Group navigation state-of-the-art report

Clodéric Mars - Golaem[*]

Contents

History

21/6/2011	Initial version.
16/10/2011	This revision.

[*] cloderic.mars@gmail.com

License

1 Introduction

Autonomous human-like characters being able to navigate in a 3D environment finding their paths and avoiding collisions while exhibiting a convincing behavior is now fairly common. The popularity and the quality of recent work [25, 17, 15, 12] shows that the simulation of hundreds of navigating entities is now within the reach of almost everyone. But most of these work focus on simulating lonely entities taking their neighbors into account only to avoid collisions. Like real ones, our virtual humans should be able to walk down the road with their group of friends. Like real ones, our virtual soldiers must be able to march on enemy positions while staying in formation. And like real ones, our virtual tourists have to be able to enjoy their tour of the Mont Saint-Michel following their guide's umbrella.

Before we start designing and implementing group navigation behaviors in Golaem Path [4], let's first see what others have done. This document is a state-of-the-art study of existing work concerning group and formations navigation with a focus on algorithms and implementations. The first part defines the different categories of navigating groups we're interested in. The second part focuses on how the group members can stay grouped or in formation during navigation. The third and last part talks about the group's navigation from path planning to steering and collision avoidance.

2 Taxonomy

Aka "flocks, herds and school" and...

Before we talk velocity, steering behaviors and gradient descent, let's first present the kind of groups we're trying to make navigate, their characteristics and their constraints. In this part we'll divide navigating groups in three categories, flocks, formations and small groups resulting of social influences.

2.1 Flocks

A flock is primarily a group of bird traveling together but it can be applied to other animal species (cf. figure 1) as well as humans (e.g. a flock of school children is crossing the street to the swimming pool). Entities in a flock travel at roughly the same speed and form a cohesive group without strict arrangement.

In what must be the two most cited articles in the field [21, 22], Reynolds studied empirically how flocks members move relatively to each others. With simple behaviors he was able to recreate a flock of autonomous entities, we'll dig into more details in part 4.1.

Figure 1: Sheep flock (from `http://flic.kr/p/Yuy89`).

2.2 Formations

While flocks do not follow more rules than the cohesion of the group, formations are a kind of group arrangement where members need to enforce strict rules. Both in a combat or a parade (cf. figure 2), the spatial arrangement, i.e. the relative positions of members, is designed for a precise purpose, tactic or aesthetic; it is the first rule that needs to be followed. Secondly, in a combat context, formation gets much of its usefulness from overlapping fields of fire and sight, that's why the orientation is another rule to be followed [3]. The last rule is to assign entities having the right role to the right slot: archers at the back, footsoldiers facing the enemy. As navigation and military simulation are important for real time strategy games, interesting and working solutions has been developed early: Dave Pottinger, who worked on the Age of Empire series, presented his in a Gamasutra article [18].

2.3 Small social groups

Beyond amorphous flocks and rigid military formations, groups that are more common in our everyday life are small and their spatial configuration is the result of social factors and crowd density. Two recent survey focuses on those small social groups [16, 13], they lead to the same conclusions. The two studies were conducted from videos taken at public spaces (in France and Ireland).

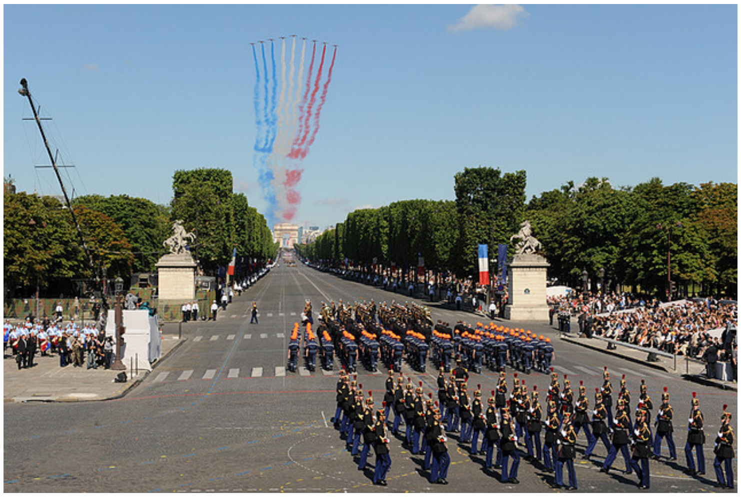


Figure 2: Foot soldiers in formations for the Bastille day military parade (from `http://flic.kr/p/57KrxH`).

