Hierarchical Terrain Representation and Flood Fill-based Computation of Large-Scale Terrain Changes for Agent-based Simulations

Luis A. L. Silva[1,2][a], Evaristo J. Nascimento[1,2][b], Eliakim Zacarias[2][c], Raul C. Nunes[1,2][d] and Edison P. Freitas[2,3][e]

[1]*Graduate Program in Computer Science, Federal University of Santa Maria, Av. Roraima 1000, Santa Maria - RS, Brazil*
[2]*SIS-ASTROS GMF Project, Federal University of Santa Maria, Av. Roraima 1000, Santa Maria - RS, Brazil*
[3]*Informatics Institute, Federal University of Rio Grande do Sul, Av. Bento Gonçalves 9500, Porto Alegre - RS, Brazil*

Keywords: Dynamic Terrain, Hierarchical Terrain Representation, Flood Fill Computation, Agent-based Simulation.

Abstract: Modern virtual training benefits from the recent advances in Agent-Based Modelling and Simulation (ABMS), making it possible to use real-world dynamic terrain scenarios that enhance the users learning from agent-based simulations. An important issue for distributed ABMS systems is the possibility of using terrain services that promptly compute large-scale terrain map changes as a result of natural phenomena such as river floods and wildfires. Performing the alterations in terrain maps is challenging since they depend on the combination of terrain features and terrain sizes. To address this problem, this work proposes the use flood fill-based techniques along with the hierarchical QuadTree approach for the terrain representation. We show that these techniques are essential to promptly compute the effects of the changes on a large number of nodes of the hierarchical map representation that captures the terrain features in different levels of detail. Also, a way to store and recover the QuadTree nodes in/from a dictionary-based memory is proposed, improving the nodes' refinement and restoration process when the terrain changes are required on the simulations. Experiments with the proposed techniques show encouraging results, with reduced computing times considering terrains with different characteristics and numbers of alterations.

1 INTRODUCTION

Agent-Based Modelling and Simulation (ABMS) (Macal 2016) is a valuable paradigm for virtual training and instruction in different application domains. One of the main ABMS challenges is the development of realistic simulations with non-static, large-scale, and real-world terrain environments. Relevant instances of the need for such virtual scenarios involve the simulation of flooding effects in large urban areas (Liang, Du et al. 2008) and the simulation of wildfires (Rui, Hui et al. 2018). Motivated by these complex phenomena, this work investigates the problem of implementing optimized techniques to compute the impact of these events on large areas of the hierarchical topology of the represented terrain maps. The possibility of modelling large-scale terrain map representations that can be extensively modified due to the effects of these phenomena is a key component for the simulation of multi-agent global and local realistic agent behaviours. With the terrain structure and a capability of modifying it, the agents have to sense the changing real-world terrain characteristics to (re)plan their tactical and strategic movement actions. The changeable nature of the represented terrain regions along with the targeted terrain attributes, whether they can be sensed by the agents either as obstacles or not, has a crucial impact on the resulting agents' models. An important aspect is the fact that the computations supporting these changes may impair the fluency and realism of the simulations if conven-

[a] https://orcid.org/0000-0002-6025-5270
[b] https://orcid.org/0000-0002-2388-895X
[c] https://orcid.org/0000-0001-7988-4141
[d] https://orcid.org/0000-0003-3228-4071
[e] https://orcid.org/0000-0003-4655-8889

Silva, L., Nascimento, E., Zacarias, E., Nunes, R. and Freitas, E.
Hierarchical Terrain Representation and Flood Fill-based Computation of Large-Scale Terrain Changes for Agent-based Simulations.
DOI: 10.5220/0011134800003274
In *Proceedings of the 12th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2022)*, pages 15-25
ISBN: 978-989-758-578-4; ISSN: 2184-2841

tional regular grid-based map representations and no other advanced technologies are used, thus negatively impacting the systems' responsiveness (Nielsen 1994) and the desired agent-based simulation realism.

