Semi-Autonomous Navigation for Virtual Tactical Simulations in the Military Domain

Juliana R. Brondani[1], Luis A. L. Silva[2], Mateus B. Rutzig[2], Cesar T. Pozzer[2], Raul C. Nunes[2], João B. Martins[2] and Edison P. de Freitas[1]
[1]*Graduate Program in Electrical Engineering, Federal University of Rio Grande do Sul, Porto Alegre, RS, Brazil*
[2]*Graduate Program in Computer Science, Federal University of Santa Maria, Santa Maria, RS, Brazil*

Keywords: Semi-autonomous Navigation, Tactical Simulation.

Abstract: Integrated constructive and virtual simulations are becoming popular for tactical training in the military domain. An important aspect concerning the integration of these simulation models in the construction of virtual tactical simulations is the modelling and implementation of different kinds of semi-autonomous agents. A fundamental feature of these agents is the capability of intelligently and realistically modelling task-oriented navigation activities in large virtual terrain simulation environments, while following underlying military doctrine and tactics. This paper reviews important navigation issues that emerge in such simulation systems and prominent Artificial Intelligence (AI) techniques that have been explored to solve them. From this analysis, a hybrid, semi-autonomous navigation framework is proposed aiming to fulfil the needs of virtual tactical training simulations, more specifically, in the military domain. As implemented in a system for the virtual tactical simulation of artillery battery tasks, the framework shows how to overcome the challenges of implementing realistic global and local navigation behaviours for military units and, at the same time, it shows that the semi-autonomous behaviours implemented are of primary importance to allow interaction with users for learning purposes in the simulation exercises.

1 INTRODUCTION

Simulation systems are gaining more popularity for educational purposes, in which one of the most traditional and prominent scenarios is the case of military training (e.g. (Heinze et al., 2002); (Fletcher, 2009)), although there is an increasing interest in other domains, like industrial (e.g. (Murphy and Perera, 2002)) and medical (e.g. (McGaghie et al., 2010)) to name a few. These simulation systems present themselves as challenging environments for the investigation and proposition of solutions for semi-autonomous navigation problems as they allow agents to recreate different behaviours realistically.

In military simulations, Computer Generated Forces (CGFs) or Semi-Autonomous Forces (SAFs) (Tambe, Johnson et al., 1995) are developed to populate constructive, virtual and blended simulation systems (Stevens et al., 2015). It is expected that these forces would act as substitutes for key real-life entities, for which semi-autonomous navigation behaviours stand out when users need to be in the control of the simulated entities as part of the development of different military training goals. For this reason, the primary goal of these agents is to replicate relevant aspects of human behaviours realistically, while following military doctrine. When dealing with the implementation of semi-autonomous navigation behaviours, there are intelligent techniques proposed in the literature. However, most of these techniques only handle part of the relevant navigation issues in these systems, which indicates that these techniques seldom consider the implementation of complex, layered simulation scenarios. As discussed in this paper, there are dynamic navigation and collision detection/avoidance issues to be considered by a combination of global and local agent navigation behaviours. These behaviours should be consistent with real-life agent' actions. Moreover, for any military simulation, agents also compute navigation actions relying on data structures that represent relevant characteristics of large real-life terrain environments.

From this landscape, this paper discusses alternative navigation techniques ought to be combined in a Hybrid, Semi-Autonomous Navigation Framework to

Brondani, J., Silva, L., Rutzig, M., Pozzer, C., Nunes, R., Martins, J. and Freitas, E.
Semi-Autonomous Navigation for Virtual Tactical Simulations in the Military Domain.
DOI: 10.5220/0006863104430450
In *Proceedings of 8th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2018)*, pages 443-450
ISBN: 978-989-758-323-0

support the development of realistic simulations in virtual tactical simulation systems. As proposed here, these virtual tactical simulations provide an environment in which low- or mid-level decision makers can train their skills in how to better move and employ their units in a battle scenario to accomplish their missions. Filling the gap that exists in between constructive and virtual simulations, this type of virtual tactical simulation is grounded on a realistic scenario that presents all the necessary elements to exercise the tactical skills of these low- or mid-level commanders.

