be found by linearly interpolating the costs $T_G(M)$ and $T_G(N)$ on X:

$$h_G(\alpha) = (1-\alpha)T_G(M) + \alpha T_G(N) + \sqrt{\alpha^2 + (1-\alpha)^2}, \quad 0 \leqslant \alpha \leqslant 1 \tag{2}$$

By finding α_0 which minimizes $h_G(\alpha)$ in $[0,1]$ we get the cost $T_G(C) = h_G(\alpha_0)$ and direction $\vec{e}_G(C) = \dfrac{\overrightarrow{CX}}{\|\overrightarrow{CX}\|}$.

(ii) RAPIDFLOWFIELDCOMPUTATION
Computation of the flow-field starts from the goal G and expands by pushing its Von Neumann neighbours to the *to-be-processed* queue. Then, while this queue is not empty, pop the first element C, push its not-yet-computed Von Neumann neighbours in the queue and calculate its cost and direction in the following manner:

(a) if there is only one Von Neumann neighbour of C which has been computed, the direction of C is towards the neighbour and the cost that of the neighbour plus 1.

(b) if there are more than 1 Von Neumann neighbours which have been computed, for each paired set of neighbours find the minimum interpolated cost and finally select the pair with the lower minimum. Finally set the cost and direction of C as provided by the minimum interpolated cost.

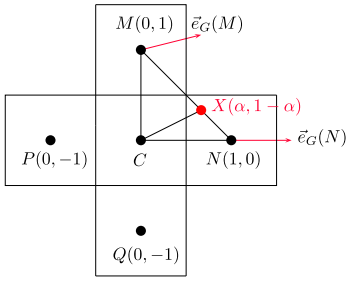


Fig. 1. A cell C with its Von Neumann neighbours M, N, P and Q. Of those, M and N are *paired Von Neumann* neighbours of C which have been already computed and as such the direction of C can be calculated.

3.2.1.3. Aggregate self-force. The *aggregate self-force* is the force exerted on agents when inside a dense crowd which essentially sweeps them to its own direction, voluntarily or involuntarily limiting their movement options. This force on an agent α is expressed in terms of the difference of α's current and the crowd's average velocities:

$$\vec{f}_\alpha^{agg}(t) = \frac{m_\alpha}{\tau_\alpha} \cdot \left(\vec{v}_\alpha(t) - \frac{\sum_i \{ m_i \vec{v}_i(t) \mid i \in A_{\vec{r}_\alpha(t)} \}}{\sum_i \{ m_i \mid i \in A_{\vec{r}_\alpha(t)} \}} \right) \tag{3}$$

where $A_{\vec{r}_\alpha(t)}$ is the set of agents in a neighbourhood of $\vec{r}_\alpha(t)$ and τ_α is a *relaxation time* controlling the agent's acceleration.

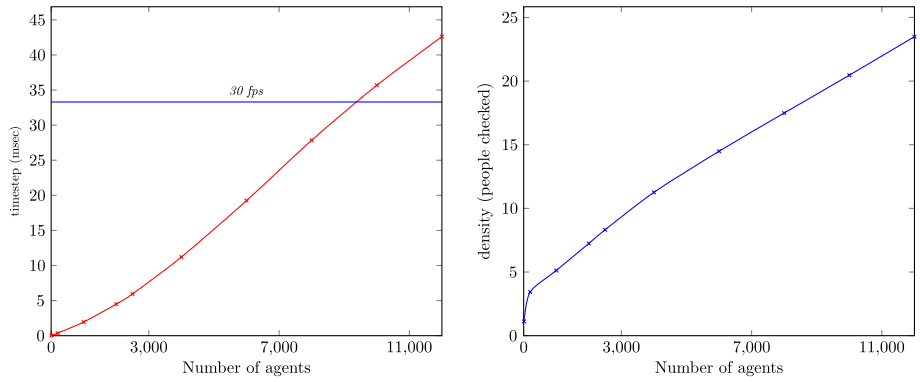
Tracking the aggregate self-force involves tracking the density of the area near each agent. To achieve this, we partition the area into cells, each one containing a set of agents which happen to be within its boundaries. Since agents' size (body radius) is known and the repulsive forces between them specific (see Section 3.2.2), there is an upper limit in the amount of agents that can fit in a cell, p_{max}. Given a neighbourhood N, the *neighbourhood density* of N is:

$$\rho_N(t) = \frac{\sum_i^n \left\{ p_{C_i}(t) \mid C_i \in N \right\}}{|N| \cdot p_{max}} \tag{4}$$

where $p_C(t)$ is the number of agents in cell C at time t.

By definition, for any N and t, $\rho_N(t)$ is a real number between 0 and 1 (inclusive). We use this in the definition of the total *self-force* to linearly interpolate the goal self-force and aggregate self-force: if $\rho_N(t)$ is near 1 (very dense crowd), then the goal self-force is almost ignored and the aggregate is responsible for the total self-force; if, on the other hand $\rho_N(t)$ is near 0 (agent is alone), then the aggregate self-force is almost ignored and the goal self-force used instead. Note that in the event of streams of crowds moving in opposite directions there is a chance that the aggregate self-force in the cell neighbouring these streams is zero. If such situations arise in a setting, one can interpolate between μ and 1 instead of 0 and 1 so that the goal self-force will always be present by a factor μ.

