

# A multipotential field model for crowds with scalable behaviors

Teófilo B. Dutra\*, Joaquim B. Cavalcante-Neto\*, Creto A. Vidal\* and Soraia R. Musse†

\*Department of Computing, Universidade Federal do Ceará, Fortaleza, CE, Brazil

Emails: {teofilobd, joaquimb, cvidal}@lia.ufc.br

†Computer Science Department, Pontifícia Universidade Católica do Rio Grande do Sul, RS, Brazil

Email: soraia.musse@puccrs.br

**Abstract**—Computer simulation of realistic crowd behavior has been the focus of active research for more than two decades now. In crowd simulation, there is usually a trade-off between performance and realistic crowd behavior. In this paper, we propose a model, based on potential fields, that enables the introduction of many behaviors in crowd simulations, while keeping good performance. The model uses multiple groups to guide agents to various different goals in the environment, and combines potential fields and reciprocal velocity obstacles (RVO) approaches, where the first sets the preferred velocities of the agents according to their current goals, whereas the second makes the agents avoid collisions. We used three scenarios to demonstrate the capabilities of our model for simulating crowds in which the agents present greater variety of behaviors in real-time without using a complex architecture.

**Keywords**—potential fields; reciprocal velocity obstacles; scalable behaviors; crowd simulation;

## I. INTRODUCTION

Crowd simulation has become the focus of interest in many areas, such as entertainment, building analysis, crowd evacuation studies during emergency situations, and collaborative virtual environments. In most applications, crowd behavior needs to be simulated realistically. Thus, even though, in a crowd, individuals present certain crowd behaviors, they still try to follow their own goals and desires. Also, it is necessary to take into account physical restrictions that prevent individuals from passing through one another or through solid objects. When the number of individuals in a simulation (agents) is low, all those factors are easier to account for and the simulation can still run in real-time. However, for large crowds, realism and real-time compete for the computational resources, and, many times, render the simulation infeasible.

Currently, it is still computationally very costly to efficiently treat a large crowd simulation by considering it as the union of the individual trajectories of all agents as they pursue their individual goals and desires in the simulated environment. A well-known approach to deal with large crowds is the use of potential fields [1], [2] to guide agents with common characteristics (including goals) to their respective goals, as Treuille et al. propose in [3]. Potential fields are effective to guide crowds with thousands of agents in real-time, but only when a few number of fields is used (what means a few number of goals in the simulation), since the computation of

potential fields is costly (the cost is proportional to the field's number of cells).

The major contribution of this work is the use of multiple potential fields to increase the number of goals in a simulation, while keeping the computational cost affordable by precomputing those fields. Our model allows the introduction of local goals through the use of local groups (called secondary groups), which enable momentary changes of the agents' main goals. Another contribution of this work is the combination of potential fields and reciprocal velocity obstacles (RVO) [4] approaches in order to handle collision avoidance. In our model, potential fields are responsible for global navigation, helping agents to avoid collisions with the scenario by providing them the preferred velocities, whereas RVO treats the collisions between agents and also prevents them from entering areas occupied by obstacles.

The remainder of this paper is organized as follows. In Section II, we discuss the most relevant related work. In Section III, we present our multipotential field model. In Section IV, we present the tests that were devised to demonstrate the usefulness of our technique, and discuss the respective results. Finally, in Section V, we draw some conclusions.

## II. RELATED WORK

Collective behaviors have been studied since the late 19th century, but only at the end of the 80s did researchers begin to simulate those types of behaviors with computers [5]. As a pioneer, in 1987, Reynolds [6] presented an agent-based model for reproducing flock behavior through the use of three simple rules. Since then, many approaches have been proposed to solve the varied problems of this multidisciplinary field.

In the following, we cite just a few relevant works on crowd behavior that are based on different approaches: rules [6], [7], [8], [9], [10], particles [11], [12], fluids [3], [13], guidance fields [14], [15], [16], [17], [18], geometry [4], [19], examples [20], synthetic vision [21], space colonization [22], and principle of least-effort [23], [24]. Some of those approaches have been combined to produce models that are capable of using the advantages of each approach when necessary. For example, Pelechano et al. [25] combined a particles-based model with a rule-based model that uses psychological rules; Yersin et al. [26] divided the environment in regions of interest, each of which is ruled by a different approach; Xiong et al.