The paper is organized as follows: First, prominent techniques used to address the agents' navigation tasks are analyzed. Second, the virtual tactical simulation is discussed, highlighting its main characteristics and differences from the other types of simulation. Third, the Hybrid Semi-Autonomous Navigation Framework exploring the studied concept of virtual tactical simulation is presented and analyzed along with an application example. Then, discussions are presented highlighting relevant aspects raised during the development of the proposed framework, and finally, the concluding remarks are presented.

2 AGENT'S NAVIGATION BEHAVIORS IN SIMULATION SYSTEMS

Navigation issues have been handled by different techniques proposed in the literature (Botea et al., 2013) (Kapadia and Badler, 2013) (Algfoor et al., 2015). This section describes the most used ones in simulation systems for the military domain, allowing one to assess whether they cover the semi-autonomous navigation requirements of agents involved in virtual tactical simulation environments.

2.1 A* Algorithms

The A* algorithm (Nilsson, 1998) is an instance of a deliberative approach to the solution of navigation problems. Although the A* algorithm is widely explored in different fields, the computational cost of executing it may become prohibitive. That is because the computation cost of the A* sharply increases as the size of the virtual simulation environment and the number of agents increase. Some proposals to deal with known limitations of the A* algorithm are worth mentioning: i) The Local Repair A* (LRA*) describes a family of algorithms based on the recalculation of the remaining of an agent route when a collision with other simulation objects is imminent (Silver, 2005). ii) The Cooperative A* (CA*) searches for a path while considering the routes which are planned for other agents being executed in the simulation system (Silver, 2005). iii) The D* algorithm (Stentz, 1994) is capable of planning paths in unknown or partially known and changing environments as it is dynamic in the sense that the cost parameters used can change during the search for the solution. This technique also has known optimizations called D* Lite (Koenig and Likhachev, 2002) and Anytime D*(Likhachev et al., 2005). However, these proposals still have computational limitations when the virtual terrain size increases as this situation is often related to the increasing of the memory space requirements and the complexity of the simulation scenario. In these cases, hierarchical A* techniques are promising as they speed up the pathfinding process by reducing the complexity of the problem scenario as the pathfinding is broken down in a hierarchic structure (Cui and Shi, 2011). One of the first proposals of this hierarchical algorithm (Botea et al., 2004) proposes the HPA* (Hierarchical Path-Finding A*), a "hierarchical approach for reducing problem complexity in path-finding on grid-based maps." This technique proposes the creation of abstraction levels relying on clusters obtained from a regular grid terrain representation. The hierarchical approach mitigates the memory space problem and allows faster results to be computed. However, when dealing with a virtual terrain with large dimensions, as it is often the case of a simulation system for representing a real-life environment, it is not feasible to use a terrain representation, which is based on a regular grid. (Dooms, 2013) uses an A* adaptation to navigate upon a quadtree representation of the terrain, allowing rapid adaptation to a terrain representation which can be dynamic as this solution only requires a limited number of nodes to be re-evaluated. This algorithm still presents some drawbacks like the absence of parallelism during the search and the use of a regular grid in the representation of the space that is inside of each quadtree node.

2.2 Potential Fields

Potential fields and influence maps are strategies commonly used to treat dynamic obstacles (Silveira et al., 2010) (Hagelbäck, 2012). They are concepts originated in the robotics fields, first introduced for the treatment of real-time obstacle avoidance, which is a central issue in the development of navigation algorithms for mobile robots (Khatib, 1986). The main idea of the potential fields approach is to place attracting or repelling charges at points of interest in a simulation map. The overall idea is that the agent can calculate the resulting force according to the fields that are available in the positions around the agent, and then navigate to the most attracting position in the near surroundings. The problem of this reactive behaviour is that the navigation algorithm may get stuck at local optima where the highest potential position is the current position of the agent, but this highest potential position is not the destination position. In (Hagelbäck, 2012), this problem was handled by assigning small repelling fields to the last agent positions, like a pheromone trail used by ants. This approach was successfully used in the Open Real Time Strategy (ORTS) system (Hagelbäck and Johansson, 2008) as the maps representations used present large open areas. As reported in (Hagelbäck, 2012), in which maps from the popular RTS game StarCraft were considered, which are complex and have many choke points, the solution proposed did not work well due to a large number of local maxima. To overcome these problems, a potential field technique combined to a traditional A* was used, resulting in a hybrid approach. Although this global-local approach generates a more realistic behaviour for each agent, it incurs in a high computation cost because the agent keeps analyzing its surroundings and calculating the potential of each cell in its local map representation.