Their observations show that: there are more groups than single pedestrians, groups of more than four are very rare and most of the groups are, indeed, pairs.

More interesting, it appears the formation adopted by the observed groups is influenced both by the lateral clearance to nearby obstacles and by the social interaction between members of the group (cf. figure 3). When motion is not constrained (i.e. when obstacles are far and the crowd density is low) a group tends to adopt an abreast formation that facilitates dialog between its members. When facing navigation constraints, to reduce its frontal width, the group compact the formation. And when the lateral space between each member become too thin, i.e. when members are shoulder-to-shoulder, the formation is staggered. The bending of the group is, most of the time, forward (V-like formation) to maintain good communication when a backward bending (inversed-V-like or wedge formation) would be more flexible moving against an opposite flow.

Finally, while groups tend to avoid collisions, with other pedestrians or with obstacles, as a whole they are able to split if needed merging back afterwards (cf. figure 4).

3 Stay Grouped!

How members enforce group constraints

In order to get a valid simulation of group navigation we first have to design the behavior of the group members. In this part we'll study two families of methods to obtain groups: using local rules or following a designed formation. Finally we'll focus on how a member's individual navigation behavior is affected when

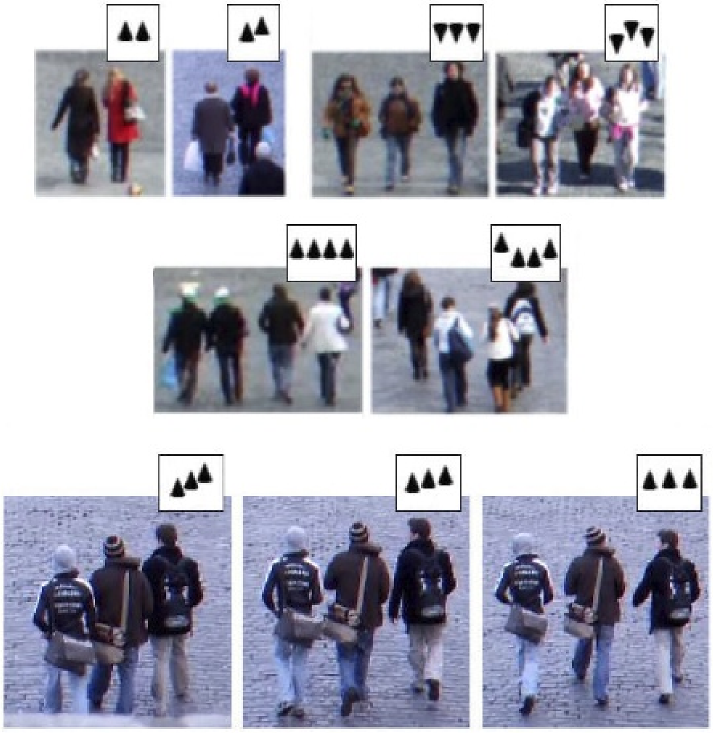


Figure 3: Examples of groups of two, three and four (from [16]).

part of a group.

3.1 Emergence

In most modern navigation engine, the simulated entities are autonomous, their behavior rely on their local "perception" to take action not on an external choreographer. With this approach in mind, several works relies on decentralized behavior to enforce group constraints.

3.1.1 Flocking

At the core of Reynolds' work [21, 22], three steering behaviors (cf. figure 5) allows entities to flock. For any given entity in the group, separation makes him move apart too close neighbors, alignment makes him go in the same direction as other members and cohesion makes him move towards the group's COM (center of mass). These simple behaviors allow the emergence of a flock. Others took inspiration from this work and adapt it to their architecture using voting steering behaviors [6] or relying on specific environment abstraction such as grids [10] or roadmaps [2, 7].