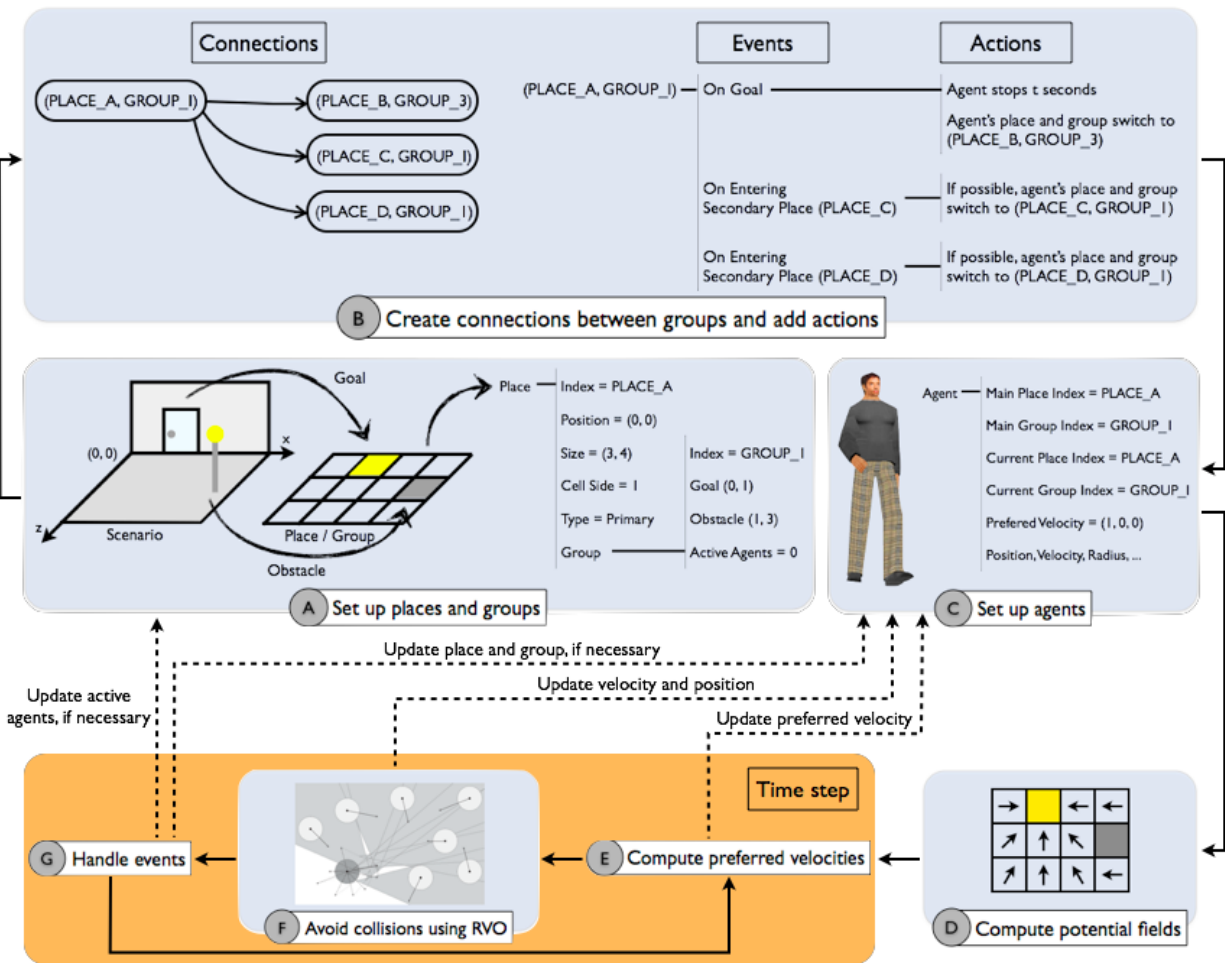


Fig. 1. Overview of the multipotential field model.

[27] proposed a way in which two models coexist and work collaboratively during the crowd simulation; Narain et al. [28] simulated discrete agents in a continuous flow; and Singh et al. [29] used multiple steering approaches according to an agent's current situation.

Currently, two approaches have been widely used in games, mainly because of their suitability for reproducing crowd behavior in real-time. The first approach is fluid based and was introduced by Treuille, Cooper and Popović [3] as an improvement to Hughes's work [30] to achieve more realistic crowd behaviors by grouping the agents in the simulation according to common goals. Each group is represented as a grid of cells, for which a potential field that guides the agents in the group towards their goal is computed. This model unifies the methodologies of global and local path planning with those of collision avoidance. Its major advantage is the ability to simulate the behavior of many agents in real-time realistically. However, real-time simulation can be obtained only if the number of groups is small, because the potential fields are computed for each time step, and that leads the agents to lose their individuality. The second approach is geometrical based and arose from the concept of Velocity Obstacle established

in the field of robotics by Fiorini and Shiller [31], where a robot is able to avoid collisions with moving obstacles, based on their velocities. In 2008, van den Berg et al. [4] extended the concept of Velocity Obstacle to account for the reactive behavior of the other agents, assuming that the agents avoid each other in the same manner, and that novel method was called Reciprocal Velocity Obstacle (RVO). The method was further optimized in [19] to become more robust, and the new concept resultant was called Optimal Reciprocal Collision Avoidance (ORCA). That approach can treat collisions among many agents and obstacles, locally, in a very efficient way, also presenting believable crowd behavior.

In this work, we put those two approaches to work together into a novel model, where precomputed potential fields are responsible for obstacle avoidance and for providing the preferred velocities of the agents to the RVO algorithm, which is in charge of moving the agents without collision among themselves efficiently. Narain et al. [28] also presented a model in which continuous fields are responsible for providing the preferred velocities for the agents without considering their neighbors, and, by employing a new mathematical constraint, the inter-agent separation in the continuous domain is en-

forced. The main drawback of that model is that the collision avoidance method employed only considers local information, i.e., future collision from distant agents cannot be anticipated. That model, despite its efficiency, only provide good results in dense situations.