2.3 Steering Behaviours

(Reynolds, 1999) defined steering behaviour for autonomous agents as the ability to navigate around their world in a life-like and improvisational manner. Different steering behaviours like seek, flee and arrive for a simple vehicle model are proposed. Then these basic behaviours can be combined as to generate ones that are more complex. The steering behaviours are described in terms of the geometric calculation of a vector representing the desired steering force (Reynolds, 1999). Due to the representation simplicity of this approach in the solution of navigation problems, it is a popular framework in the computer game scenario. Despite this fact, problems like a trajectory with oscillations or a resultant vector being zero can appear when the steering approach is used. (Frey, 2015) describes attempts to overcome these steering difficulties with the introduction of weighting, prioritization and awareness concepts. However, these attempts might not solve the cited problems or, if they do, they increase the complexity of the system. For this reason, a technique called Context Steering is proposed. The basic idea is the use of context maps, where these maps describe interest and danger vector-based points around an agent. Then, map information is combined to elect which vector describes the better final decision according to the current state of the system. This resulting approach maintains the benefits of the steering behaviour since it creates a more “intelligent” behaviour at a local level for each agent.

2.4 Velocity Obstacles

Velocity Obstacle (VO) is most commonly used in the solution of reactive or local navigation problems. This technique emerged in the field of robotics (Fiorini and Shiller, 1998). VO defines a set of robot velocities (in geometric representation) that would result in a collision between the robot and a moving obstacle. However, the VO approach presents some drawbacks. One of them is the high computation cost to keep updating the velocities of agents during the execution of the system. Another difficulty is that agents do not consider that other agents also have decision-making abilities. In (Van den Berg et al., 2008), this issue is addressed with the development of the Reciprocal Velocity Obstacle (RVO) approach. RVO assumes that other agents in the simulations are also capable of similar collision-avoidance reasoning while guaranteeing oscillation-free motions. However, RVO ended up creating another issue called “reciprocal dance” (Feurtey, 2000). Reciprocal dance occurs when two agents cannot reach an agreement on which side to navigate around each other. To overcome this problem, a Hybrid Reciprocal Velocity Obstacle (HRVO) (Van Den Berg et al., 2011) approach was proposed. HRVO eliminates almost all oscillations in practice, but it does not guarantee smoothness in the movement. Therefore, trying to address all the problems cited before, the Optimal Reciprocal Collision Avoidance (ORCA) (Snape et al., 2012) can theoretically guarantee no oscillations and smooth movement.

3 VIRTUAL TACTICAL SIMULATIONS: FILLING THE GAP BETWEEN CONSTRUCTIVE AND VIRTUAL SIMULATION SYSTEMS

There is a live discussion about military simulation systems and their applicability. A consensus that emerges among the practitioners in this area is that there are clear benefits of using live, virtual and constructive simulations for military training. Such benefits are even augmented when these three conceptual types of simulation are combined in joint LVC simulations (Hodson and Hill, 2014). Despite these conceptual classifications, there are intermediary zones between constructive and virtual, and between virtual and live simulations (Meyer et al., 2014). The use of virtual simulations usually targets the training of technical and operational skills regarding a given military equipment. Thus, this specific application of virtual simulations can be characterized as virtual technical simulations. The second application of virtual simulations, which is more linked to constructive ones, does not target the training of how to handle a piece of equipment, but it is concerned with tactical aspects. This usage contrasts to constructive simulations, which are concerned with higher-level (strategic) aspects, completely abstracting finer details of a specific terrain, for instance. Thus, it is possible to define this second kind of virtual simulations as virtual tactical simulations.