3.2.1.4. Total self-force. The total self-force is the force an agent applies onto herself in her attempt to reach her goal combined with the involuntary movement she makes when part of a dense



(a) Performance (b) Density

Fig. 2. Simulator performance: time needed for each time step with respect to number of agents simulated and the average density measured by how many agents are taken into account for repulsive forces.

crowd and is a linear combination of the goal self-force and aggregate self-force:

$$\vec{F}_\alpha^{self}(t) = \vec{f}_\alpha^{goal}(t) + \rho_{N_{\vec{r}_\alpha(t)}}(t)(\vec{f}_\alpha^{agg}(t) - \vec{f}_\alpha^{goal}(t)) \quad (5)$$

where $N_{\vec{r}_\alpha(t)}$ is the set of the 4 cells with the closest centres to $\vec{r}_\alpha(t)$.

3.2.2. Repulsive force

When pedestrians move in an environment, they tend to avoid – for psychological reasons – being very close to strange individuals and walls. Instead, they choose to travel in a way which allows for some distance between them and other animate and inanimate objects. We model this behaviour by applying repulsive forces to each agent, exerted from other agents and walls. The total repulsive force exerted to an agent has three components:

i. the **direct repulsive force** $\vec{f}^R$ because of the proximity of the agent to an entity,
ii. the **friction force** $\vec{f}^F$ due to agents' touching other entities and
iii. the **counteracting body compression** force $\vec{f}^C$ which the agent exerts on himself to counteract other agents compressing him.

To describe the direction of the repulsive forces, we will employ two characteristic vectors: $\vec{n}_{\alpha,i}(t)$ which will be the direction from agent i to agent α if i is an agent or perpendicular to the wall or obstacle i if i is a wall or obstacle and $\vec{q}_{\alpha,i}(t)$ which will be the direction tangential to $\vec{n}_{\alpha,i}(t)$. Finally, we will be using the non overlapping distance between agent α and entity i:

$$d_{\alpha,i}(t) = d^0_{\alpha,i}(t) - (r_\alpha + \hat{r}_i) \quad (6)$$

where $\hat{r}_i$ is the radius of i's body if i is an agent and 0 otherwise and $d^0_{\alpha,i}(t)$ is the distance between two agents' centres of mass or an agent's centre of mass and the closest point of a wall or obstacle.

3.2.2.1. Direct repulsive force. The direct repulsive force is a force exerted on individuals because of their psychological pressure to have an unoccupied comfort zone around them. It is being represented as a repulsive force, much like an electromagnetic one between two charges of the same type: it increases as agents get closer and decreases as agents move away. The *direct repulsive force* exerted on α because of i is:

$$\vec{f}^r_{\alpha,i}(t) = \left(A_\alpha e^{-\frac{d_{\alpha,i}(t)}{B_\alpha}}\right)\frac{\vec{n}_{\alpha,i}(t)}{\|\vec{n}_{\alpha,i}(t)\|} \quad (7)$$

where A_α is a constant expressing the magnitude of influence that the personality of agent α receives (provided by the cognitive layer) and B_α is a constant controlling how sensitive agent α is to other entities closing in.

Eq. (7) presumes that the influence pedestrians receive from other agents and objects in their environment is dependent only on the distance between them. In reality, this influence is not constant but also depends on whether the source of the influence is within the agent's field-of-view. For this, we introduce *direction-dependent weights*:

Table 1
Simulator performance on a single intel i7 3.4 GHz CPU, compiled on ×64.

Agents	Timestep (ms)	Density (average people checked)
10	0.03	1.12
200	0.36	3.43
1000	1.95	5.12
2000	4.50	7.24
2500	5.96	8.32
4000	11.21	11.26
6000	19.25	14.49
8000	27.84	17.49
10000	35.68	20.46
12000	42.61	23.49

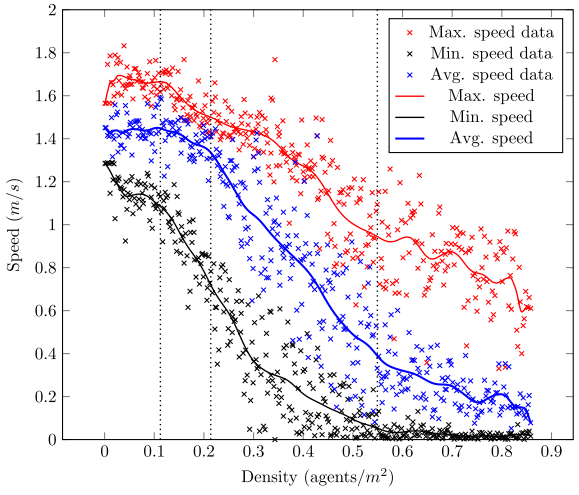


Fig. 3. Fundamental diagram of pedestrian flow: minimum, average and maximum speed over a 100 s period.

$$w_c(\vec{e},\vec{f}) = \begin{cases} 1, & \text{if } \vec{e}\cdot\vec{f} \geqslant \|\vec{f}\|\cos\phi \\ 1-c, & \text{otherwise} \end{cases} \quad (8)$$

where 2ϕ is the *effective angle of sight* and $0 \leqslant c \leqslant 1$is the *influence attenuation factor* which represents how much the influence is attenuated when the source of influence is out of sight.