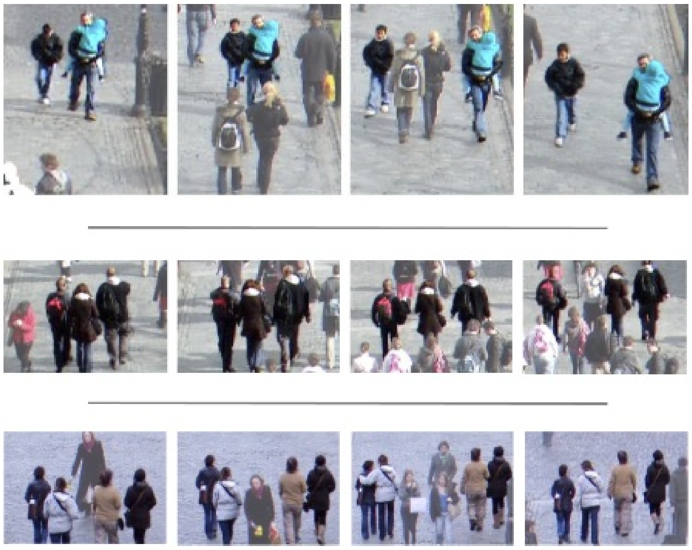


Figure 4: Examples of groups of two, three and four splitting to avoid collison (from [16]).

3.1.2 Social relations

In his recent paper, Moussaïd [13], extends the social force model [5] to obtain flocking as well as more structured small formation introducing a communication force. The gaze of group members is attracted by the center of interest of the group; here its center of mass (COM) is used. The communication force tries to limit the gaze deviation by decelerating the entity thus enabling the observed V-like formation (cf. 3.3).

In order to take into account social relations between members of the group, Qiu uses a member-to-member influence matrix [19]. This matrix is taken into account when computing the attraction between members of the group. This approach allows the definition of one or more attractive (or repulsive) members

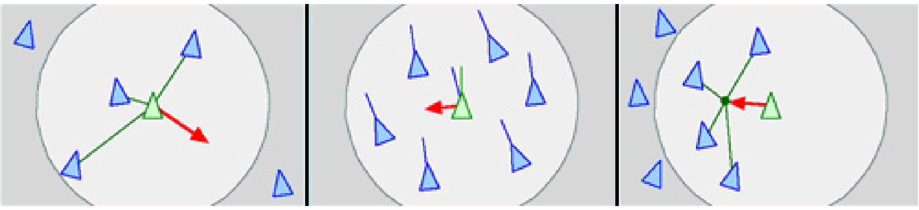


Figure 5: Flocking steering behaviors, from left to right: Separation, Alignment and Cohesion (from [22]).

of the group such as more talkative persons or tourist guides.

3.1.3 More rigid formations

Local rules can also exhibit a more strict formation. Taking inspiration from molecular crystals, Balch and Hybinette designed the attachment sites method [1]. Each entity, given its desired formation, computes several attachment sites on its neighbors (cf. figure 6) and steers to reach the nearest available. The resulting formation arrangement is a direct result of the attachment sites position and it can scale to any number of group members. But as the attachment rules are local, no control on the formation overall shape is possible.

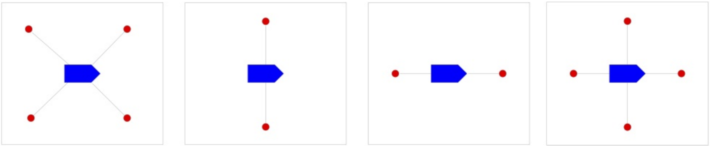


Figure 6: Blue entities and different red attachment sites (from [1]).

3.2 Choreography

While groups whose members are implementing local rules can exhibit convincing behavior, they can't take into account the group as a whole and thus are not fully controllable. If an exact group arrangement is needed, some of the behavior must be deported to an upper level of control [14]. In this part we'll study the three features needed to make a group stay in a given formation: the formation design, the slot assignment and the formation hold.