Considering its proximity to the constructive simulation, the best way to build-up this virtual tactical simulation concept is to present examples of situations in which this type of simulation is used. In our project, the employment of an artillery battery in a military operation can be taken as a motivational example. In this simulation application, the role of a constructive simulation is to train high-ranked commanders in the situations in which it is worth using the battery. In this type of simulation, higher-ranked officers (e.g., generals and their high-level staff) select priority areas and targets in which the battery will act, for instance. In this constructive simulation, however, the engagement of desired targets occurs according to given rules, and this is enough to provide automatically generated results to be analyzed by these high-ranked commanders. Everything in between from the moment of the decision-making to the return of the simulation results is entirely abstracted, i.e., without any interference of the intended users. Analyzing what happens between these above described moments, several tactical decisions about how to better employ the battery are taken in a real-life military operation. From the moment in which commanders of the battery receive the command to engage a given set of targets, low- or mid-level military personnel also need to tactically analyze the terrain, to select where the most suitable routes are as to safely navigate with vehicles that compose the battery, among other doctrine-based tasks. It is relevant to observe that a virtual technical simulation system is not able to train someone with the necessary skills to take these tactical decisions. However, where will the commander of the battery train his/her tactical skills to best select and execute a set of tactical actions as mentioned above? It is particularly for this type of real-life training situation that the virtual simulation is tailored for.

In virtual tactical simulations, there are different degrees of user interaction, which go from very high-level orders (like in the constructive ones) to more detailed ones (which are closer to technical-virtual or life simulations). Depending on the situation being simulated, these levels of interaction, which need to be reflected in the levels of autonomy of the simulation algorithms being used, may also be used alternately. In this context, an important issue for virtual tactical simulation systems in the field of military training is that they are used for educational purposes. In these systems, therefore, simulated agents have the capacity of receiving both computational and human inputs, as well as the capacity of reasoning about the best outcome of these actions according to their current situation.

4 A CASE FOR A HYBRID SEMI-AUTONOMOUS NAVIGATION FRAMEWORK FOR VIRTUAL TACTICAL SIMULATIONS

Navigation algorithms described in the literature are mostly focused on solving specific navigation problems. It means that they are rarely combined to handle realistic simulation scenarios where the navigation issues change dynamically. To address this drawback, a Hybrid, Semi-Autonomous Navigation Framework is proposed aiming to fulfil the needs of virtual tactical training military simulations. The overall idea of the proposed hybrid navigation framework is to combine global and local navigation approaches to tackle static and dynamic simulation issues over open fields and roads of a large virtual terrain considering the doctrine-oriented navigation demands of an artillery battery in a tactical battle scenario.

The global navigation approach works over a symbolic representation of the virtual terrain environment to search longer paths where the algorithms used have to consider simulation performance constraints. To couple with that, the proposed framework employs a multi-resolution quadtree data structure that symbolically represents static obstacles found in the virtual terrain scenario (Figure 1 (A)). In many senses, such multi-resolution quadtree supports the use of a hierarchical A* navigation algorithm. Thus it is possible to find paths in large virtual terrains with lower overhead than traditional A* algorithm.

In Figure 1 (A), a real large terrain (50km²) is divided into a quadtree containing thirteen representation levels. The proposed quadtree coupled to the hierarchical A* automatically adjusts the depth of the structure considering non-functional requirements of the simulation system, such as available computation capability and memory footprint. For instance, if there are enough memory and execution time for a finer search of an agent route in the virtual terrain environment, the algorithm considers finer terrain resolutions to allow the construction of a more accurate navigation result. Based on the division shown in Figure 1 (A), the representation structure can reach a resolution around 2 meters in its finest level, allowing the representation of relevant obstacles such as rivers, mountains and trees, as these obstacles are identified in real-life terrain regions of military training (i.e., using the different map representation of a military training field). As far as static navigation obstacles are concerned, the multi-resolution terrain representations allow the algorithm to quickly detect a path from one point to another in the virtual terrain if such path exists. When constructing such quadtree structure upon the terrain, the navigation algorithm easily identifies all the navigation-prone neighbouring nodes of the one that is being analyzed (Figure 1 (B)). As a result, a path is returned (Figure 1 (C) as required to simulate a tactical action involving a path-following movement of an agent in a simulation exercise – such kind of path is represented by the line having circle markers in this Figure 1).