Using (8) we can modify (7) to account for the field-of-view dependency of the repulsive influence, providing the final equation for the direct repulsive force:

$$\vec{f}^{R}_{\alpha,i}(t) = w_c(\vec{v}_\alpha(t),\vec{f}^{r}_{\alpha,i}(t))\cdot\vec{f}^{r}_{\alpha,i}(t) \quad (9)$$

3.2.2.2. Friction force. When agents come very close with other agents or objects, their movement is impaired because of the friction this closeness implies. The friction depends on the difference of the velocities and the closeness of the entities involved. The force exerted on an agent α because of the friction generated when moving close to entity i is:

$$\vec{f}^{F}_{\alpha,i}(t) = m_\alpha \kappa g_{\alpha,i}(t)\Delta v^{\vec{q}_{\alpha,i}(t)}_{\alpha,i}\vec{q}_{\alpha,i}(t) \quad (10)$$

where $g_{\alpha,i}(t)$ is the *granularity impairment* which is the length of the overlap between agents α and i or agent α and wall i and κ is a parameter defining the intensity of the friction because of compression and

$$\Delta v^{\vec{q}_{\alpha,i}(t)}_{\alpha,i} = (\vec{v}_i(t) - \vec{v}_\alpha(t))\cdot\vec{q}_{\alpha,i}(t)$$

Obviously, if i is a wall or obstacle, $\vec{v}_i(t) = 0$.

3.2.2.3. Counteracting body compression force. The *counteracting body compression force* is a force exerted on agents when the agents' density reaches a level high enough that agents are forced to overlap (or squeeze with each other). The force on an agent α because she is being compressed on entity i is:

$$\vec{f}^{C}_{\alpha,i}(t) = m_\alpha k g_{\alpha,i}(t)\frac{\vec{n}_{\alpha,i}(t)}{\|\vec{n}_{\alpha,i}(t)\|} \quad (11)$$

where $g_{\alpha,i}(t)$ is the *granularity impairment* which is the length of overlap between agents α and i or agent α and wall i and k is a parameter defining the intensity of the obstruction because of friction.

This force is important for another reason besides navigation. If this force exceeds a certain threshold, the pedestrian would start suffering health issues from breathing difficulties to even death. This can be tracked by the simulator for evacuation performance reports with respect to costs in human lives.

3.2.2.4. Total repulsive force. Given Eqs. (9)–(11) one can give the full repulsive force exerted on an agent α because of an entity i, which is the sum of these forces:

$$\vec{F}^{rep}_{\alpha,i}(t) = \vec{f}^{R}_{\alpha,i}(t) + \vec{f}^{F}_{\alpha,i}(t) + \vec{f}^{C}_{\alpha,i}(t) \quad (12)$$

3.2.3. Attractive force

Real-life pedestrians often travel in groups such as families or friends whose members exert attractive influences on each other. Furthermore, there exist distractions such as shows, displays which attract the pedestrians' attention who may divert towards the attractions. These are modelled much the same way as the various repulsive forces but with one difference: attraction forces may be permanent (between families and friends) or temporary, fading as time passes for each individual (various distractions).

The *attractive force* exerted on an agent α because of a simulation entity i is:

