



HAL
open science

Multi-Agent Simulation Collision Avoidance of Complex System

Mohammed Chenoufi, Fatima Bendella, Maroua Bouzid

► **To cite this version:**

Mohammed Chenoufi, Fatima Bendella, Maroua Bouzid. Multi-Agent Simulation Collision Avoidance of Complex System. *International Journal of Ambient Computing and Intelligence*, 2018, 9 (1), pp.43-59. 10.4018/ijaci.2018010103 . hal-01975416

HAL Id: hal-01975416

<https://hal.science/hal-01975416>

Submitted on 17 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-Agent Simulation Collision Avoidance of Complex System: Application to Evacuation Crowd Behavior

Mohammed Chennoufi, Department of Computer Science, University of Science and Technology of Oran, Oran, Algeria
Fatima Bendella, Department of Computer Science, University of Science and Technology of Oran, Oran, Algeria
Maroua Bouzid, University of Caen, Normandy, France

ABSTRACT

In this work, we present a collision avoidance technique for a crowd robust navigation of individuals in evacuation which is a good example of a complex system. The proposed algorithm is inspired from the Reynolds model, with the addition of several individuals' behavioral criteria as well as a microscopic perception of the environment, which affects their travel speeds and emerging appeared phenomena. Our system is modeled by agent and tested by a Netlogo simulation, several modules such as A* planning, physical and psychological factors of agents have been programmed and successfully inserted into a 3D environment. Our application can be used as a framework to simulate real situations (evacuation of a stadium, a building...) in order to arrive at strategies to decision support of a complex system, which is a real problem in our daily life.

KEYWORDS

A*, Agents, Behavior, Collision Avoidance, Crowd, Emergence, Obstacle, Path

INTRODUCTION

Our work concerns the microscopic level of a complex system and more precisely the intercalations between crowd individuals in evacuation. The collision avoidance is a primordial phase in the interaction individuals-environment, whether the avoidance of static (wall) or dynamic obstacles (neighborhood individuals).

We distinguish several microscopic models of collision avoidance such as social forces (Helbing, Farkas, & Vicsek, 2000), rule-based models (Reynolds, 1987) and cellular automata models (Chenney, 2004), (Suwais, 2014). The difference between them is in the discretization of space and time. A (virtual) social force is analogous to real force such as repulsive interaction, friction force, dissipation, solving motion Newton's equations for each individual. In the rule-based model, the displacement of the crowd is governed by behavioral rules of the form "if condition then action". Recently there has been a lot of work on crowds and their behavior as models that check the characteristics of complex systems (Reynolds, 1987; Lamarche & Donikian, 2004).

The first work of Reynolds (1987) on the concept of flocking describes the behavior of the units individually as a group using only local rules, some year after, Reynolds (1999) introduces the notion of autonomy for each agent to find its way so as to avoid collisions. The disadvantage of this approach is that it operates on the basis of local information, putting individuals in congested environments. The work of Musse (2000) is to create rules for managing directly a set of information, as a (Kirchner,

DOI: 10.4018/IJACI.2018010103

Namazi, Nishinari, & Schadschneider, 2003) group of people operating in the environment. But this approach has the disadvantage of specializing the model, making again its generalization even more difficult.

However, in models of cellular automata, the space is represented by a grid of uniform cells; each cell to a local state which depends on a set of rules describes the behavior of individuals in which space and time are discrete (Chenney, 2004) and in behavioral domains we can cite the work of Hans and Marsland (2016), Roman, Nawaf, Darryl, and Abdallah, (2014) and Francesca (2016).

Musse and Thalmann (1997) focused on basic collision handling by proposing two techniques of collision avoidance. The first involves intersection of two lines and distance between two points in order to detect possible collision events. If two virtual humans are potentially colliding, only one will be allowed to go on first with its path. The second method is straight forward and it depends on the change of directions. An intelligent virtual human can avoid the collision by changing its directions through angular changes. Leitão, Vinhas, Machado, and Câmara (2014) proposed a genetic algorithm with two scenarios for inverse shortest path length problems.

The work of Foudil, Noureddine, Sanza, and Duthen (2009) inspired by the Reynolds model (1987) considers three types of collision between two agents:

- **Front collision:** Occurs if agents move towards each other;
- **Away collision:** When the agent is behind another agent;
- **Side collision:** Occurs if two agents walk almost in the same direction.

Another collision technique that we can use for crowd system is proposed by Loscos, Marchal, and Meyer (2003). This technique presents collision detection between avatar and other objects (such as building). The strategy is to use collision map and grid system. The technique outlines three types of collision strategies which are frontal, following and perpendicular. The technique compares the direction of each agent, the velocity factor and the distance between the agents. In order to deviate from an appropriate angle, there are a few ways to decide either to slow down or to completely stop.

Haifa, Ayesh, and Daniel (2012) added emotions as cognitive characteristics of agents to the behaviors of crowds and cellular automata models for collision avoidance. Silva, Urbano, and Lyhne (2014) propose and evaluate a novel approach to the online synthesis of neural controllers for autonomous robots. The work of Stephane, Gaud, Alves, and Koukam, (2013) presents a new model of collision avoidance allowing the design of realistic and effective virtual behaviors between pedestrians and cyclists. It's based on a sliding force to allow gentle avoidance of potential collisions while allowing the pedestrian to continue to progress towards his goal with the use of dynamic time windows to predict future potential collisions (principle of least effort). Hughes, Ondrej, and Dingliana (2014) presented a holonomic collision avoidance algorithm for crowd simulation based on experimental data, which allowed us to observe both the conditions under which holonomic interactions, as well as the strategies that walkers use during these interactions to avoid the collision, the main disadvantage is at the level of the discretization of time and the dynamic obstacles. Narang, Best, Curtis, and Manocha (2015) proposed a crowd simulation algorithm based on density filters which depend on the sensitivity of the local planner at the preferred speed to generate human-like crowd flows which generates pedestrian trajectories and which present the speed-density relationships expressed by the fundamental diagram. This approach is based on biomechanical principles and psychological factors. The fact that adaptation is done at the local level implies that density filters may prove ineffective in scenarios where navigation techniques dependent on global density are more appropriate.