3.2.1 Design

A formation is basically a list of slots to which members of the group will be assigned. As we focus on pedestrian navigation, a slot is basically a 2D position relative to the group "center". In a military context two properties of the slot must be added, as we saw in section 3.2: the orientation and the role (i.e which kind of entity should be assigned to this slot).

Such a formation definition can be designed manually following military principles or artistic concerns [3]. Another approach has been taken by Karamouzas and Overmars [9], they extracted from the data collected by Moussaïd et al. [13] a set of typical formations for small social groups of 2 to 4 members. The figure 7 shows the three kinds of formations they get for groups of 3; similar formations are obtained for groups of 2 or 4.

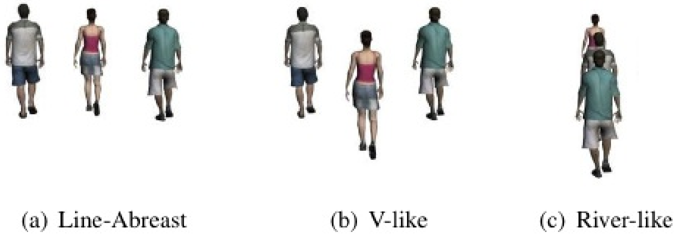


Figure 7: Three members social group formations (from [9]).

3.2.2 Assign

We now assume the formation design is chosen and defines as many slots as group members; if not, simple techniques can be used to select the used slots or create needed slots [24]. The next needed step before our entities can navigate as a group, is to assign each of them to a slot. This part might seem trivial but should be implemented right to avoid destroying the simulation credibility with traffic jams between members of the same groups.

Implementation-oriented papers [3, 11] describe different solutions and their quality. As everyone should expect, random or index-based (ith members with ith slot) assignment are most of the time really bad: the entities might have to cross each others and to circle around the group to get to their slot. To avoid this bad result, one solution could be to, sequentially assign to each entity, its closest free spot, this solution is easy to implement and might work but last entities might end up to far slots as the closest one are already taken. The best solutions would be to globally minimize the distance the entities are covering to get to their slots but its implementation would lead to an $O(n!)$ complexity as every permutation would have to be tested.

One solution to get a good result and keep a good complexity is a two steps process. First compute for each members of the group what is their cost to be assigned to which slot, allowing certain slot to be specialized for certain entity roles. Then, from the most expensive to assign to the cheapest assign entities to their preferred available slot. This solution might fail to get the optimal assignment but should have decent result while keeping a low complexity $O(n^2)$ [11].

Another solution works only when no specialized slots are defined. Given a formation design, first sort the slots spatially; for a horizontal line formation, the slot might be sorted from left to right along a horizontal vector. Then sort the group members in the same way; finally assign the ith entity to the ith slot [3].

3.2.3 Maintain

The formation being built, entities are able to reach their slots and to arrange into the designed formation. Let's see how they are able to maintain it while the group is moving.

In earlier work [18], a simplistic approach is taken: once part of a formation, entities are no longer responsible for their steering, their position is set at each time step according to the formation. This solution is fine if the group steering (cf. part 5) is robust enough to handle the desired level of collision avoidance. Using the latter solution, the members do not have an own steering behavior: it is difficult to get anything but a strictly followed formation. To allow formations of individuals, several works uses a more loose approach, members are given a local goal to reach in order to stay in formation and are in charge of fulfilling this goal. A simple approach is to provide each member with a simple "reach target" steering behavior where the target is its slot's future position [9, 23]. This approach works better if the slots are within navigable space as shown in Schuerman et al. videos. To avoid this problem, Silveira, Prestes and Nedel define a group potential map using slots as attractors and obstacles as repulsors thus providing to members a velocity that maintain the formation while keeping a smooth path when obstacle are present. It allows entity to break formation to pass through tight corridors and around small obstacles [24].

3.3 Group members steering

Whether the group is a flock or a formation, the individual steering behaviors, the collision avoidance in particular, of its members has to be considered. We've seen that in some case the individual steering behaviors are disabled [18]. In most work though, the group navigation is designed to work in conjunction with its members', the main problematic is to blend the individual behavior with the group orders.