Many crowd simulation methods focus on navigational issues in crowds represented as conglomerate of agents with global goals. However, crowd simulations that show believable behaviors are also important. For example, it is relevant to simulate the interaction between an agent and its environment, as well as the interpersonal relationships among agents. To confer believable behaviors to crowd simulation, methods such as behavior maps [32], composite agents [33], situation agents [34], transactional analysis [35], scalable situation-based behaviors [36] and data-driven methods [37], [38], [39], [20], [40] have been developed.

Aiming to provide heterogeneous behaviors to the agents in the simulation, in this work, we introduce the use of potential fields for helping the agents to achieve local goals. Those fields will be used to momentarily change the agent's main goal to a local goal, allowing the introduction of scalable situation-based behaviors.

### III. THE MULTIPOTENTIAL FIELD MODEL

In models based on potential fields, agents with the same characteristics are grouped, and a potential field is generated for each group. A potential field is defined over a regular grid, the cells of which store the field values, which are used to compute the velocities of agents in the group. The size of the grid is determined by the size of its cells. Thus, depending upon the size of the environment and upon the size of the cells, the number of cells in the grid can be huge, and, therefore, computing the potential field can become very costly.

In Treuille's work [3], the computation of a potential field depends on topographical and flow speeds. However, since the flow speed depends on the density of the crowd, which changes continuously, the potential fields need to be updated at every time step of the simulation.

In this work, we compute potential fields considering only topographical speed. Thus, potential fields need to be re-computed only when the grid changes, i.e., by adding or removing goals and obstacle cells. That decision led to a drastic reduction of the computational cost of a time step, since potential fields are no longer computed at each time step. Consequently, the spared computational power allows for more potential fields to be used in order to provide varied local goals in real-time simulations. Due to the use of multiple potential fields, we named the proposed model multipotential field model.

An agent, in our model, sets its preferred velocity according to the potential field where it stands, and that velocity is used as input to the RVO algorithm. Thus, RVO is responsible for treating the collisions among the agents and for giving them the best routes according to their positions, to their preferred velocities and to the local crowd density.

The developed model is shown, step by step, in Figure 1. In the next subsections, each of those steps is detailed (Step A in Subsection A, Step B in Subsection B, and so on).

#### A. Setting up places and groups

In this first step, the environment must be manually covered by places representing its walkable regions. A place represents a region of the environment and it is composed of one or more groups. Each group belonging to a place represents its region as a grid, in which its cells contain the information used to guide the agents in that group to their goal. Places exist to allow a better management of the groups that share the same characteristics (region, cell size, obstacles, etc.), except goals. Figure 2 shows a place with two groups that have different goals.

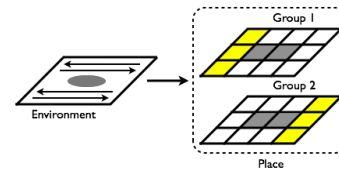


Fig. 2. An environment with an obstacle represented as an ellipse. For that environment and the expected flow (represented by the arrows), a place is created with two groups with different goals (yellow cells represent goals and gray cells represent obstacles).

Places are classified into two types: primary and secondary. Throughout the article, when referring to a group as primary or secondary, we mean that it belongs to a place of that type. A primary group guides agents to a main goal, whereas a secondary group guides agents to a local goal. In this work, a local behavior is represented as the movement to a local goal or to a sequence of local goals. Thus, secondary groups are used to provide local behaviors to agents through local goals. A local behavior consisting of a sequence of local goals can be provided by establishing connections between groups. For example, the goal of group *A* leads to group *B*, and the goal of group *B* leads to group *C*, and so on. Each group keeps the number of agents that are assigned to it at every time step, the so-called active agents of that group. That information can be used to limit the number of agents in secondary groups.

#### B. Creating connections between groups and adding actions

A connection is a simple association between two groups, regardless of the places to which they belong (they can belong to the same place or to two different places). However, two groups can be connected only if they have a transition area in common through which the agents can switch from one group to the other. The cooperation between groups is only possible if connections between them are established, i.e., a group needs to know the other groups with which it can interact when actions are triggered. Then, in this second step, those connections are manually created (an example can be seen in step B of Figure 1), and will be used by actions.

Two types of events trigger actions: the event of an agent entering a new group, and the event of an agent reaching a

goal. In the first type of event yet two cases are possible: the agent enters a primary group; or the agent enters a secondary group. In the first case, the triggered action is an immediate change of group, so that the agent’s current group is instantaneously changed for the new group it entered. In the second case, the triggered action is also a change of group. However, secondary groups do not allow a new agent to become instantaneously active. Thus, a decision has to be made as to whether or not the incoming agent will be accepted into the group. That decision can be the result of a random process, or can be the result of an analysis of the list of personal states of the agent with respect to the group’s requirements.