Knowing that these complex systems possess a nonlinear and an unstable behavior during their executions hence their modeling is difficult at a higher level, thus the simulation remains an efficient way to test the proper functioning of the system and see the emergence at the macroscopic level by simple interactions between individuals. Each agent has a simple behavior and collectively, agents can accomplish a complex task whose goal is not to switch to chaos.

We proposed a collision avoidance algorithm inspired by the Reynolds model (1987), it's based on simple rules with the addition of several criteria. After seeing to plan the way of our individuals' crowd towards the arrival destination by avoiding the static obstacles by a simple A*, we launch our simulation of crowd movement by activating the module of collision avoidance in real time, the latter is based on a local perception of the environment (Angle of vision, comfort distance, density) and a speed of movement (desired speed). Each individual in the course of his or her path, he realizes one of these interactions to follow, flee or avoids a dynamic collision if it exists by decreasing or increasing its speed and or by a slight deviation to the right or to the left then a return towards its planning. This one will influence instantly on its initial trajectory. Our simulation works even if the crowd is dense; we have added physiological factors as age that has a direct relation with velocity. Activation and deactivation of the collision avoidance module occurs during the simulation for a good visualization of the collision.

The rest of the article is organized as follows: the first section describes complex systems while a collision avoidance model is presented in the second section. The third section presents a simulation of our model with discussions of emerging phenomena. A comparative study is presented in fourth section. Finally, we end with a conclusion and perspectives.

COMPLEX SYSTEM

Research in the complex systems is becoming increasingly important; It presents itself everywhere in our world and in several disciplines whether in biology, economics, chemistry, physics, transportation, internet, security... even human society is complex. However, there is no single definition of a complex system. As a first approximation, complex systems are systems composed of interacting entities with capacity to evolve over time; they adopt a non-linear dynamic behavior. A complex system can be defined as a large number of interacting components allowing the system to restructure or modify the pattern of interaction between its components. It is impossible to predict its evolution and its future behavior by a simple calculation with a possibility of switching to chaos. Even if the interactions between these components are simple; there emerges a global behavior that was not described from these rules of interaction emerges. In order to will predict this behavior well and to know the evolution of the system, simulation and experiments are primordial (Bar-yam, 1997). Figure 1 shows two examples of complex systems in two different levels.

Two levels are distinguished in complex systems:

- A microscopic level (low level) which represents the local properties between the, components of the system;
- A macroscopic level (high level) which represents the whole system with emergence of the new properties.

Studying interactions is not easy because they can be direct, indirect, with or without feedback loops. In the case of direct interactions, it is difficult to follow these links over time between the entities of the system, imagine if we add the indirect case as well as the dynamics of the environment which becomes more and more complex and difficult. Different levels of interaction observation exist: the interactions between systems of the same level and the interactions between systems of different levels.

In the literature, several scientists define the concept of emergence as being the whole is more than the sum of the parts. This concept is found in the philosophy of science and in other complex adaptive systems such as the neural system, ant colonies (Vijver, 1997; Prokopenko & Wang, 2004).

Another approach to emergence involves the concept of causality descending. A characteristic is emerging if it has a kind of causal power on lower level entities. While we assume that these (lower level) entities must have an upward causality on emerging characteristics, this approach assumes a two-way causal relationship (Couture, 2007).

Figure 1. Example of complex systems: a) Transport networks, b) Emergency evacuation



Characteristics of Chaotic Systems

Chaos theory has applications in many fields, including networking, large data analyzes, fuzzy logic, game theory and systemic thinking (Radwan, Kamel, Mohammed, & Hassanien, 2015). It is a domain of deterministic dynamics proposing that random events may result from normal equations because of the complexity of the systems involved.

The main characteristics of chaotic systems are:

- Chaos results from a deterministic process;
- It occurs only in non-linear systems;
- It occurs in retroactive systems; where events of the past affect current events and current events affect events of the future;
- The details of the chaotic behavior are hypersensitive to the change of the initial conditions;
- It can occur from relatively simple systems: - with discrete time;
- Information on initial conditions is irretrievably lost;
- It is not yet possible to determine in advance the particular path that the dynamic process will follow in order to move towards chaos (Williams, 1997).

PROPOSED APPROACH

This section describes the modeling of our approach in three steps:

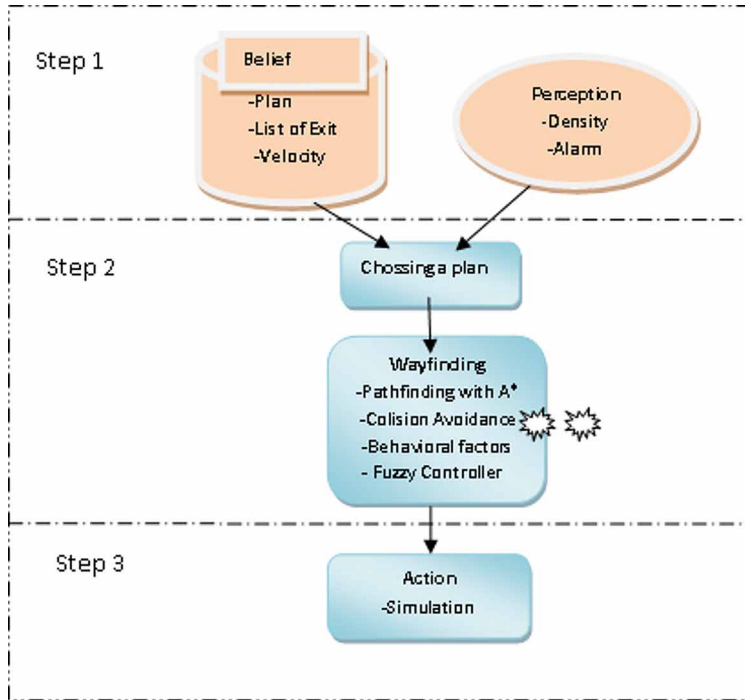
1. Step 1 is paramount in the navigation process; it contains the characteristics of BDI agents (perceptions and beliefs);
2. Our contribution is in step 2 more precisely to make a hybridization between the A* (see Algorithm 1), our collision avoidance model and behavioral factors which avoids real-time collisions;
3. Step 3 contains the simulation part which is an efficient way to validate a nonlinear system by means of several experiments.