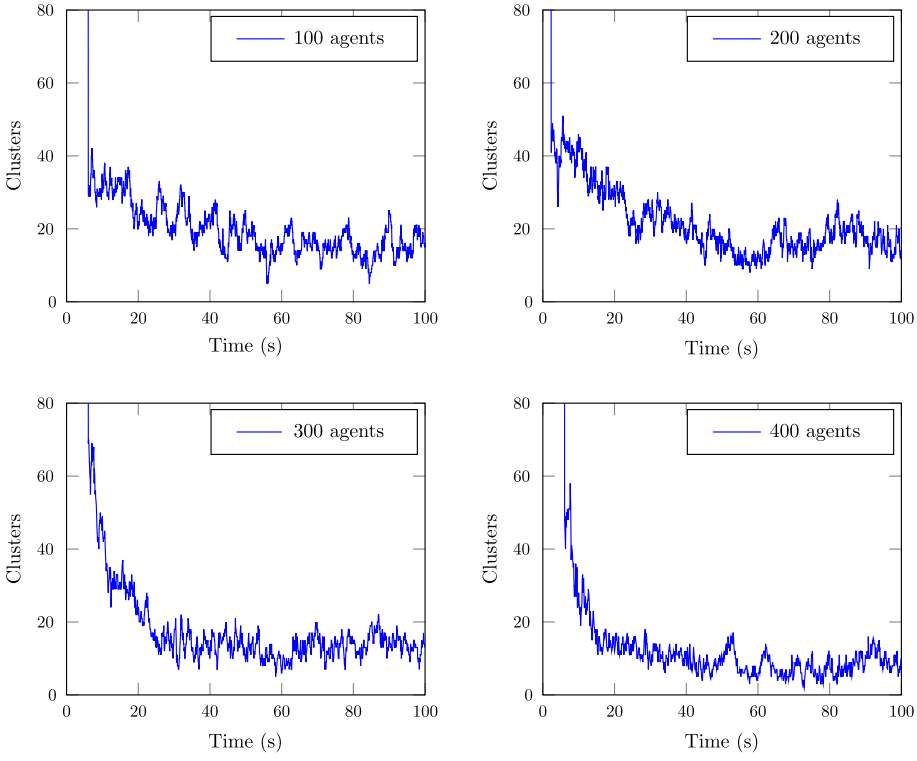


Fig. 4. Evolution of clusters for 100, 200, 300 and 400 agent crowds (1 agent per 11.6 m^2, 5.8 m^2, 3.9 m^2 and 2.9 m^2. Clusters were calculated with τ = 1 s and δ = 1.2 m/s which is about the average speed of the crowd for these densities.

$$\vec{F}^{attr}_{\alpha,i}(t) = \left(A_\alpha K_{\alpha,i}(t-t_0) e^{-\frac{d^0_{\alpha,i}(t)}{B_\alpha}} \right) \frac{\vec{n}_{\alpha,i}(t)}{\|\vec{n}_{\alpha,i}(t)\|} \tag{13}$$

where A_α is a constant expressing the magnitude of influence that the personality of agent α receives (provided by the cognitive layer; *leader* agents have much smaller A_α than *follower* ones and for every group there is an agent with the lowest A_α which is considered the 'head' of the group), B_α is a constant controlling how sensitive agent α is to other entities closing in, t_0 is the time where agent α starts being attracted by entity i, $d^0_{\alpha,i}(t)$ is the non-overlapping distance between α and i (see Eq. (6)) and and $K_{\alpha,i}(t-t_0)$ is a monotonically decreasing function between (inclusive) 0 and 1 representing the fading of interest agent α has for entity i over time.

3.2.4. Total force

Having defined the self, repulsive and attractive forces on an agent α because of entity i we can write the equation for the total force exerted on α because of i:

$$\vec{F}_{\alpha,i}(t) = \vec{F}^{self}_{\alpha,i}(t) + \vec{F}^{rep}_{\alpha,i}(t) + \vec{F}^{attr}_{\alpha,i}(t) \tag{14}$$

Since there are many entities in the simulation which influence a specific agent α, the total force exerted on her is the sum of all the forces exerted on α due to all other entities:

$$\vec{F}_\alpha(t) = \sum_{i}^{i \in E} \vec{F}_{\alpha,i}(t), \quad i \neq \alpha \tag{15}$$

where E is the set of all entities in the simulation. In the actual implementation, not all entities contribute to the total force on an agent α; only those that are in the neighbouring cells closest to the one agent α is in.

3.3. Special movement areas

The social forces/flow-field/aggregate dynamics model is the dominating movement model of our simulator. This is complemented with arbitrary specialized movement models, each tailored to specific situations. Such models are, e.g. the movement model of pedestrians inside a waiting line (queue), or the movement model of pedestrians inside an escalator.

3.3.1. Waiting lines

Pedestrians do not move according to social forces, flow-fields and aggregate dynamics inside waiting lines. Overtaking is socially prohibited and every pedestrian closely follows the one in front until they are in front at which point they wait for the next service point to become free and head towards it. Upon finishing, they may head towards a queue exit point and then continue to their next goal.

Such a behaviour is incompatible with the enhanced social forces model we propose; in cases like these out agents have the ability to use different movement models. The movement model inside waiting lines and queue is based on a simple PID controller (Minorsky, 1922) controlling the agent's desired velocity $\vec{S}_\alpha(t)$ with respect to her distance from the agent in front, d_f. The "error" in velocities is $E(t) = \vec{S}_\alpha(t) - \vec{v}_\alpha(t)$ where $\vec{v}_\alpha(t)$ is the current velocity of α. The correction, therefore, in agent α's velocity $\vec{v}_\alpha$ is:

$$\vec{c}_\alpha(t) = K_p E(t) + K_i \int_0^t E(x)dx + K_d \frac{dE(t)}{dt} \tag{16}$$

where K_p, K_i and K_d are constants controlling the acceleration and stability of the controller. Naturally, $\frac{d|\vec{c}_\alpha(t)|}{dt}$ is capped by the maximum acceleration allowed by the state α is in. When α is in front of the line and a service point is available, d_f is the distance to the service point.