In some architecture, each entity has several independent behaviors producing different steering orders that are blended together using weight and priorities. In this kind of approach the group orders are executed by another behavior and are part of the final blend [13, 20, 6].

In most modern architecture, a pipeline of behaviors produce a steering order each element taking the previous into account while enforcing new constraints or orders, the last one being collision avoidance [4, 12]. The group orders are implemented as a part of this pipeline, they are fed to the following elements among which the collision avoidance [9, 24].

Schuerman et al. state that while the same collision avoidance behavior can be used by the entities when they are part of a group, it must be adapted. As a matter of fact, collision avoidance algorithms such as RVO [25] try to enforce a safe distance to obstacles and other entities that might forbid tight formations. A "boldness" factor is introduced controlling how likely an entity is to yield to other entities during steering. When a strict formation is desired the "boldness" factor of its members is set high making them only try to avoid imminent collisions [23].

4 Who's in charge?

Group level decision making

In the previous part, we studied how group members are able to maintain cohesion or stay in formation. In this part, we'll study how the group, a whole, is able to make navigation decisions: path planning, path following, steering (including collision avoidance) and formation changes.

4.1 Anchor

While some behavior might be decentralized (c.f. 4.1), in order to leverage groups in a context where we need to make them go from A to B, top-down decision making is needed [14]. A group level process will be able to make the group move while each of its member follows. We call anchor a virtual object that group members use as a reference for their formation or flocking behaviors and to which they delegate some navigation processes. It aggregates various data relative to the group, such as its position, orientation and velocity and it's where the group level decision-making is done.

4.1.1 Leader

Trying to enforce a strict equivalence between simulated entities and actual characters, lots of works rely on a leader-followers approach. In such approach, one member of a group is the leader where the others are the followers. The leader takes responsibility for the whole group regarding the navigation; it becomes the anchor of the group [10, 19].

The implementation of such an approach using a navigation engine for independent entities is straightforward: the leader is similar to an independent entity; while the followers uses a subset of the default processes and maintain a reference to their leader.

But the leader can't reuse the exact same processes of an independent entity. Its navigation must take into account the bulk of the whole group as well as the different locomotion constraints of its followers. It is also better to differentiate the the leader's own attributes (position, orientation and velocity) from the anchor's [11]. Taking all these constraints into account makes the decision-making process of the leader very different from the other members.

4.1.2 Group entity

Noting that the leader-based approach as several flaws, a growing proportion of architectures chose to move the group anchor from the leader to a virtual group entity [23, 24, 9]. This virtual entity is similar to any other simulated entities but doesn't have visual or physic (i.e. regarding collision detection) representation. In such architecture, the group members are identical to one another. In Schuerman et al. [23] work, other entities detect groups entities during their collision avoidance process, trying to avoid collisions with groups as the whole. Entities are thus able to simulate the fact that pedestrians tend to avoid to pass through a group.

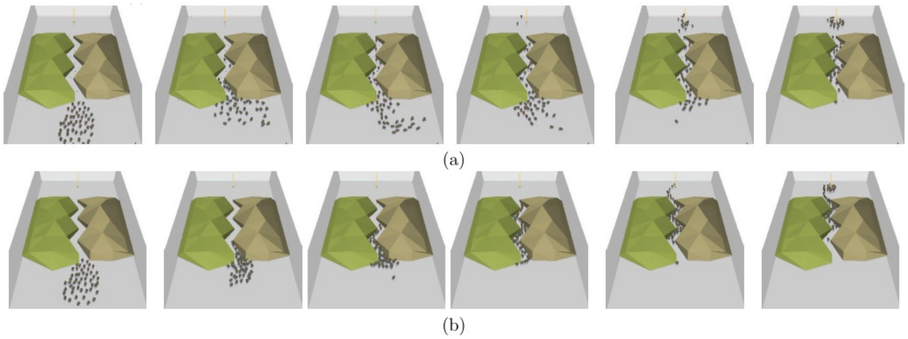
The group entity creates a one level deep hierarchy of entities. This approach can be taken a step further to create groups of groups and so on [23, 11] allowing a more structured crowd.