Mathematical Modeling of The Problem

Each agent is represented by a circle (turtle in Netlogo) (see Figure 3) with:

- Position $A_i(x, y)$: where x and y are the coordinates of the agent A_i in the virtual space;
- $s_{init} \in S$ is the initial state (starting node) for each A_i with $S = \{s_1, s_2, \dots, s_i\}$;

Figure 2. BDI architecture of a cognitive agent



- $s_{Exit} \in \mathcal{S}$ is the final state (Exit node) for all A_i ;
- $h_s \in \mathcal{S}$ (Manhattan distance) is the heuristic function which computes the approximate distance from the current state to the goal state defined as:

$$h_{(s)} = |x_s - x_s'| + |y_s - y_s'|$$

For first A* algorithm we use the formula:

$$f(x) = g(x) + h(x) + \delta$$

which is the current approximation of the shortest path to the goal, where:

- $g(x)$ is the total distance between the initial position to the current position;
- $g(As_{init})$ is the minimum cost of the agent A_i ;
- $h(As_{init})$ is the minimum heuristic of the same agent;
- Open List is an ordered list of states; the search has been generated but has not been expanded yet;
- Closed List is the set of states; the search has been expanded (it is used to prevent re-expansion);
- h_i is the heuristic for agent A_i to exit.

The congestion value δ to know if the closest part to the path of the agent is free or not and to guide the agent in the least populated areas is defined in our algorithm according to the following function:

$$\delta = \left\{ \begin{array}{lll} \text{infini} & \text{if} & NA_i \geq \max \\ NA_i - \varphi & \text{if} & \varphi \leq NA_i \leq \max \\ 0 & \text{if} & NA_i < \varphi \end{array} \right\}$$

where:

- NA_i is the number of individuals present in a comfort zone of an agent A_i ;
- \max is the Maximum number of people occupying a comfort and a risk zone;
- φ is the is the threshold of congestion, we opted for one half of the \max ($1/2 * \max$);
- For the second algorithm we use;
- A comfort distance COD or d is the maximum distance between two agents to avoid collision;
- A risk distance $RID = COD / 2$: is the minimum distance between two agents to avoid collision;
- Direction: depending on the direction of the agent A_i , it updates its position right or left to avoid collision;
- Radius of vision α : Custom domain of vision;
- Angle of deviation β : the agent is steering with angle β for deviation;
- Speed \vec{v} : the speed varies with the state of health and mobility, ranging from 0 (when the health or mobility = 0) at 4 m / s (when running in panic);
- The density D : Number of individuals in a comfort zone.

Collision Avoidance Model

The collision avoidance module presented and explained in this paper (see Algorithm 2) is seen as an important action in the overall planning simulation process of our agent-based shed in step 2 of Figure 2. This avoiding collision is a crucial phase in the paths of individuals, the detection of collisions will depend on the shape and size of fixed or dynamic obstacles. When making an attempt to avoid an obstacle, there are many decisions to make. The latter ones depend on the available paths, finding the best and the shortest path depends on the algorithms used.

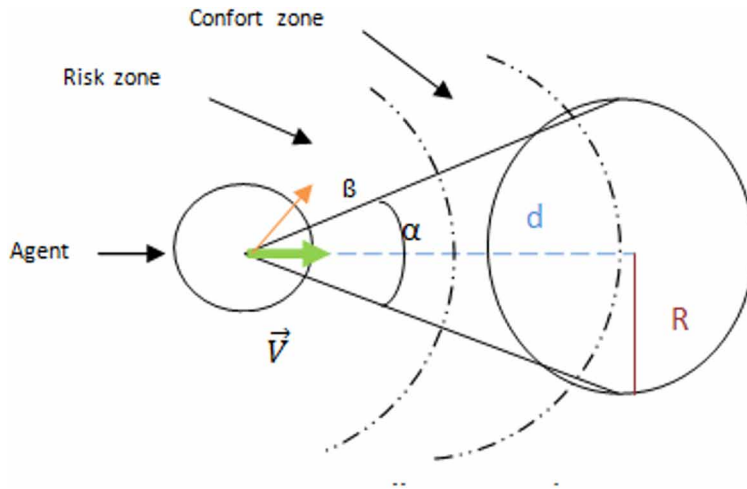
In order to eliminate any risk of collision with agents, we use the information obtained in perceiving the environment. In our approach we consider a single collision that encompasses all the cases in order to decrease the tests, it is based on a comfort distance d , Radius of vision α , Risk zone (DZ), Comfort zone (CZ), a desired speed $\overline{V_d}$, A permissible speed $\overline{V_{ad}}$, the density D (Number of individuals in comfort zone), Angle of deviation β and a radius of comfort $R = 2 * (d * \tan(\alpha / 2))$ (see Figure 3).