In the second type of event, when the agent reaches a goal, the triggered action(s) can be defined by the user, depending on the type of behavior that fits best a specific situation. Some of the actions can be simple, direct actions, such as: changing the group of an agent; making an agent to wait for  $s$  seconds; making an agent to look at a specific direction; removing an agent from the simulation; etc. Other actions can be a combination of simple actions that are specified at certain goals, such as: wait for  $s$  seconds and then change the agent’s group. A sequence of connected groups with actions on goals can be used to reproduce some scenarios, as shown in Table I. The secondary groups shown in Table I deviate the agents from their main goal, but, after performing the actions of those groups, the agents must return to their original groups that lead them to their main goals.

TABLE I  
EXAMPLES OF REPRODUCIBLE SCENARIOS.

Scenarios	Secondary groups	Action on the goals
Supermarket	Groups for things to be taken	Stop and get something
Museum	Groups for pictures to be observed	Stop and observe a picture
Square	Groups for benches, observable places, observable artists	Stop and sit; stop and observe a place; stop and observe an artist

Finally, potential fields (based on Treuille’s approach) do not produce local minima, but it is necessary to be careful when creating and connecting them to avoid unpleasant situations, such as secondary groups that do not let agents achieve their main goals.

### C. Setting up agents

An agent must have the necessary attributes to act in conformity with its groups. It has information about a primary group (which leads it to a main goal) and the group in which it is active at the moment (current group). It is necessary to keep the primary group, because after local interactions (by secondary groups) the agent must return to it. The group where an agent is active at a given moment determines the speed of that agent. However, that speed cannot exceed the agent’s maximum speed. Thus, agents with different speeds (children, elderly, etc.) can coexist within a group.

Also, an agent will have all the characteristics needed by the RVO algorithm (radius, velocity, preferred velocity, maximum speed, maximum neighbor distance, etc.). Moreover, the user can define personal states to the agents, and also can assign constraints to the secondary groups, such that an agent becomes active for a secondary group only if it respects the group’s constraints.

### D. Computing potential fields

At this step, for each group of each place, a potential field is computed in a similar manner as that proposed by Treuille et al. [3], but modified in such a way that the computation does not have to be performed at each time step (see discussion in the beginning of Section III).

The potential field approach proposed in [3] is derived from four hypothesis: 1) Each person tries to reach a geographic goal; 2) People move as fast as possible; 3) There is a discomfort field  $g$  that may change the path of people; and 4) A person must choose the shortest path that takes less time and avoids discomfort. Our modifications occur in the speed field computation and we also do not allow a dynamic discomfort field.

Mathematically, the fourth hypothesis means that, given the set  $\Pi$  of all paths from a person’s location  $x$  to some point in the fixed goal  $G$ , that person must choose the path  $P \in \Pi$  that minimizes

$$\underbrace{\alpha \int_P 1 ds}_{PathLength} + \underbrace{\beta \int_P 1 dt}_{Time} + \underbrace{\gamma \int_P g dt}_{Discomfort}, \quad (1)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  are weights, and  $g$  is a discomfort field. Assuming a fixed speed field  $f$ , and knowing that  $ds = f dt$ , equation (1) can be rewritten as

$$\int_P C ds, \quad (2)$$

where

$$C \equiv \frac{\alpha f + \beta + \gamma g}{f} \quad (3)$$

is the unit cost field. Our model does not allow a dynamic discomfort field, since the potential field is precomputed, but  $g$  remains in Equation 1 because it is still possible to define a static discomfort field.

The speed field  $f$  necessary to compute  $C$  measures the maximum speed allowed in each direction at each point, and is no longer dependent on the local crowd density, but only on the topography of the terrain. After computing  $f$  and  $C$ , we define a potential function  $\phi$  that, at a specific location  $x$ , represents the cost of reaching the goal through the optimal path beginning at  $x$  ( $\phi = 0$  at the goal location). Thus, at any location, a person should move in the negative direction of the local gradient of  $\phi$ . The function  $\phi$  satisfies the *eikonal equation*:  $\|\nabla\phi(x)\| = C$ . Finally, we compute the velocity field, multiplying the speed field by the normalized gradient of the potential field at corresponding locations. For a more detailed explanation of the potential and velocity fields’ computation, we refer the reader to Treuille’s work [3].

### E. Computing preferred velocities

After computing the fields, the main loop of the simulation starts, encompassing this and the next two steps. At this step, the agents' preferred velocities are computed, by interpolating the velocity field of their current group according to the agent's position in that field.

### F. Avoiding collisions

In our model, the preferred velocities provided by the potential fields prevent collision of the agents with static obstacles in the scenario, whereas the RVO algorithm prevents collisions of the agents with one another. That algorithm also needs the information concerning the static obstacles in order to impede that an agent that deviates from another agent trespasses the boundaries of an obstacle.

Thus, in this step, the RVO algorithm treats collisions among the agents, using the preferred velocities provided by the potential field as input, and provides collision-free velocities and positions of the agents. This algorithm was selected due to its efficiency and due to the visual results it provides.

The RVO algorithm is based on velocity, i.e., the new velocity of an agent is computed taking into account the velocities of the other agents that are inside its area of perception. The inherent interdependency involved in this type of computation is efficiently resolved through linear programming.