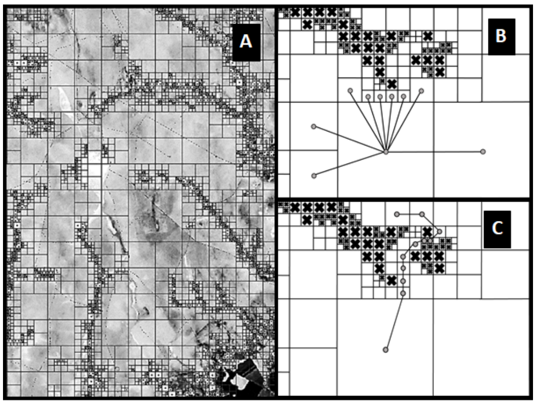

Figure 1: (A) Example of the terrain represented in a quadtree structure. (B) Connections between the nodes of the quadtree. (C) Path defined by the A* upon the quadtree representation.

Overtaking dynamic obstacles while the simulations are running is not a straightforward task. Constantly recalculation of paths for every agent may be prohibitive during real-time simulation executions even when an optimized virtual structure is available, such as a quadtree, to also represent the whereabouts of dynamic obstacles. For this reason, a local navigation strategy needs to be used in combination with a global pathfinding algorithm. Unlike the global navigation, this local navigation perspective considers that agents do not have a global view of the simulation environments since they just maintain a continuous and reactive relationship with their local surroundings. Importantly, such local strategy is responsible for the detection of dynamic obstacles and other dangers that an agent may face during the virtual tactical simulation exercises. To do so, the proposed framework explores steering behaviour techniques that are based on mathematical representations of the forces in the modelling and implementation of local navigation behaviours. Among other reasons, these algorithms allow a quick calculation of the resulting steering force, which is then used in the agents' local navigation actions.

As previously described, besides solving the global and local navigation, the framework needs to consider the military tactical doctrine to perform these agents' movements, as this requirement is fundamental to implement intelligent navigation algorithms in virtual tactical simulation settings realistically. During the movement of a battery, although agents adapt according to the current battle and terrain situations, they are implemented as to not deviate too much from a formation-kind of convoy organization detailed by the doctrine (e.g., to maintain a column formation while moving). Different from many computer game implementations, the proposed navigation framework considers that the navigation guidance coming from the military doctrine has a higher priority than the global and local navigation algorithms being executed.

5 THE SOLUTION OF A PRACTICAL SIMULATION SITUATION FOR GLOBAL AND LOCAL SEMI-AUTONOMOUS NAVIGATION

A practical simulation situation of a concrete military exercise involving an artillery battery is illustrated in Figure 2. This situation is representative of others that occur when virtual tactical simulations are developed in the implemented simulation system. As described here, this situation illustrates the use of the global and local semi-autonomous navigation algorithms implemented in the framework. In doing so, this simulation case involves a convoy with five vehicles parked at point A as shown in Figure 2. According to the tactical goals of an artillery battery mission, the convoy needs:

- to move from A to B;
- to deploy its artillery units in tactical positions in B according to the military doctrine.

For the first goal, the hierarchical A* algorithm defines, over the multi-resolution structure of the virtual terrain representation, the best path from A to B (represented by dotted line in Figure 2). The best path is not only the shortest path between these points, but also the path that can consider key terrain characteristics and the other forces (i.e. other agents) being simulated in the current battle situation. It is important to notice that the defined path which is generated by the pathfinding algorithm considers movement over different types of scenarios: roads and open fields. When there are roads in the area (from A to C), the pathfinding algorithm computes the route over an undirected graph that contains information about the roads and their interconnections. However, when an open field area is considered in the movement (from C to B), the pathfinding algorithm computes the route according to the obstacle information present in the nodes of the aforementioned quadtree structure. In the end, this quadtree structure represents the information that is observed by the agents allowing them to move over the virtual terrain representation.