Algorithm 1: A*

```

Begin
    //Initialization;
    Current node = Starting node;
Repeat
```

Figure 3. Collision avoidance



```

    For each element  $s$  of the list of neighbors do
        If ( $s.state == free$ ) and  $s \notin$  closed list then
             $f(s) = g(s) + h(s) + \delta$ ; //  $\delta$  is a congestion
            factor
            End if
            If  $s \notin$  open list then
                Add the node to the open list;
                Put the current node as the parent of
                the node  $c$ ;
            Else
                If the new cost is better then
                    Update the open list;
                Update the parent of node;
                End if
            End if
        End for
        If the open list is empty then
            No solution;
        Else
            Current node = the best node in
            the open list;
            Remove the best node from the
            open list;
            Place it in the closed list;
        End if
        Until current node = destination node
    End.
    
```

Algorithm 2: Collision avoidance


```

Begin
  For each frame do
    For each agent  $A_i$  do
      Perception of the environment; //  $\alpha$  = Radius
      of vision &  $d$  = comfort distance
      If ( $\delta == 0$ ) //
      the path is free
        Then The agent follows his A* path;
      Else
        If ( $\delta == (NA_i - \varphi)$ ) //  $\exists$  Agents in  $R$  Individual
        detects another within its radius  $R$ 
          Then
            rt( $\beta$ ); // Turn right with a  $\beta$ 
            deviation (change direction)
            lt( $\beta/4$ ); // Turn left with a  $\beta/4$ 
            deviation (Instant return to its path)
            Wating; // If the distance between
            the agent and the expected position of the
            Collision is sufficient.
            
$$\overline{V}_d = \frac{\overline{V}_l}{D};$$
 // moving along A* path with
            desired speed = speed of individuals /
            density .
          Else
            
$$\overline{V}_{ad} = \left(\frac{\overline{V}_l}{D}\right) N_i + \xi;$$
 // moving with A* as a
            function of  $N_i$  chronological
            identification number of
            the crowd,  $\xi$  is threshold of speed  $\in [0 \ 1]$ 
          End if
        End if
      End for
    End for
  End.
    
```

Fuzzy Controller

Fuzzy Logic (Zimmermann, 2001) is ideal for modeling and controlling a crowd movement in a large scale for collision avoidance. This fuzzy is used to represent an imprecise value between true and false. The input variables of the regulator are transformed into linguistic variables with the definition of membership functions. Thus, this operation consists of determining the degree to a value to a fuzzy set as the work of Sarkar, Banerjee, and Hassanien (2015) and Deepak and John (2016). For custom vision domain, we observe two distances (COD: comfort distance), (RID: risk distance = COD /2) and an angle from vision α (SMV: small vision), (GRV: Great vision), see Figure 4.

Table 1 illustrates the fuzzifying for classifying real input variables into different sets, which are illustrated in Figure 5 a, b.

After that the rules have been evaluated, we use an aggregation centre-of-gravity function for velocity to merge the outputs. To obtain the real output defuzzified velocity, we take the membership

Figure 4. Illustration Fuzzy of collision avoidance with input distance and angle

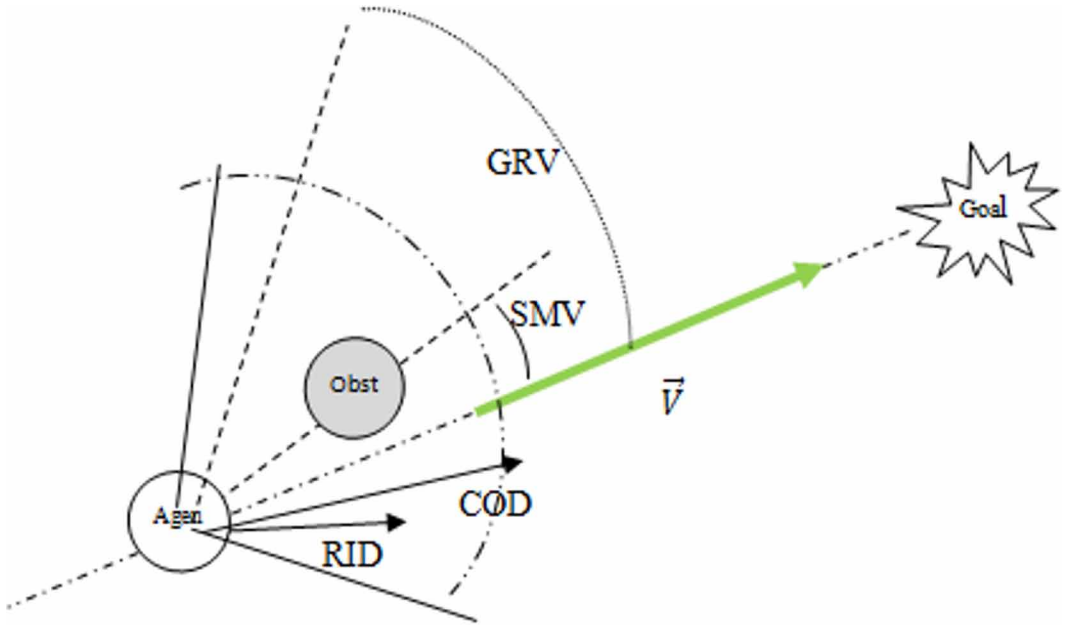


Table 1. Fuzzy input definitions for avoidance collision

Real	Fuzzy
Comfort distance	COD
Risk distance	RID
Small vision	SMV
Great vision	GRV

values (v) for each fuzzy output set, multiply each by its set centre value (v_i), and divide this by the sum of all of the centre values illustrated in Figure 5c:

$$v_{defuzz} = \frac{\sum_{i=1}^m (v * v_i)}{\sum_{i=1}^m v_i}$$

with m which is the number of rules and v , the control variable.

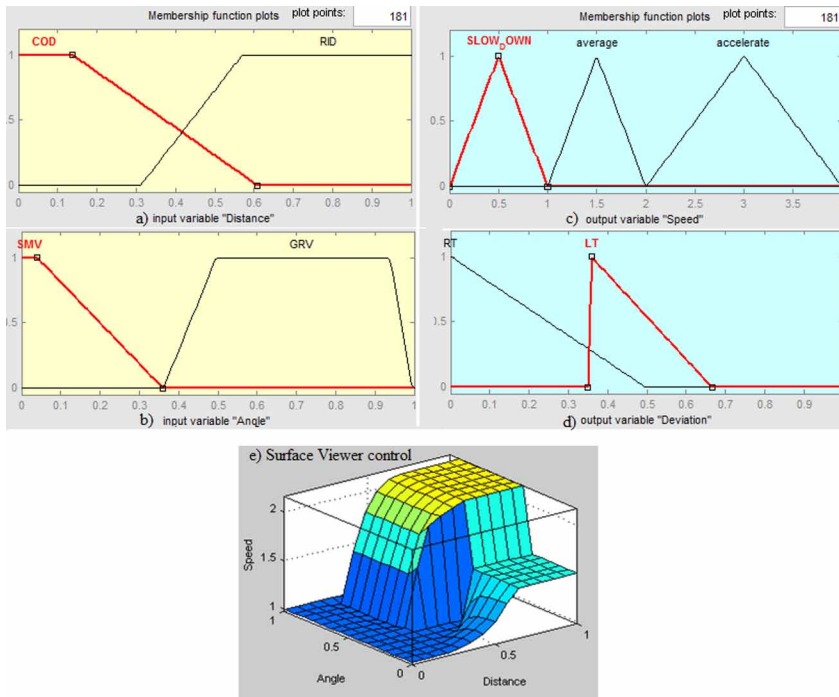
Similarly, the real output defuzzified Deviation is obtained by centre-of-gravity function.