When dealing with collision avoidance between moving obstacles, it is always necessary to look for an efficient way of finding the neighbors of an agent. The RVO algorithm uses a k-d tree for performing a fast spatial search. Finally, the algorithm is natively parallel, which also increases its performance when many processors are available. For a detailed explanation of RVO, we refer the reader to van den Berg's work [4], [19].

### G. Handling events

In this last step, the agents are monitored to detect when they produce events that trigger corresponding actions. If an agent reaches the goal of its group, the actions associated with that event are performed. In the case of entering the area of another group, it will be necessary to analyze the agent's personal states and the group's restrictions in order to verify whether or not the agent can change its group for that new group. It is up to the user to define the agent's personal states and the group's restrictions. For example, an user may define that a group should have at most five agents at any instant, or that a specific agent is not allowed to enter a given group. Whenever an agent leaves a group and enters another, the number of agents of the groups involved must be updated as well as the agent's current group.

## IV. RESULTS AND DISCUSSION

The tests were performed with a MacBook of 2.13 GHz Intel Core 2 Duo processor, 4.0 GB of RAM and an NVidia GeForce 9400M graphics card with 256 MB of RAM. The algorithm was written in C/C++ using the open source high performance 3D graphics toolkit OpenSceneGraph [41]. A

model with some useful animations, available in the open source project ReplicantBody [42], was used.

The tests were performed to analyze both the performance of the model and the behavior of the agents. We begin analyzing the use of secondary groups to reproduce a scenario with varied local behaviors, and comparing the performances of simulations with precomputed potential fields to those in which potential fields are computed every time step. Then, we show that the secondary groups also can be used to organize agents in specific situations. Finally, an evacuation scenario is presented showing that the model can handle evacuation situations and also that groups can be connected to create large scenarios with varied groups. The presented results are related only to the simulation step (execution time in milliseconds).

### A. Results

Our simulations were performed in the following scenarios:

- **Art gallery.** This scenario was used to demonstrate the suitability of the model for simulating an environment with varied behaviors. In this scenario, eight secondary groups (six with 4 x 8 cells and two with 6 x 12 cells) were used to lead agents to specific points of the environment. For those groups, the following actions were defined to be performed on their goals: an agent turns to look at a pre-defined point, stops for five seconds and then returns to its main goal. The agents randomly become active for those secondary groups and only three agents can be active in a group at a time. Two primary groups with 16 x 6 cells were used to cover the entire environment to keep the agents walking in the room. Up to three hundred agents interact with the scenario. Each simulation was performed for five minutes. Results can be seen in figures 3, 4 and 5.

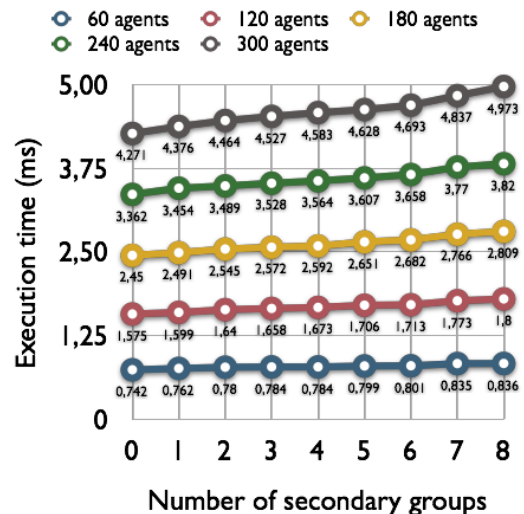


Fig. 3. Average execution time (ms) for simulations performed in the art gallery scenario. The graph shows that the addition of secondary groups barely affects the execution time of the model.

- **Metro entrance.** This scenario was used to demonstrate the suitability of the model for organizing the



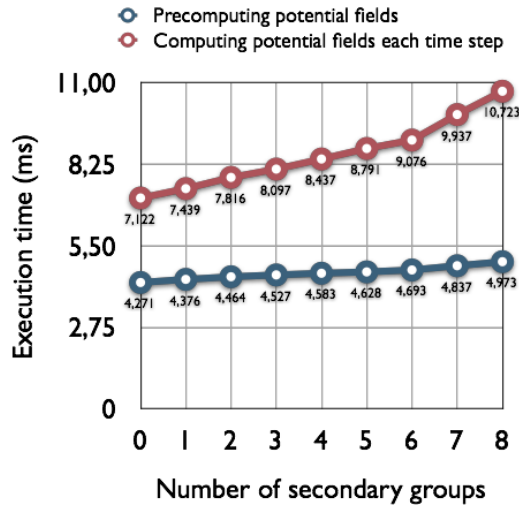


Fig. 4. Average execution time (ms) for simulations performed in the art gallery scenario with three hundred agents. The graph shows the performance gain when precomputing potential fields.