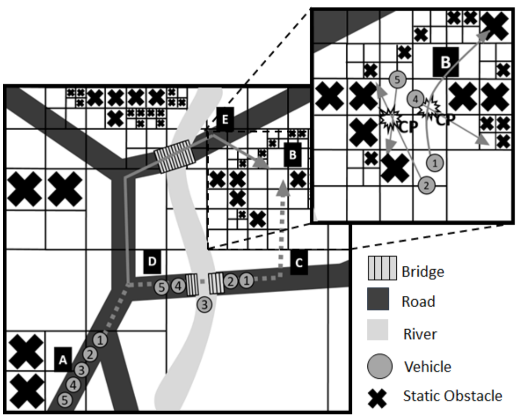

Figure 2: Example of a convoy execution a navigation task.

During the navigation of the convoy towards B, the vehicles may face situations that were not expected when the A* algorithm was executed (i.e., when the path was calculated). During the execution of the simulation exercise, as illustrated in Figure 2, the first two vehicles of the convoy crossed the bridge that connects A to B (passing by C). However, the third vehicle broke the bridge and fell into the river as a result of such a non-deterministic simulation situation. When this happens in this simulation system, the simulation stops since the route from A to B as planned by the global navigation algorithm no longer exists. As defined in this virtual tactical simulation system, this is a relevant military problem which was selected to be simulated due to its educational value to the users, among other such problems treated by the system. In summary, such kinds of problems offer relevant opportunities for simulation-based training. Different from fully-autonomous simulation scenarios, which are more common in constructive simulation systems, the users here are invited to decide which course of action they need to take according to their knowledge and experience with the underlying military doctrine and their mission goals, under these new conditions. One course of action is to allow the vehicles which already crossed the bridge (vehicles 1 and 2) to continue over the original route to reach the destination B. When this happens, the hierarchical A* algorithm is called again as to compute a new route from D to B for the remaining vehicles (vehicles 4 and 5). This way, a new route from the D to B (represented by the solid line in Figure 2) is created to allow the movement of vehicles 4 and 5. It is important to notice the split of the original convoy and the need for the user interaction to decide what the simulation system should do next, which reveals that a completely autonomous solution is not suitable in this kind of simulated situation. As this fragment of simulation exercise shows, there is a need for a certain degree of agent semi-autonomy, which is provided by the proposed framework that was implemented in the simulator. Similar to this one, other situations also require the intervention of users as the tactical training goals of the simulation system requires. So, this simulation training scenario requires the implementation of other degrees of semi-autonomy in the global and local agent navigation behaviours as this is likely to be the case of other simulation systems similar to this one.

For the second goal, the convoy is deployed in a selected region in the virtual terrain scenario according to the rules of the tactical military doctrine. In the example illustrated in Figure 2 (zoomed portion on the top-left), when vehicles approach their tactical positions, many conflict points (CP) are found, as it is expected to be the case of other multi-agent simulation systems. It means that the A* algorithm was used to define paths for different agents, where these paths produced overlapping routes, which may cause collisions among the agents while the simulations are running. In these dynamic simulation situations, the local navigation algorithm takes over as implemented in the simulator, applying steering behaviour forces to avoid collision between dynamic obstacles. In the implementation of social agent rules to solve such conflict situations, such as the rule: stop the movement and give preference to, for instance, the switch between the global and local navigation does not need to have user interaction, thus emphasizing the importance of combining both navigation types in a hybrid solution. Even when these local algorithms are in the control of the agents' actions, it is relevant to notice that the implemented algorithms also need to consider that the guidelines of a military doctrine have to be followed; otherwise, the local movement actions will look like randomly implemented, which is something that provides a poor realism for the simulations. While in many kinds of computer games the realism may not be an issue while dealing with local movement actions, this is a relevant aspect implemented in our simulation system. As described here, parameters in the semi-autonomous local algorithms (e.g., velocity, direction, etc.) are also explored by users under training to allow them to guide these local actions of movement as to fulfil defined training goals of the tactical simulation exercises.