The surface control in Figure 5e represents a series of rules if - then. The inference is based on min or max operations to make the rule inference and max operator for the aggregation of rules.

NETLOGO SIMULATIONS

Our Netlogo simulation (see Figure 6) is implemented in 3D real-time with an HP core I3, 2.40 GHz and 4 GB memory. The motivation to choose Netlogo as a simulation system in this work because it is very well adapted to the modeling of complex systems to explore the emergence due to interactions between agents, and it is easy to use with good documentation support.

Figure 5. Fuzzy input set membership functions for classifying the distance (a) and angle (b) to the nearest obstacle in Fuzzy terms, Fuzzy output value for obstacle avoidance desired speed (c) and deviation (d), surface viewer control (e)



Programming the speed of the agents (Acceleration and waiting) has a good relation with the other program class (A* algorithm), as well as fixed obstacles programmed by patches and dynamic or mobile agents by turtles. Another advantage of Netlogo is its flexibility to change the direction of an agent to right or left by a selected angle β with a simple Netlogo command “rt” or “lt.”

The virtual interface of our evacuation scene can be easily designed by the user, individuals are randomly dispatched, the user has the power to choose destination and locations of obstacles as well as change speed, size of agents and the density of the crowd. The simulation can be started by frame or direct and interrupted at any time. For a good visualization and comprehensions, the collision avoidance module can be activated and deactivated at any time.

Our application can be used as a framework to simulate real situations (evacuation of a building, a stadium, a super market, etc.) in order to arrive at strategies to decision support of a complex system.

Discussions

The figures in this section show the execution of several scenarios with and without collision avoidance:

- Random movement behavior of a crowd;
- Evacuation behavior of a crowd.

Scenario 1: Random movement behavior of a crowd.

Figure 7 shows a random movement of a crowd without collision avoidance to the left and with collision avoidance to the right where the agents are represented with chronologically numbered circles for distinguishing between the agents which are in collision (agent 28, agent 29, agent 14, agent 8) or not-collision (the agent 11 and 17).

Figure 6. Netlogo simulation

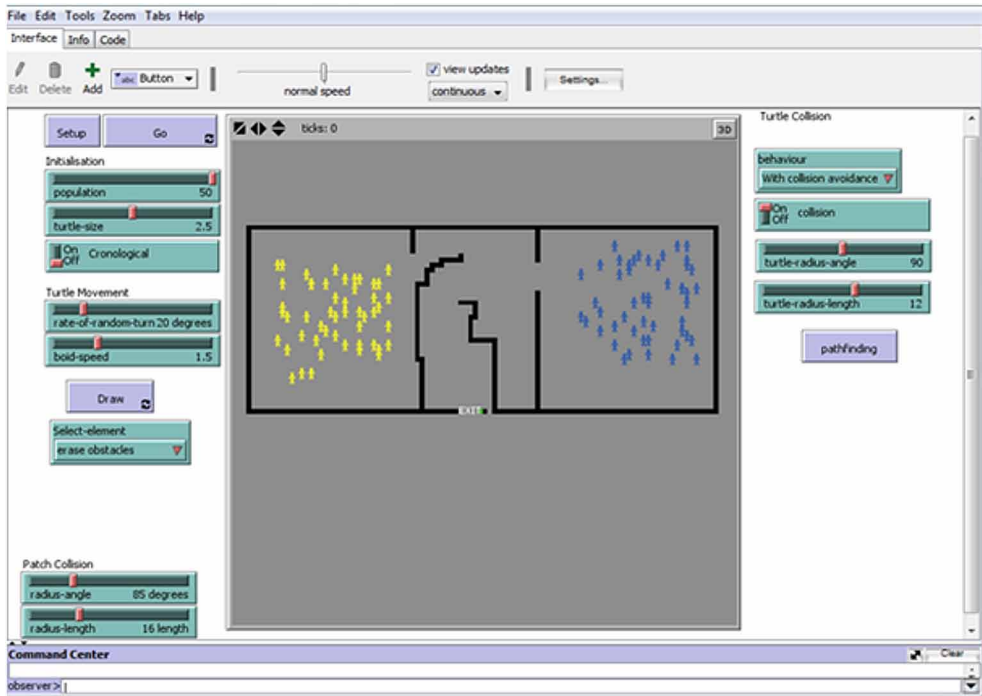
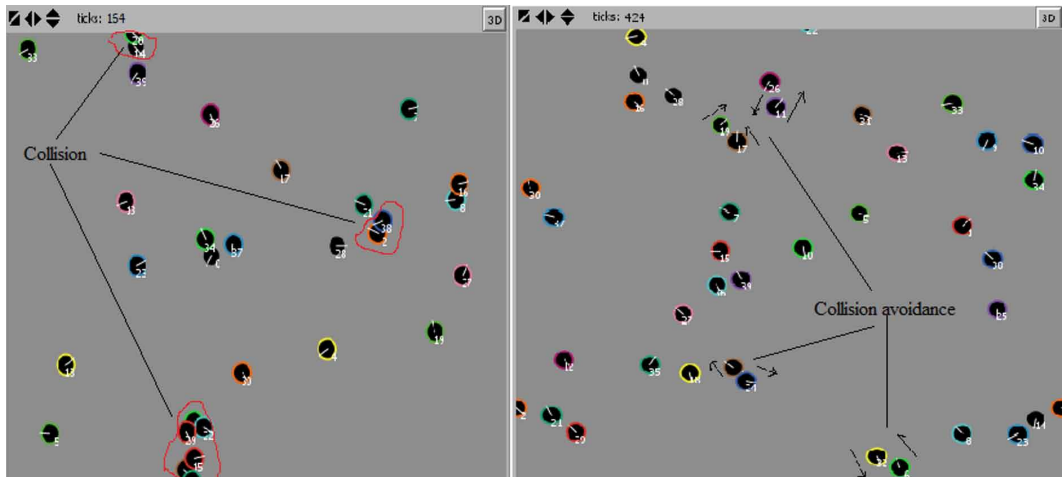