3.3.2. Escalators

Pedestrians in escalators (and corridors) generally either stay put or keep a constant velocity with respect to the escalator/corridor while maintaining a comfortable distance from the agent ahead. The agent's speed, therefore, increases quickly if she continues to walk (with respect to the escalator or corridor's velocity) or matches the corridor's and keeps constant until the agent exits or gets close to the agent ahead at which point he slows down and stops. To model this behaviour, we use the technique and a PID controller in the same way as the case for the waiting lines. The velocity of an agent α on an escalator or a corridor is

$$\vec{v}_\alpha = \left(\frac{\|\vec{S}_\alpha(t)\|}{\|\vec{V}_e(t)\|} + 1 \right) \vec{V}_e(t) \tag{17}$$

where $\vec{V}_e(t)$ is the escalator's velocity and $\vec{S}_\alpha(t)$ is the desired velocity as corrected by the PID controller.

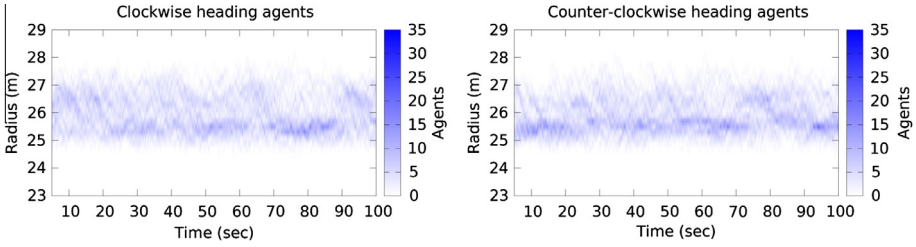


Fig. 5. Distribution of agents with respect to their distance from the annulus' center over time for a crowd of 100 agents.

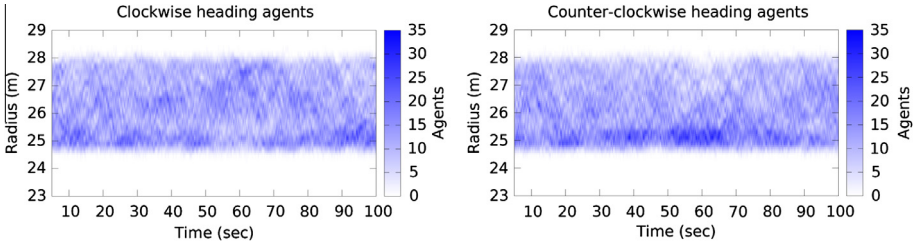


Fig. 6. Distribution of agents with respect to their distance from the annulus' center over time for a crowd of 400 agents.

4. Performance

In its current form, the simulator runs many threads in parallel, each responsible for a specific task. There exists one dispatch thread (or main thread) which handles synchronization issues and coordinates all the other ones in respect to how frequently they start working (tick). In our crowd simulator, the movement thread (which runs the movement model) runs 5 times more frequently than the strategy thread and 10 times more frequently than the cognitive thread. The performance of the movement model is carefully monitored and the target is to achieve less than 33.3 ms per timestep in order to have 30 timesteps per second. Obviously, as the number of agents increases, so does the time needed to calculate each timestep.

To measure the performance of the simulator, we placed agents in a large airport's terminals (we used Athens International Airport – AIA, a partner of the *TASS* consortium in the TASS-FP7-SEC-2010–241905 FP7 project) and assigned destinations to each: others were waiting to check-in, others to board their planes, other going from the check-in to the boarding area, others entering the airport and progressing to the check-in counters and others having just disembarked from their aircraft and proceeding to the baggage claim and from there to the airport's exits. We consider this to be an average case for the following reason. The major issue that slows down the calculation is the calculation of all the interdependent forces (between agents and agents, and agents and objects). Although optimizations have been implemented to keep these calculations at a minimum (there is no need to consider forces with objects or agents further than a certain distance) there exist situations where all these tests and calculations have to be performed. These situations generally include complex floor-plan geometries with many walls and obstacles and small areas where agents segregate. On the other hand, large rectangular areas without many obstacles and walls are easier since only a few agents are close to an agent in any given time and even fewer walls or obstacles. In our case, the airport has some large areas where agents can spread out as well as some narrow corridors where agents squeeze together by necessity. Naturally, as the number of agents increase (and the airport stays the same) the situation departs from an average case (AIA normally services 3000–5000 passengers at any given time and we reached up to 9000–10000 agents) at least from an everyday operational point-of-view.

In its current form, the simulator is capable of simulating a little over 9000 agents in real-time before hitting the 33.3 ms per timestep barrier which drops it below 30 fps (see Table 1 and Fig. 2). The simulator's performance is further decreased as agents get squeezed together many agents have to be checked for repulsion forces. If we conducted our experiment in a larger area where the average number of people checked would have been considerably lower, we could have probably achieved over 10000 agents in real-time.