6 DISCUSSION

In the context of military tactical training, it happens to be impractical to recreate complete scenarios using real people and equipment due to the cost, amount of required resources and even the danger in the handling of military equipment without making sure that users achieved a certain level of training maturity. In this setting, constructive and virtual simulations have been used for personnel training by familiarizing the trainees with tools, vehicles, equipment, military doctrines and routines employed in real life. It also means that there has been an increase of interest in the combination of different types of simulations leading to the term "blended training" (Stevens et al., 2015). Examples in the military simulation field that use blended architectures integrating a constructive system with human-in-the-loop are the Royal Australian Force (RAAF) simulation system (Heinze et al., 2002) and the semi-automated forces (SAFORS) in the large-scale SIMNET environment (Tambe et al., 1995). Different from these approaches, the proposal presented in this paper is a hybrid semi-autonomous navigation framework that allows a customized control of the tasks and agents being simulated. In addition, in virtual tactical simulations, the use of semi-autonomous algorithms become relevant as they combine a constructive navigation behavior with the possibility of human interference, which can provide different benefits for trainees involved in the simulation exercises.

In the field of simulation, a hybrid solution is also proposed in (Sahli and Moulin, 2005), describing how to support wildland fire suppression actions in a virtual environment scenario. As described in their work, the single-use deliberative techniques (like A*) is not sufficient to solve real problems, as their application problem can present too many constraints. These reasons also sustain the proposal of hybrid navigation algorithms to navigate in simulation environments. The use of multiple navigation techniques allows agents to plan a safe and fast route using a global planner. It also allows such agents to deal with dynamic obstacles, other agents, and unforeseen situations by using local navigation techniques.

7 FINAL REMARKS

The modeling and implementation of realistic navigation behaviors is a fundamental feature for intelligent agents in military simulation systems. This paper revisits the most prominent solutions in this area, highlighting key aspects of algorithms used in the construction of alternative navigation strategies. While reviewing the existing approaches, the paper proposes a framework that combines navigation solutions to address the realistic modeling of task-oriented military navigation needs. This framework addresses local and global navigation issues, the handling and avoidance of static and dynamic obstacles, the existence of multiple military agents along with their navigation needs, besides the possibility of semi-autonomy and user interaction in a simulation scenario. Moreover, the paper also proposes the description of large realistically terrains to allow the planning algorithms to maintain a global vision of the scenario allowing for better results in the pathfinding.

ACKNOWLEDGMENTS

We thank the Brazilian Army for the financial support through the SIS-ASTROS Project (813782/2014), developed in the context of the PEE ASTROS 2020.

REFERENCES

Algfoor, Z. A., M. S. Sunar and H. Kolivand, 2015. *A comprehensive study on pathfinding techniques for robotics and video games*. International Journal of Computer Games Technology 2015: 7.

Botea, A., B. Bouzy, M. Buro, C. Bauckhage and D. Nau, 2013. *Pathfinding in games*. Dagstuhl Follow-Ups, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Botea, A., M. Müller and J. Schaeffer, 2004. *Near optimal hierarchical path-finding*. Journal of game development 1(1): 7-28.

Cui, X. and H. Shi, 2011. *A*-based pathfinding in modern computer games*. International Journal of Computer Science and Network Security 11(1): 125-130.

Dooms, A., 2013. *Parallel multi-agent path planning in dynamic*.

Feurtey, F., 2000. *Simulating the collision avoidance behavior of pedestrians*. Master's Thesis.

Fiorini, P. and Z. Shiller, 1998. *Motion planning in dynamic environments using velocity obstacles*. The International Journal of Robotics Research 17(7): 760-772.

Fletcher, J., 2009. *Education and training technology in the military*. Science 323(5910): 72-75.

Frey, A., 2015. Context Steering: Behavior-Driven Steering at the Macro Scale. *Game AI Pro 2: Collected Wisdom of Game AI Professionals*. S. Rabin. Natick, MA, USA, A. K. Peters Ltd.