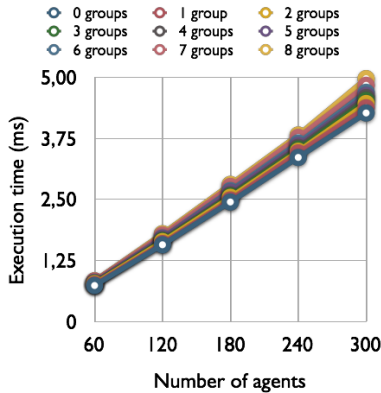


Fig. 5. Average execution time (ms) for simulations performed in the art gallery scenario. The graph shows that the addition of agents has a linear impact in the execution time of the model.

flow of agents into an environment. In that scenario, two secondary groups with 20 x 16 cells were used to organize the flow in a metro entrance with four passages. One of the secondary groups leads agents to the upper passages, whereas the other group leads agents to the lower passages. Two primary groups with 12 x 4 cells were used to cover the entire environment to keep the agents walking in the corridor. Up to three hundred agents interact with the scenario. Each simulation was performed for five minutes. Results can be seen in Figure 6.

- **Building evacuation.** This scenario was used to demonstrate the suitability of the model for simulating evacuation scenarios and for representing scenarios as set of connected groups. The scenario is composed of four rooms and two corridors (six primary groups (four with 10 x 9 cells, one with 11 x 3 cells and one with 3 x 7 cells)), from where up to nine hundred agents must evacuate. The performance of each simulation was measured until

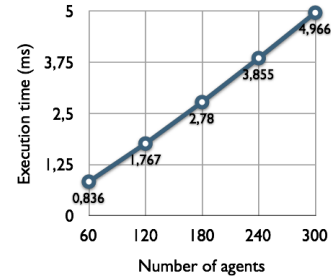


Fig. 6. Average execution time (ms) for simulations performed in metro entrance scenario.

all agents left the building. Results can be seen in Figure 7.

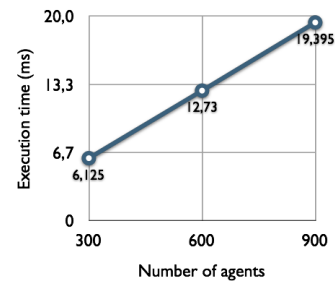


Fig. 7. Average execution time (ms) for simulations performed in the building evacuation scenario.

## B. Discussions

1) *Performance analysis:* The results obtained with the first scenario's simulations showed that increasing the number of secondary groups causes little impact on performance. Since the potential fields are precomputed, the increase in execution time observed by adding groups (Figure 3) is justified by the search for groups by agents. Also, it is perceivable that the higher the number of agents is, the higher the cost of searching for groups becomes. In Figure 4, we compare the performances (execution time) of simulations in which the potential fields were precomputed against those in which the potential fields were computed every time step. The results show a great performance gain, a reduction of more than 50% of the execution time in the simulation with eight groups and three hundred agents. In Figure 5, we can observe that execution time grows linearly with the addition of agents regardless of the number of groups. That linear growth was expected, because RVO has linear behavior.

The results obtained with the second and third scenarios also showed the expected linear behavior (figures 6 and 7). Comparing the three scenarios with 300 agents, we can observe that the first two have a similar performance with execution time of five milliseconds approximately, whereas the third scenario has an execution time of six milliseconds approximately. That difference is justified by the characteristics of an evacuation scenario, since in that kind of scenario the crowd is more

dense, i.e., agents have more interactions with one another and more interactions lead to more computational cost to treat the collisions among agents. The RVO algorithm has some adjustable parameters that allow relieving the cost of collision avoidance, such as, the number of neighbors to take into account and the maximal distance to them.

Finally, the results showed that the performance of the model is more affected by adding agents than by adding groups, and that computational cost grows linearly according to the number of agents. Thus, even with a low number of agents in the first two scenarios (because of their nature), it is easy to forecast their performance behavior with a higher number of agents.

2) *Behavior analysis:* The proposed model uses potential fields for global navigation and RVO for local collision avoidance. Thus, it is expected that the agents behave according to RVO when avoiding collisions, making turns and adapting their velocities. In all scenarios, it is possible to observe emergent behaviors and the variety of agents with different velocities interacting with the groups (see accompanying video). In our model, predictive behavior is no longer provided by potential fields since they are not density dependent nor dynamic (in Treuille’s model [3], for example, agents reacts to each other in advance because discomfort is set in front of them). Nevertheless, that kind of behavior can be achieved by RVO, only setting its parameters accordingly.

In the first scenario, the agents are always interacting with secondary groups, giving the crowd heterogeneous behaviors with no need to pre-configure each agent individually to interact with the environment (Figure 8).

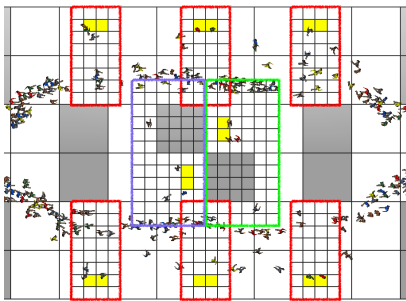


Fig. 8. Art gallery scenario with eight secondary groups (highlighted boundaries represent secondary groups, yellow cells represent goals and gray cells represent obstacles). Agents randomly enter in secondary groups. They interact with those groups stopping at goals to observe points of interest.