4.2 Path planning/following

One of the reasons for group navigation is to factorize a costly aspect of navigation: path planning. As a matter of fact the members of a group are expected to follow the same high-level path through the environment, a single query should be sufficient for the whole group.

The most important aspect of group level path planning is to choose how to take the bulk of the group into account. Contrary to a single entity where its bulk can't be changed and thus is a hard constraint, a group can reconfigure in order to pass through narrower corridors. The query has to be tuned in order to prefer paths on which the group, in its current spatial configuration, can navigate but be able to select narrow passages if necessary. This implies that the cost of falling back to a narrower spatial configuration can be compared to the cost of taking a longer path [7, 18, 2].

Once the path is computed, the path following process is able to provide local steering orders resulting in the entity following the path. Adapted to groups, this process makes the anchor follow the path. In some work [2, 18], the group level path following is also responsible for environment aware formation adaptation, allowing the formation to change when the clearance to obstacles changes. The figure 8 shows how a formation change allows a group to pass through a narrow passage more smoothly.


(a)

(b)

Figure 8: Passing through a narrow passage using flocking (a) or river-like formation (b) (from [2]).

4.3 Collision avoidance

Groups tend to stay coherent when navigating between obstacles and among other pedestrians, that's why several work features group level collision avoidance. Numbers of existing algorithm for entities can be applied directly or adapted for group level collision avoidance. As we noted for path planning and following, the main difference between groups and single entities is that their bulk is not a hard constraint. The spatial configuration of a group can be adapted to occupy less frontal space, less longitudinal space or both.

Schuerman et al. consider the bulk of the group as a disc allowing them to use RVO [23, 25]. The resulting collision avoidance is very conservative as the disc is, most of time greatly overestimating the real footprint of the group. Karamouzas and Overmars adapted their own collision avoidance algorithm, based on velocity space sampling and sweep collision test, to work on the oriented bounding box of the group [8, 9]. They further extend the algorithm by allowing formation adaptation. In practice, they generate samples based on velocity changes and formation changes interpolating the current formation with a library of valid formation (cf. figure 9). Each sample is weighted depending on its distance from the desired velocity and the desired formation and its time to collision. The number of candidate formations is limited to 15 as there's 5 possible formations and 3 interpolations computed per formation. These limits lower the number of considered samples, preserving the performances of the algorithm.

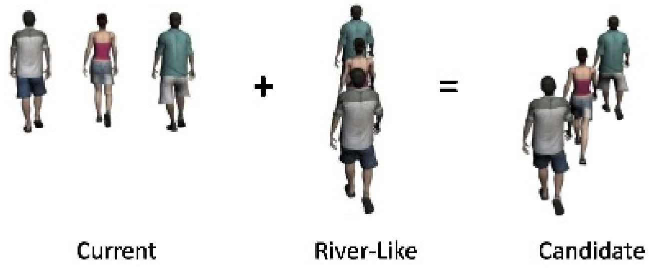


Figure 9: Linear interpolation between formations (from [9]).

Peters et al. rule based navigation method is able to handle both formation adaptation and group splitting which is not supported by the previously described methods [16]. Unfortunately the article doesn't provide many details on the algorithm.

Those group level navigation methods allows the group to take responsibility for a part of collision avoidance and more easily preserve the group cohesions. The members' own navigation behaviors are still necessary both to preserve individual behaviors and to avoid remaining collisions.

5 Conclusion

This review was limited to the navigation aspect of the group, it is supposed to answer to the question: *"How to make a group navigate in my simulation/game?"*. A group is, most of the time, the result of social interactions and relations between individuals that have other consequences beyond navigation strategy: body language, facial expressions, and speech... For a simulated group to be believable those other aspects must be addressed.

This state-of-the art study was instrumental in the design of Golaem Path upcoming group navigation features [4]. I hope others developers or even researchers in the field will find it useful. I'll try to update its content as other works come to my knowledge or are published.

References

[1] Tucker Balch and Maria Hybinette. Social potentials for scalable multirobot formations. In *IEEE International Conference on Robotics and Automation*, 2000.