5. Qualitative simulation results

5.1. Density effect on pedestrian movement speed/ Fundamental diagram

Large scale experiments involving tracking the location of (non-volunteering) pedestrians are hard to organize and execute from both technical and ethical points of view and as such virtually non-existing. What does exist, however, is experiments using naive volunteers such as Jelić et al. (2012) where location-monitored pedestrians are told to walk inside an annulus-shaped "room". Unfortunately, the experiment in Jelić et al. (2012) was one-dimensional, explicitly prohibiting overtaking and free movement in general. Nevertheless, it was as close to reality as one can get under the circumstances and some important observations regarding the density of the pedestrians and its relation to each pedestrian's speed were made. We imitated this experiment under full two-dimensional space (our room's dimensions were $r_1 = 24$ m and $r_2 = 28$ m) and unrestricted agent movement and tracked one specific agent regarding her maximum, minimum and average speed throughout a 100 s period as well as the average speed of the whole crowd. Our results can be seen in Fig. 3.

Our tracked agent had a maximum comfortable (when not in panic) speed of 1.9 m/s. For densities below 0.11 agents per square meter (about 1 agent/9 m^2) she more or less was traveling at about 1.4 m/s average, regardless of the density. When the density climbs over the threshold of 0.11 $\frac{\text{agents}}{m^2}$, we observe a slight decrease in the average agent's speed at a rate of about 1 $\frac{m}{s} / \frac{\text{agents}}{m^2}$ up until the density reaches about 0.21 $\frac{\text{agents}}{m^2}$ where the decrease suddenly becomes much steeper at a rate of 2.7 $\frac{m}{s} / \frac{\text{agents}}{m^2}$. This trend continues up to a density of 0.55 $\frac{\text{agents}}{m^2}$ where the average speed evens out at about 0.2 m/s. Our results are consistent with Jelić et al. (2012) and the three distinct zones they observed – even for our two dimensions – and seem to suggest that our model is adequate in capturing the essence of pedestrian movement in these kinds of situations (see Fig. 4).

5.2. Clustering/lane formation

In the same room used in (5.1) we placed 100, 200, 300 and 400 agents, half of them instructed to go round in a clockwise fashion and the other half in a counter-clockwise. We used the same method to track clusters of crowds as in Moussaid et al. (2012): assume agents α_0 and α_1; then α_0 belongs in the same cluster as α_1 if α_0 passes on time t closer than δ from the position α_1 was on $t - \tau$:

$$\|\vec{r}_{\alpha_0}(t) - \vec{r}_{\alpha_1}(t-\tau)\| \leqslant \delta \tag{18}$$

Our observations were consistent with the experiments with actual pedestrians in Moussaid et al. (2012) where they observed a high number of clusters at the beginning that rapidly decreases to a state of higher organization which is characterized with alternating periods of higher and lower organization levels (see Fig. 8).

Figs. 5 and 6 show the radial distribution of clockwise and counter-clockwise traveling agents over the 100-s period.

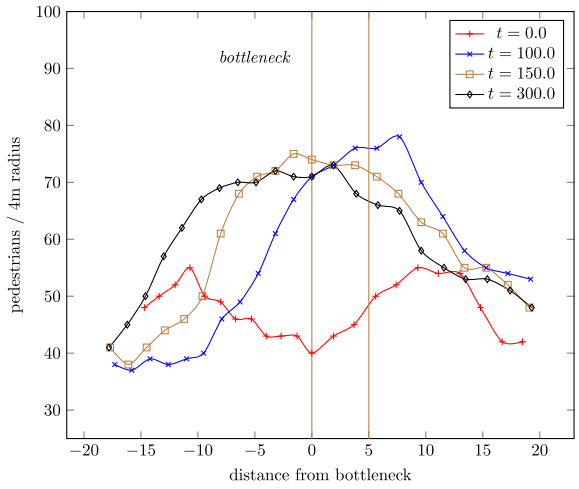
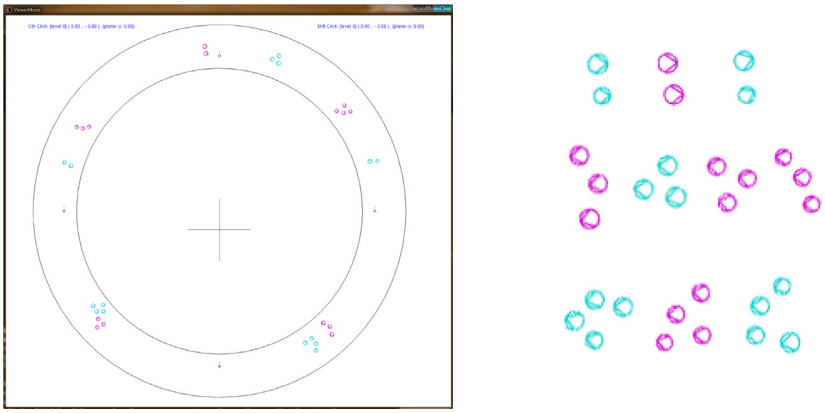


Fig. 7. Evolution of the pedestrian density distribution near a bottleneck. As the time progresses, the density peaks farther away.



(a) Snapshot of the simulation with groups (b) Ordering of groups with respect to size.

Fig. 8. Snapshot and grouping of agent groups with respect to the group's size. 2-agent groups' agents always travel side by side. 3-agent groups almost always walk in inverted 'V' but normal 'V' patterns are visible as agents 'recover' from interactions with other agents. 4-agent groups almost always walk into normal 'V' patters since the disturbance from other agents is bigger and need (but do not have) longer to recover.