The second scenario shows that the secondary groups can successfully organize the flow of agents in a crowd (Figure 9), and the third scenario shows that the model also can be used for building evacuation simulations (Figure 10). In those last two scenarios, it is perceivable the formation of jams/bottlenecks and arching at the passages, what was expected, since RVO is able to reproduce those and other emergent behaviors.

In the first two scenarios, the agents change directions instantaneously when they start to suffer the action of another group. That behavior can be visually awkward depending on

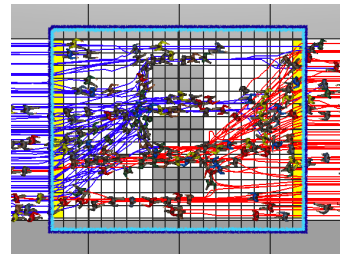


Fig. 9. Metro entrance scenario (highlighted boundaries represent secondary groups, yellow cells represent goals and gray cells represent obstacles). Two secondary groups were used to organize the flow of the agents at the passages.

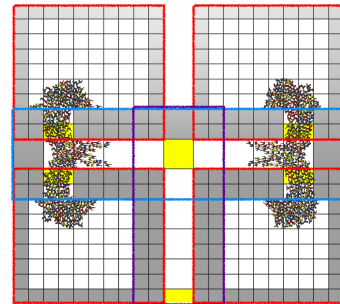


Fig. 10. Building evacuation scenario with six primary groups (highlighted boundaries represent groups, yellow cells represent goals and gray cells represent obstacles). Agents evacuating 4 rooms of a building, forming arches at exits.

the velocity of the agent and on the turn made by it, because inertia is not taken into account in the model.

## V. CONCLUSIONS

The model proposed in this work enables the introduction of many behaviors in crowd simulations. That is possible through the use of secondary groups in an approach based on potential fields. The use of potential fields with RVO is another contribution of this work. The potential fields are used to provide the agents with preferred velocities toward their goals, whereas RVO is used to treat collisions among the agents and to prevent them from penetrating into obstacles.

The tests have shown the suitability of the model for representing a variety of scenarios: scenarios with secondary goals like art galleries, museums, and malls; scenarios where there is a need for organizing the flow of the agents such as in narrow passages, and in building entrances and exits. Moreover, the model includes all the advantages of RVO, supporting dense crowds and presenting many emergent behaviors, and can also be used to represent evacuation scenarios.

In our work, potential fields are precomputed, thus, the use of many of those fields presents little impact on performance, since that impact is related to the search of groups during the simulation. Also, the execution time of a simulation grows linearly according to the number of agents.

Despite the useful characteristics of our model, some improvements can still be proposed, such as the use of a hierarchical structure to maintain the groups, and the introduction of a way to lead an agent to pass inside a secondary group

and interact with it. A spatial partitioning technique could be used to reduce the cost of searching groups. And, finally, it is necessary to investigate a way to avoid abrupt turns by the agents while changing groups.