Figure 7. Crowd movements without collision avoidance in the left, and with collision avoidance in the right



Scenario 2: Evacuation behavior of a crowd.

The agents represented by persons in this second scenario are randomly divided into two different places, with blue color on the right and yellow on the left representing different behavior (ages, stress, panic.), the evacuation path is programmed with A* to the output located between two parts, and the collision avoidance module can be activated or deactivated in any time (see Figure 6).

This second scenario reproduces emerging phenomena as line and group formation by interactions between 100 people, the agent perceives its environment avoiding static and dynamic obstacles, it is these behaviors that generate the emergence of the complex system with activation collision avoidance module.

Figure 8 show a 3D view of crowd evacuation by deactivating the collision avoidance module where it is clear the occurrence of collision at the doors, contrary in Figure 9 representing our approach, we find appearance of people's voices by activating the module.

These emerging phenomena appeared at the exit when the collision avoidance module is deactivated and, contrary to the absence of any type of collision when the collision avoidance module is activated.

Figure 10 illustrates the relation between the density of the crowd and the evacuation time of our simulation scene, where the role of the collision avoidance module is clearly seen when the crowd is dense, for example from 100 agents during the time, the exhaust is at 300 measurement unit when the collision avoidance module is activated and 539 in the event that it is deactivated.

Temporal Complexity

In order to calculate the time of our algorithm in terms of relation of occurrence, I used a simple technique called incremented and counted by calculating the number of elementary instructions executed by the algorithm to solve the complex system, this complexity depends on a parameter n .

Our algorithm of collision avoidance uses the basic algorithm A^* to calculate the path to the output with the avoidance of static obstacles.

The *time complexity* of A^* for one agent depends on the heuristic function h , it meets the following condition:

Figure 8. Evacuation without collision avoidance in 3D

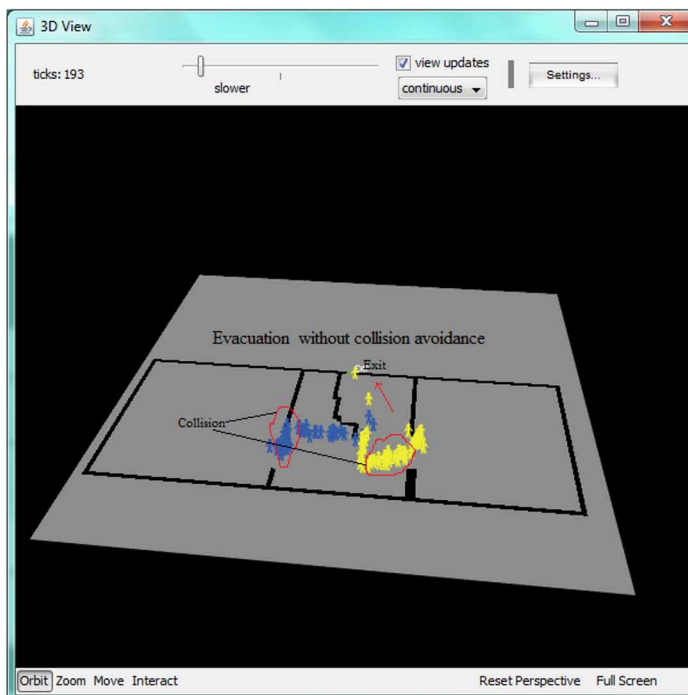


Figure 9. Evacuation with collision avoidance in 3D

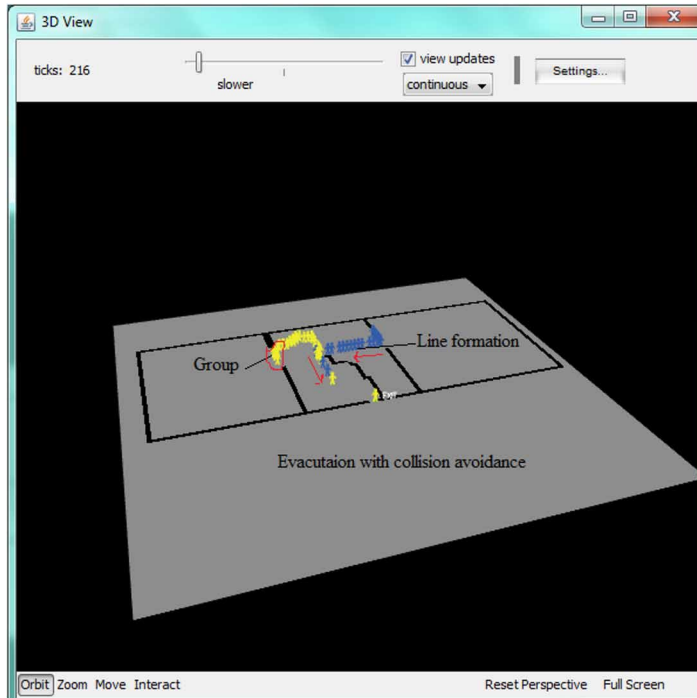
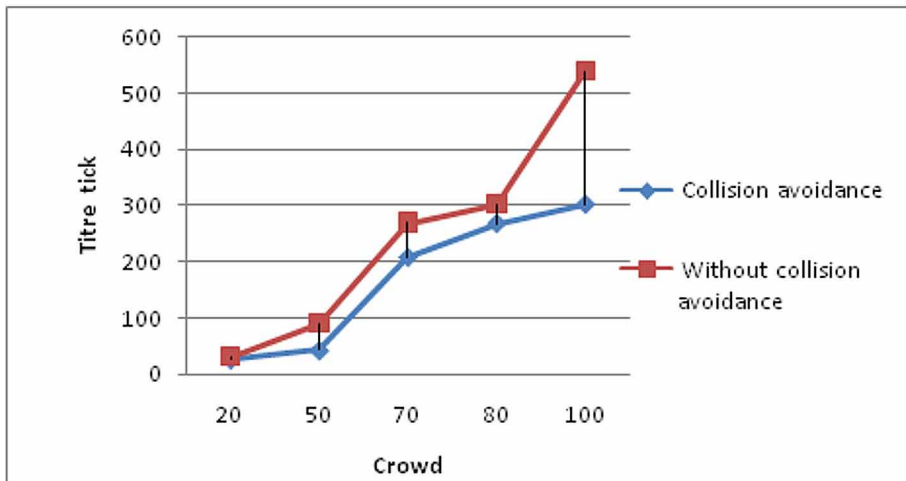