Hagelbäck, J., 2012. *Potential-field based navigation in starcraft*. Computational Intelligence and Games (CIG), 2012 IEEE Conference on, IEEE.

Hagelbäck, J., 2016. *Hybrid pathfinding in StarCraft*. IEEE Transactions on Computational Intelligence and AI in games 8(4): 319-324.

Hagelbäck, J. and S. J. Johansson, 2008. *The rise of potential fields in real time strategy bots*. Fourth Artificial Intelligence and Interactive Digital Entertainment Conference, Stanford University.

Heinze, C., S. Goss, T. Josefsson, K. Bennett, S. Waugh, I. Lloyd, G. Murray and J. Oldfield, 2002. *Interchanging agents and humans in military simulation*. AI Magazine 23(2): 37.

Hodson, D. D. and R. R. Hill, 2014. *The art and science of live, virtual, and constructive simulation for test and analysis*. The Journal of Defense Modeling and Simulation 11(2): 77-89.

Johnson, C. and A. J. Gonzalez, 2008. *Automated after action review: State-of-the-art review and trends*. The Journal of Defense Modeling and Simulation 5(2): 108-121.

Kapadia, M. and N. I. Badler, 2013. *Navigation and steering for autonomous virtual humans*. Wiley Interdisciplinary Reviews: Cognitive Science 4(3): 263-272.

Khatib, O., 1986. *Real-time obstacle avoidance for manipulators and mobile robots*. The international journal of robotics research 5(1): 90-98.

Koenig, S. and M. Likhachev, 2002. *D* Lite*. AAAI/IAAI 15.

Likhachev, M., D. I. Ferguson, G. J. Gordon, A. Stentz and S. Thrun, 2005. *Anytime Dynamic A*: An Anytime, Replanning Algorithm*. ICAPS.

McGaghie, W. C., S. B. Issenberg, E. R. Petrusa and R. J. Scalese, 2010. *A critical review of simulation□based medical education research: 2003–2009*. Medical education 44(1): 50-63.

Meyer, R., T. Andre, K. Conning Chik and G. Liming, 2014. *Performance Evidence Management in Live, Virtual, and Constructive Training*. Journal of Applied Learning Technology 4(4).

Murphy, S. P. and T. Perera, 2002. *Successes and failures in UK/US development of simulation*. Simulation Practice and Theory 9(6): 333-348.

Nilsson, N. J., 1998. *Artificial intelligence: a new synthesis*. Elsevier.

Reynolds, C. W., 1999. *Steering behaviors for autonomous characters*. Game developers conference.

Sahli, N. and B. Moulin, 2005. *Real-world pathfinding using agent-based simulation*. Agent-Directed Simulation Symposium, part of the SpringSim.

Silveira, R., L. Fischer, J. A. S. Ferreira, E. Prestes and L. Nedel, 2010. *Path-planning for RTS games based on potential fields*. International Conference on Motion in Games, Springer.

Silver, D., 2005. *Cooperative Pathfinding*. AIIDE 1: 117-122.

Snape, J., S. J. Guy, D. Vembar, A. Lake, M. C. Lin and D. Manocha, 2012. *Reciprocal collision avoidance and navigation for video games*. Game Developers Conf., San Francisco.

Stentz, A., 1994. *Optimal and efficient path planning for partially-known environments*. Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on, IEEE.

Stevens, J., P. Kincaid and R. Sottilare, 2015. *Visual modality research in virtual and mixed reality simulation*. The Journal of Defense Modeling and Simulation 12(4): 519-537.

Tambe, M., W. L. Johnson, R. M. Jones, F. Koss, J. E. Laird, P. S. Rosenbloom and K. Schwamb, 1995. *Intelligent agents for interactive simulation environments*. AI magazine 16(1): 15.

Van Den Berg, J., S. Guy, M. Lin and D. Manocha, 2011. *Reciprocal n-body collision avoidance*. Robotics research: 3-19.

Van den Berg, J., M. Lin and D. Manocha, 2008. *Reciprocal velocity obstacles for real-time multi-agent navigation*. Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, IEEE.