## REFERENCES

- [1] C. Warren, "Global path planning using artificial potential fields," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, 1989, pp. 316–321 vol.1.
- [2] —, "Multiple robot path coordination using artificial potential fields," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, 1990, pp. 500–505 vol.1.
- [3] A. Treuille, S. Cooper, and Z. Popović, "Continuum crowds," in *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*. New York, NY, USA: ACM, 2006, pp. 1160–1168.
- [4] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *IEEE International Conference on Robotics and Automation*, May 2008, pp. 1928–1935.
- [5] D. Thalmann and S. R. Musse, in *Crowd Simulation*. Springer London, 2008.
- [6] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.
- [7] —, "Steering behaviors for autonomous characters," in *Game Developers Conference 1999*, 1999, pp. 763–782.
- [8] —, "Interaction with groups of autonomous characters," in *Game Developers Conference 2000*, 2000, pp. 449–460.
- [9] S. R. Musse and D. Thalmann, "Hierarchical model for real time simulation of virtual human crowds," *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, pp. 152–164, April 2001.
- [10] W. Shao and D. Terzopoulos, "Autonomous pedestrians," *Graphical Models*, vol. 69, no. 5-6, pp. 246–274, 2007.
- [11] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, 1995.
- [12] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, 2000.
- [13] H. Jiang, W. Xu, T. Mao, C. Li, S. Xia, and Z. Wang, "Continuum crowd simulation in complex environments," *Computers & Graphics*, vol. 34, no. 5, pp. 537–544, 2010.
- [14] S. Chenney, "Flow tiles," in *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ser. SCA '04. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2004, pp. 233–242.
- [15] X. Jin, J. Xu, C. C. L. Wang, S. Huang, and J. Zhang, "Interactive control of large-crowd navigation in virtual environments using vector fields," *Computer Graphics and Applications, IEEE*, vol. 28, no. 6, pp. 37–46, 2008.
- [16] M. Kapadia, S. Singh, W. Hewlett, and P. Faloutsos, "Egocentric affordance fields in pedestrian steering," in *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, ser. I3D '09. New York, NY, USA: ACM, 2009, pp. 215–223.
- [17] R. Silveira, F. Dapper, E. Prestes, and L. Nedel, "Natural steering behaviors for virtual pedestrians," *The Visual Computer*, vol. 26, no. 9, pp. 1183–1199, 2010. [Online]. Available: <http://dx.doi.org/10.1007/s00371-009-0399-0>
- [18] S. Patil, J. van den Berg, S. Curtis, M. C. Lin, and D. Manocha, "Directing crowd simulations using navigation fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 244–254, February 2011.
- [19] J. van den Berg, S. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Springer Berlin / Heidelberg, 2011, vol. 70, pp. 3–19.
- [20] E. Ju, M. G. Choi, M. Park, J. Lee, K. H. Lee, and S. Takahashi, "Morphable crowds," *ACM Trans. Graph.*, vol. 29, pp. 140:1–140:10, December 2010.
- [21] J. Ondřej, J. Pettré, A.-H. Olivier, and S. Donikian, "A synthetic-vision based steering approach for crowd simulation," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 123:1–123:9, Jul. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1778765.1778860>
- [22] A. de Lima Bicho, R. A. Rodrigues, S. R. Musse, C. R. Jung, M. Paravisi, and L. P. Magalhães, "Simulating crowds based on a space colonization algorithm," *Computers & Graphics*, vol. 36, no. 2, pp. 70–79, 2012, virtual Reality in Brazil 2011.
- [23] S. J. Guy, J. Chhugani, S. Curtis, P. Dubey, M. Lin, and D. Manocha, "Pedestrians: a least-effort approach to crowd simulation," in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '10. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2010, pp. 119–128.
- [24] S. J. Guy, S. Curtis, M. C. Lin, and D. Manocha, "Least-effort trajectories lead to emergent crowd behaviors," *Phys. Rev. E*, vol. 85, p. 016110, Jan 2012.
- [25] N. Pelechano, J. M. Allbeck, and N. I. Badler, "Controlling individual agents in high-density crowd simulation," in *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007, pp. 99–108.
- [26] B. Yersin, J. Maïm, F. Morini, and D. Thalmann, "Real-time crowd motion planning: Scalable avoidance and group behavior," *Visual Computer*, vol. 24, no. 10, pp. 859–870, 2008.
- [27] M. Xiong, M. Lees, W. Cai, S. Zhou, and M. Y. H. Low, "Hybrid modelling of crowd simulation," *Procedia Computer Science*, vol. 1, no. 1, pp. 57–65, 2010.
- [28] R. Narain, A. Golas, S. Curtis, and M. C. Lin, "Aggregate dynamics for dense crowd simulation," *ACM Transactions on Graphics*, vol. 28, pp. 122:1–122:8, 2009.
- [29] S. Singh, M. Kapadia, B. Hewlett, G. Reinman, and P. Faloutsos, "A modular framework for adaptive agent-based steering," in *Symposium on Interactive 3D Graphics and Games*, ser. I3D '11. New York, NY, USA: ACM, 2011, pp. 141–150.
- [30] R. L. Hughes, "The flow of human crowds," *Annual Review of Fluid Mechanics*, vol. 35, no. 1, pp. 169–182, 2003.
- [31] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.
- [32] I. Rudomín, F. Perez, and E. Millán, "Groups and crowds with behaviors specified in the environment," in *IVEVA*, ser. CEUR Workshop Proceedings, vol. 97. CEUR-WS.org, 2004.
- [33] H. Yeh, S. Curtis, S. Patil, J. van den Berg, D. Manocha, and M. Lin, "Composite agents," in *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2008, pp. 39–47.
- [34] M. Schuerman, S. Singh, M. Kapadia, and P. Faloutsos, "Situation agents: agent-based externalized steering logic," *Computer Animation and Virtual Worlds*, vol. 21, no. 3-4, pp. 267–276, 2010.
- [35] B. Ricks and P. Egbert, "More realistic, flexible, and expressive social crowds using transactional analysis," *The Visual Computer*, pp. 1–10, 2012.
- [36] M. Sung, M. Gleicher, and S. Chenney, "Scalable behaviors for crowd simulation," *Computer Graphics Forum*, vol. 23, no. 3, pp. 519–528, 2004.
- [37] M. Paravisi, A. Werhli, J. J. Junior, R. Rodrigues, C. R. Jung, and S. R. Musse, "Continuum crowds with local control," *Computer Graphics International*, pp. 108–115, 2008.
- [38] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Computer Graphics Forum*, vol. 26, no. 3, pp. 655–664, 2007. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2007.01089.x>
- [39] A. Lerner, E. Fitusi, Y. Chrysanthou, and D. Cohen-Or, "Fitting behaviors to pedestrian simulations," in *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '09. New York, NY, USA: ACM, 2009, pp. 199–208.
- [40] S. J. Guy, S. Kim, M. C. Lin, and D. Manocha, "Simulating heterogeneous crowd behaviors using personality trait theory," in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '11. New York, NY, USA: ACM, 2011, pp. 43–52.
- [41] OSG, "OpenSceneGraph," 2013, available at: <http://www.openscenegraph.org>.
- [42] P. Sunna, A. Backman, and D. Sjölie, "ReplicantBody," 2013, available at: <http://www.vrlab.umu.se/research/replicantbody>.