Figure 10. Execution time of crowd emergency



$$|h(n) - h'(n)| \leq O(\log h'(n))$$

where h is the estimated distance and h' is the optimal heuristic.

If cost of additions and comparisons are $O(1)$ complexity, the complexity of A* is:

$$O(n + m \log(n))$$

For a crowd of k agents we have a temporal complexity for dynamic collision avoidance in the worst case scenario given by the following approximation:

$$\begin{aligned} O(n + 1) &= (O(1) + O(1) + O(1) + O(1) + O(1) + O(n + m \log(n))) * k \\ &= (1 + 1 + 1 + 1 + 1 + O(n + m \log(n))) * k \\ &= (5 + O(n + m \log(n))) * k \\ &= 5 * k + k * O(n + m \log(n)) \end{aligned}$$

A COMPARATIVE STUDY

Table 2 shows a comparison between our approach and different works of previous years on crowd planning according to the following criteria: environmental modeling; time/Space; Collision avoidance; results and reviews.

Table 2. A comparative study between our approach and some research work on evacuation crowd planning

Author/ Year	Environmental Modeling	Time / Space	Collision Avoidance	Results	Reviews
Haifa et al. 2012	Hierarchical decomposition by cellular automata	Heterogeneous	Reynolds model	-Panic propagation -Queue line at the exit	-Absence of real model -No cognitive notion
Jocelyn et al 2013	Not discussed	Heterogeneous	Sliding force	-Predict future potential collisions	-Absence of dynamic environment
Rowan et al. 2014	Regular grid	Heterogeneous	Holonomic Interactions	Observation the conditions under which holonomic interactions occur	Level of the discretization of time and the dynamic obstacles
Sahil et al. 2015	Cell grid	Heterogeneous	-Density filters -Local planner	Generates pedestrian trajectories	Ineffective in scenarios with global density
Our approach	Quadtree Decomposition	Heterogeneous	Explained in section 2	-Appearance of emergent behaviors: queue line, group formation, arch formation at the level of each door	

CONCLUSION

Simulation of virtual crowds is not easy topic because we have to manage thousands of individuals. Some methods manage crowds as one entity. In this paper, we are rather interested in the microscopic simulation of the system. We have successfully implemented an effective rule-based collision avoidance module to minimize the maximum “if then else” tests, path planning is programmed by the A* as well as agent perception decision aiding when a problem has arisen. Emergence as an important feature of the complex system has become clear in addition our system has not switched to chaos.

We have used a hybrid architecture improved A* to compute evacuation paths of a crowd dispatched randomly in an environment, in order to guide them in the least populated areas by means of a congestion factor δ added to the heuristic (see Algorithm 1), and only a collision avoidance procedure (see Algorithm 2) instead of the 3 Reynolds procedures: separation, cohesion and alignment. If the agent perceives a dynamic obstacle in its near path, it deviates with a desired velocity and then returns to its path by inverse deviation.

In the near future, we are interested in the spatial part of the environment and thinking a new model of collision learning in order to manage new situations with GPU programming.