5.3. Density and bottlenecks

To test the evolution of agent density near a bottleneck we place evenly agents along a bottlenecked corridor and run the simulation with the agents' target to pass through the bottleneck. The bottleneck has an average length of 5 m (represented by the two vertical lines in Fig. 7) starting at 0 m and extending up to +5m. At the beginning ($t = 0.0$ s), the density distribution is even at about 50 pedestrians in a 4 m-radius area. As the simulation progresses, we can clearly see the density building up on and right before the bottleneck at about 75 pedestrians per 4 m-radius area ($t = 100.0$ s), dropping immediately afterwards to 50. Even later and with no relief available, the maximum density starts to extend backwards ($t = 150.0$ s) an effect that gets stronger as time passes and people keep coming and pushing ($t = 300.0$ s).

5.4. Grouping behaviour

To test the grouping behaviour of our model, we used the same annulus-shaped room and placed 10 groups of agents (30 agents total), 5 walking clockwise and 5 counter-clockwise. The grouping was the following: 3 groups of 2 agents, 4 groups of 3 agents and 3 groups of 4 agents. All agents' personalities were neither strictly *follower* nor *leader* (and therefore had almost identical A_α in Eq. (13)). 2-agent groups almost always travelled side by side, perpendicular to the direction of movement. 3-agent groups generally tended to travel in 'V' shaped patterns mostly depending on their initial positioning and agent maximum comfortable speeds as well as on interactions from other agents. 4-agent groups received the biggest disturbance from interactions with other agents and as such their form is generally a squished inverted 'V'. Due to the attractive nature of the forces involved, the group travelled at speeds in-between the comfortable speeds of its members.

6. Conclusions

We have proposed a crowd behaviour model which can be efficiently implemented and can simulate independent agents, each with her own physiology, emotional state and strategic goal. Our model supports grouping behaviours such as families and friends and accounts for unexpected distractions in the form of time-dependent attractive behaviours. Our own implementation in its current state (without heavy optimizations as it is still under constant development) reaches easily over 10,000 agents at more than 30 frames or timesteps per second. Furthermore, testing shows that it exhibits realistic emergent crowd behaviour in terms of formation of crowds before bottlenecks and overflown service points and in terms of general space-filling distribution in case of a common target (e.g. evacuation).

We are currently working to incorporate steering behaviours (Reynolds, 1999) in our model which will help agents move with more foresight. Furthermore, we are working to incorporate support for dynamically placed obstacles such as debris from an explosion or agents who have fainted or even died. One way to address this could be by re-calculating and updating the necessary flow-fields but we are looking into a more robust and reactive way.

Finally, as in all crowd behaviour models, validation against empirical data is of utmost importance. We are currently working on a testbed that will allow us to test and validate our model against experimental data in various crowd conditions. In the context of the project TASS,[1] a detailed 3D model has been built in cooperation with the Athens International Airport to allow the validation of our model against realistic crowd condition at airports taking into consideration all ethical issues related with data recording in such environments. These datasets not only will facilitate the validation and calibration of our movement model but provide invaluable insights on how our strategic and cognitive layers measure compared with the behaviour of real crowds.

References

Arikan, O., Chenney, S., Forsyth, D., 2001. Efficient Multi-agent Path Planning. In: Proceedings of the 2001 Eurographics Workshop on Animation and Simulation, Machester, UK, pp. 151–162.

Ball, G., Breese, J., 2000. Emotion and Personality in a Conversational Character. MIT Press, 189–219.

Bouvier, E., Cohen, E., Najman, L., 1997. From crowd simulations to airbag deployment: particle system, a new paradigm of simulation. Journal of Electronic Imaging, 94–107.

Braun, A., Musse, S.R., Oliveira, L., Bodmann, B., 2003. Modeling Individual behaviours in Crowd Simulation, vol. 143. IEEE Computer Society, Washington, DC, USA.

Braun, A., Bodmann, B.E.J., Musse, S.R., 2005. Simulating virtual crowds in emergency situations. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, pp. 244–252.

Conde, T., Thalmann, D., 2004. An artificial life environment for autonomous virtual agents with multi-sensorial and multi-perceptive features: research articles. Computer Animation and Virtual Worlds 15, 311–318.

Cucker, F., Smale, S., 2007. Emergent behaviour in flocks. IEEE Transactions on Automatic Control, 0018-9286 52 (5), 852–862.

[1] TASS: Total Airport Security System DOW: TASS-FP7-SEC-2010–241905.

Funge, J., Tu, X., Terzopoulos, D., 1999. Cognitive Modeling Knowledge, Reasoning and Planning for Intelligent Characters. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, SIGGRAPH '99, New York, pp. 28–38.

He, X., Chen, L., Zhu, Q., 2011. A Novel Method for Large Crowd Flow. In: Pan, Z., Cheok, A., Muller, W. (Eds.), Transactions on Edutainment VI. Vol 6758 of Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 67–78.

Helbing, D., Molnár, P., 1998. Social force model for pedestrian dynamics. Physical Review 51, 4282–4286.