[2] O Burchan Bayazit, Jyh-Ming Lien, and Nancy M Amato. Better group behaviors in complex environments using global roadmaps. In *8th International conference on Artificial Life*, 2003.

[3] Chad Dawson. Formations. In *AI Game Programming Wisdom*. 2002.

[4] Golaem. Golaem Path.

[5] Dirk Helbing, Illés Farkas, and Tamás Vicsek. Simulating dynamical features of escape panic. *Nature*, 2000.

[6] Terry R Hostetler and Joseph K Kearney. Strolling down the avenue with a few close friends. In *Third Eurographics Irish Workshop on Computer Graphics*, 2002.

[7] Arno Kamphuis and Mark H Overmars. Finding paths for coherent groups using clearance. In *2004 ACM SIGGRAPH/Eurographics symposium on Computer Animation*, 2004.

[8] Ioannis Karamouzas and Mark H Overmars. Simulating Human Collision Avoidance Using a Velocity-Based Approach. In *VRIPHYS 10: 7th Workshop on Virtual Reality Interactions and Physical Simulations*, 2010.

[9] Ioannis Karamouzas and Mark H Overmars. Simulating the local behaviour of small pedestrian groups. In *17th ACM Symposium on Virtual Reality Software and Technology*, 2010.

[10] Céline Loscos, David Marchal, and Alexandre Meyer. Intuitive Crowd Behaviour in Dense Urban Environments using Local Laws. *Proceedings of the Theory and Practice of Computer Graphics*, 2003.

[11] Ian Millington. Movement. In *Artificial intelligence for games*. 2006.

[12] Mikko Mononen. Navigation Loop. In *Paris Game/AI Conference 2010*, 2010.

[13] Mehdi Moussaïd, Niriaska Perozo, Simon Garnier, Dirk Helbing, and Guy Theraulaz. The Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics. *PLoS ONE*, 2010.

[14] Soraia Raupp Musse and Daniel Thalmann. Hierarchical Model for Real Time Simulation of Virtual Human Crowds. *Transactions on Visualization and Computer Graphics*, 2001.

[15] Jan Ondrej, Julien Pettré, Anne-Hélene Olivier, and Stéphane Donikian. A synthetic-vision based steering approach for crowd simulation. In *The 37th International Conference and Exhibition on Computer Graphics and Interactive Techniques*, 2010.

[16] Christopher Peters, Cathy Ennis, and Carol O'Sullivan. Modeling groups of plausible virtual pedestrians. *IEEE Computer Graphics and Applications*, 2009.

[17] Julien Pettré, Jan Ondrej, Anne-Hélene Olivier, Armel Crétual, and Stéphane Donikian. Experiment-based Modeling, Simulation and Validation of Interactions between Virtual Walkers. *Symposium on Computer animation*, 2009.

[18] Dave Pottinger. Implementing Coordinated Movement, 1999.

[19] Fasheng Qiu and Xiaolin Hu. Modeling Dynamic Groups for Agent-based Pedestrian Crowd Simulations. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2010.

[20] Fasheng Qiu, Xiaolin Hu, Xue Wang, and Saurav Karmakar. Modeling social group structures in pedestrian crowds. *Simulation Modelling Practice and Theory*, 2010.

[21] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *1987 International Conference and Exhibition on Computer Graphics and Interactive Techniques*, 1987.

[22] Craig W Reynolds. Steering behaviors for autonomous characters. *Game Developers Conference*, 1999.

[23] Matthew Schuerman, Shawn Singh, Mubbasir Kapadia, and Petros Faloutsos. Situation agents: agent-based externalized steering logic. In *International Conference on Computer Animation and Social Agents, 2010*, 2010.

[24] Renato Silveira, Edson Prestes, and Luciana P Nedel. Managing coherent groups. *Computer Animation And Virtual Worlds*, 2008.

[25] Jur van den Berg, Ming C Lin, and Dinesh Manocha. Reciprocal Velocity Obstacles for real-time multi-agent navigation. *International Conference on Robotics and Automation*, 2008.