REFERENCES

- Bar-yam, Y. (1997). *Dynamics of Complex Systems*. Westview Press.
- Buisson, J., Galland, S., Gaud, N., Gonçalves, M., & Koukam, A. (2013). Real-time collision avoidance for pedestrian and bicyclist simulation: a smooth and predictive approach. In *Proceedings of the 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013)* (pp. 815-820).
- Chenney, S. (2004). flow tiles. [Grenoble, France: ACM SIGGRAPH.]. *SCA, 04*, 233–242.
- Couture, M. (2007). *Complexity and chaos - State-of-the-art Overview of theoretical concepts* (pp. 2006–2453). Canada: DRDC Valcartier TM.
- Deepak, D., & John, S. J. (2016). Information systems on hesitant fuzzy sets. *International Journal of Rough Sets and Data Analysis*, 3(1), 71–97.
- Foudil, C., Nouredine, D., Sanza, C., & Duthen, Y. (2009). Duthen2Path Finding and Collision Avoidance in Crowd Simulation. *Journal of Computing and Information Technology CIT*, 17(3), 217–228. doi:10.2498/cit.1000873
- Francesca, O. (2016). Technology studies and the sociological debate on monitoring of social interactions. *International Journal of Ambient Computing and Intelligence*, 7(1), 1-26.
- Haifa, A., Ayesh, A., & Daniel, O. (2012). Emotional based architecture for crowd simulation. *Journal of Intelligent Computing*, 3(2), 815–820.
- Hans, W. G., & Marsland, S. (2016). Using contextual information for recognising human behaviour. *International Journal of Ambient Computing and Intelligence*, 7(1), 27–44.
- Helbing, D., Farkas, I., & Vicsek, T. (2000). Simulating dynamical features of escape panic. *Nature*, 407(6803), 487–490. doi:10.1038/35035023 PMID:11028994
- Hughes, R., Ondrej, J., & Dingliana, J. (2014). Holonomic Collision Avoidance for Virtual Crowds. *SCA '14 Symposium on Computer Animation* (pp. 21 - 23). Switzerland: ACM SIGGRAPH/Eurographics.
- Kirchner, A., Namazi, A., Nishinari, K., & Schadschneider, A. (2003). Role of Conflicts in the Floor Field Cellular Automaton Model for Pedestrian Dynamics. In *Proceedings of the 2nd International Conference on Pedestrians and Evacuation Dynamics. (PED)*, London (pp. 51–62).
- Lamarche, F., & Donikian, S. (2004). Crowd of Virtual Humans: A New Approach for Real Time Navigation in Complex and Structured Environments. *Computer Graphics Forum*, 23(3), 509–518. doi:10.1111/j.1467-8659.2004.00782.x
- Leitão, A., Vinhas, A., Machado, P., & Câmara, F. (2014). A Genetic Algorithms Approach for Inverse Shortest Path Length Problems. *International Journal of Natural Computing Research*, 4(4), 36–54. doi:10.4018/ijncr.2014100103
- Loscos, C., Marchal, D., & Meyer, A. (2003). *Intuitive Crowd Behaviour in Dense Urban Environment Using Local Laws. In Theory and Practice of Computer Graphics (TPCG'03)*. Ipress Computer Society IEEE.
- Musse, S. R. (2000). *Human crowd modelling with various levels of behaviour control*. Unpublished doctoral dissertation, École Polytechnique Fédérale de Lausanne Suisse.
- Musse, S. R., & Thalmann, D. (1997). *A Model of Human Crowd Behavior: Group Inter-Relationship and Collision Detection Analysis. In Computer Animation and Simulation '97* (pp. 39–51). Budapest: Springer.
- Narang, S., Best, A., Curtis, S., & Manocha, D. (2015). Generating pedestrian trajectories consistent with the fundamental diagram based on physiological and psychological factors. *PLoS ONE*, 10(4), e0117856. doi:10.1371/journal.pone.0117856 PMID:25875932
- Odella, F. (2016). Technology studies and the sociological debate on monitoring of social interactions. *International Journal of Ambient Computing and Intelligence*, 7(1), 1–26. doi:10.4018/IJACI.2016010101

- Prokopenko, M., & Wang, P. (2004). On Self-referential Shape Replication in Robust Aerospace Vehicles. In *Proceedings of the 9th International Conference on the Simulation and Synthesis of Living Systems (ALIFE9)*, Boston (pp. 27-32).
- Radhwan, A., Kamel, M., Dahab, M. Y., & Hassanien, A. E. (2015). Forecasting exchange rates: a chaos-based regression approach. *International Journal of Rough Sets and Data Analysis*, 2(1), 38–57. doi:10.4018/ijrdsda.2015010103
- Reynolds, C. W. (1987). Flocks, Herds, and Schools: A Distributed Behavioral Model. *SIGGRAPH*, 87(4), 25–34. doi:10.1145/37402.37406
- Reynolds, C. W. (1999). Steering behaviors for autonomous characters. In Proceedings of Game Developers conference, San Jose, CA (pp. 763-782).
- Roman, V. Y., Nawaf, A., Darryl, D., & Abdallah, A. M. (2014). Behavioral biometrics: categorization and review. *International Journal of Natural Computing Research*, 4(3), 85–118.
- Sarkar, M., Banerjee, S., & Hassanien, A. E. (2015). Evaluating the degree of trust under context sensitive relational database hierarchy using hybrid intelligent approach. *International Journal of Rough Sets and Data Analysis*, 2(1), 1–21. doi:10.4018/ijrdsda.2015010101
- Silva, F., Urbano, P., & Lyhne, A. (2014). Online Evolution of Adaptive Robot Behaviour. *International Journal of Natural Computing Research*, 4(2), 59–77. doi:10.4018/ijncr.2014040104
- Suwais, K. (2014). Assessing the Utilization of Automata in Representing Players Behaviors in Game Theory. *International Journal of Ambient Computing and Intelligence*, 6(2), 1–14. doi:10.4018/IJACI.2014070101
- Vijver, G. V. (1997). Emergence et explication. *Intellectica*, 25(2), 7–23.
- Williams, G. P. (1997). *Chaos Theory Tamed*. USA: A Joseph Henry Press book.
- Zimmermann, H.-J. (2001). *Fuzzy Set Theory—and Its Applications*. New York: Springer Netherlands. doi:10.1007/978-94-010-0646-0

Mohammed Chennoufi received an engineer diploma in 2007 and a Magister degree in Computer Science from the University of Science and Technology of Oran 'Mohamed Boudiaf' in 2012. He teaches at the University of Oran 2 and his main research concerns are in the areas of artificial intelligence, Multi-agents system, Complex system and crowd behavior.

Fatima Bendella received an engineer diploma in Informatics from the University of Oran and a PhD in Computer Science from the University of Science and Technology of Oran, Algeria in 2005. She has been involved in research of knowledge management systems, learning environments, collaborative learning, ontology's and embodied agents. She has also investigated the domain of serious games for medical domain. She is currently a Professor in the department of computer science at the University of Sciences and Technology of Oran Mohamed Boudiaf.

Maroua Bouzid received an engineer diploma in Computer Science from the University of Constantine (Algeria) in 1990 and her M.Sc. degree from the University of Nancy 1 (France) in 1991 as well as her PhD degree in 1995 in Temporal Reasoning. She obtained the HDR degree (habilitation to supervise research) from the university of Caen (France) in 2006 in Spatio-Temporal Reasoning. From 1996 to 2002, she was Associate Professor at the University of Artois in Lens (France) and from 2002 to 2009, she was Associate Professor at the University of Caen. Since 2009, she is Professor at the University of Caen in the Computer Science Department.