Helbing, D., Farkas, I., Vicsek, T., 2000. Simulating dynamical features of escape panic. Letters to Nature, 487–490.

Hughes, R., 2002. A continuum theory for the flow of pedestrians. Transportation Research Part B: Methodological 36, 507–535.

Hughes, R.L., 2003. The flow of human crowds. Annual Review on Fluid Mechanics, 169–182.

Hy, R.L., Arrigoni, A., Bessiere, P., Lebeltel, O., 2004. Teaching Bayesian behaviours to video game characters. Robotics and Autonomous Systems (Elsevier) Special issue: Robot Learning from Demonstration 47, 177–185.

Jelić, A., Appert-Rolland, C., Lemercier, S., Pettré, J., 2012. Properties of pedestrians walking in line: fundamental diagrams. Physics Review E, Statistical, Nonlinear and Soft Matter Physics 85 (3–2).

Koh, W.L., Zhou, S., 2011. Modeling and simulation of pedestrian behaviours in crowded places. ACM Transactions on Modeling and Computer Simulation 21.

Kuffer, J.J.J., Latombe, J.C., 1999. Fast Synthetic Vision, Memory, and Learning Models for Virtual Humans. In: Computer Animation, 1999. Proceedings, pp. 118–127.

Luo, L., Zhou, S., Cai, W., Low, M.Y.H., Tian, F., Wang, Y., Xiao, X., Chen, D., 2008. Agent-based human behaviour modeling for crowd simulation. Computer Animation and Virtual Worlds, 271–281.

Minorsky, N., 1922. Directional stability of automatically steered bodies. Journal of American Society for Naval Engineering 34, 280–309.

Monzani, J.S., Thalmann, D., 2000. A sound propagation model for interagents communication. In: Proc. on VW. pp. 135–146.

Moussaid, M., Helbing, D., Theraulaz, G., 2011. How simple rules determine pedestrian behaviour and crowd disasters. In: Proceedings of the National Academy of Sciences. http://dx.doi.org/10.1073/pnas.1016507108.

Moussaid, M., Guillot, E.G., Moreau, M., Fehrenbach, J., Chabiron, O., Lemercier, S., Pettre, J., Appert-Rolland, C., Degond, P., Theraulaz, G., 2012. Traffic instabilities in self-organized pedestrian crowds. PLoS Computational Biology 8.

Narain, R., Golas, A., Curtis, S., Lin, M.C., 2009. Aggregate dynamics for dense crowd simulation. ACM Transactions on Graphics 28.

Noser, H., Renault, O., Thalmann, D., Thalmann, N.M., 1995. Navigation for digital actors based on synthetic vision, memory and learning. Computers and Graphics 19, 7–19.

Pan, X., Han, C.S., Dauber, K., Law, K.H., 2007. A multi-agent based framework for the simulation of human and social behaviours during emergency evacuations. AI & Society 22, 113–132.

Patil, S., van den Berg, J., Curtis, S., Lin, M.C., Manocha, D., 2011. Directing crowd simulations using navigation fields. IEEE Transactions on Visualization and Computer Graphics 17, 244–254.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Pelechano, N., Allbeck, J.M., Badler, N.I., 2007. Controlling individual agents in high-density crowd, simulation. In: Proceedings of the 2007 ACM SIGGRAPH/ Eurographics symposium on Computer animation. SCA '07. Eurographics Association, Aire-la-Ville, Switzerland, pp. 99–108.

Rao, A.S., Georgeff, M.P., 1992. An abstract architecture for rational agents. In: Nebel, B., Rich, C., Swartout, W. R., Morgan Kaufmann, KR. (Eds.), pp. 439–449.

Renault, O., Thalmann, N.M., Thalmann, D., 1990. A vision-based approach to behavioural animation. Journal of Visualization and Computer Animation 1, 18–21.

Reynolds, C., 1999. Steering behaviours for autonomous characters.

Sakuma, T., Mukai, T., Kuriyama, S., 2005. Psychological model for animating crowded pedestrians: virtual humans and social agents. Computer Animation and Virtual Worlds 16, 343–351.

Shao, W., Terzopoulos, D., 2007. Autonomous pedestrians, Graph. Models 69, 246–274.

Shendarkar, A., Vesudevan, K., Lee, S., Son, Y.-J., 2006. Crowd simulation for emergency response using BDI agent based on virtual reality. In: Proceedings of the 38th conference on Winter simulation. WSC '06. Winter Simulation Conference, pp. 545–553.

Treuille, A., Cooper, S., Popovic, Z., 2006. Continuum crowds. In: ACM SIGGRAPH 2006 Papers. SIGGRAPH '06. ACM, New York, NY, USA, pp. 1160–1168.

Yu, Q., Terzopoulos, D., 2007. A Decision Network Framework for the Behavioural Animation of Virtual Humans. In: Proceedings of the 2007 ACM SIGGRAPH/ Eurographics symposium on Computer animation. SCA '07. Eurographics Association, Aire-la-Ville, Switzerland, pp. 119–128.