From a preliminary discussion presented in (Nascimento, Zacarias et al. 2019) and the proposition of a distributed service-oriented multi-agent simulation architecture in which the proposed techniques are inserted as terrain simulation services, the contributions of this work are:

(1) The hierarchical approach to efficiently represent real-world terrain maps for agent-based simulations. To enable the development of simulations running over large-scale dynamic terrain maps captured in different levels of representation detail, this work proposes the use of a hierarchical QuadTree representation (e.g. (Brondani, de Lima Silva et al. 2019)) to store data used to compute changes caused by the effects of dynamic events. This work innovates by proposing the use of this representation to allow the optimized computation of the changing effects due to simulated natural phenomena in the terrain map;

(2) The use of Flood Fill-based techniques (also known as Seed Fill algorithms (Glassner 1990)) to execute extensive changes in the QuadTree. With that, the general consequences of river floods and wildfires, for instance, are represented on large areas of the terrain maps. The implemented algorithms guide the advance and retreat of the river floods and the wildfires, and the efficient blocking and releasing of nodes processed by the simulated agents' algorithms;

(3) The proposition of a dictionary-based memory for the storage and the restoration of not only the nodes but also the branches of the QuadTree that were subjected to the terrain changes. The proposed solution explores the changing of terrain map areas that should not simply be lost/not properly reused after they were processed. That is because the restoration of the hierarchical terrain structure to alternative representation states has a meaningful impact on the realistic execution of agents' planning and navigation tasks.

(4) The analysis of different kinds of tests to assess the enhanced performance of the proposed techniques. The tests present computing time results for increasing levels of changes due to river floods and wildfires in three different real-world terrain map models. The results are obtained with the use of the QuadTree representation approach, and they are contrasted with similar testing results obtained when a regular grid structure is used in the same maps.

2 BACKGROUND AND RELATED WORK

The exploration of irregular map representation topologies (Algfoor, Sunar et al. 2015) for the modelling of virtual terrains is important for the development of simulation systems to real-world application problems. While the regular grids with representation nodes of the same size are often used in the capture of small-scale terrains mostly for computer games, the irregular grids are formed by nodes with distinct dimensions and shapes.

To capture real-world terrain areas in the simulation system, map cells are often modelled according to information extracted from real-world GIS-based maps. Depending on the level of detail that such maps are produced, the storing and use of such terrain features usually results in a high computational cost for the Artificial Intelligence (AI) algorithms supporting the agent simulations. This is true not only when one considers the simulated agents and their relationships but also when such costs are due to the "Environment-Environment" interactions (Hawe, Coates et al. 2012) between the simulated terrain objects.

Simulations based on dynamic terrain scenarios have a highly positive impact on the quality of the simulation-based training activities in many application problems. For instance, (Pfeiffer and Tamash 2014) describe a system that allows the simulation of combat operations. They seek to explore the realism in the physics of the simulated environment, the use of climate changes and their effects, leading to operational problems to be approached by the agents deployed in the virtual scenario. Simulated natural phenomena are presented in (Liang, Du et al. 2008), where the algorithms simulate rescue situations due to terrain floods. Once the terrain changes occur in the map structure, simulated agents (and users who may be in control of these agents) are prompted to take new decisions along with the development of the simulations. An algorithm to model the propagation of wildfires is presented in (Rui, Hui et al. 2018). This algorithm is applied to a non-hierarchical grid structure that represents the virtual terrain. The grid cells capture state information to represent the cell situation (e.g. its burning state). Computational techniques to dynamic virtual (synthetic) environments are presented in (Smelik, van Wermeskerken et al. 2018). To build these dynamic terrains, a set of constraints are considered: performance when working with terrains in different levels of representation detail, simulation realism and control of dynamic events occurring in the terrain environment, among others. Examples of terrain deformations are the trails left by vehicles, the erosion effects caused by heavy rains or river dam ruptures, the explosion of buildings and bridges, and others. All these events can have an impact on the terrain representation, and the consequent computations guiding the simulation of various kinds of agents' movement behaviours.

Among the works that explore the hierarchical representation of the terrain structure, the Dynamic Probabilistic QuadTree technique is described in (Cocaud and Jnifene 2010). There, each node contains information about the probability of an obstacle being located in it. The QuadTree node expansion and shrinking operations are based on the alteration of the probability value of the nodes affected by the computed terrain changes. (Liang, Du et al. 2008) explore the adaptive QuadTree representation in the modelling of the surface water runoff due to floods. In (Wouter G. van Toll, Atlas F. Cook IV et al. 2012), the costs due to the computation of alterations in virtual terrain structures are investigated. The situations that modify the virtual terrain map are the insertion of obstacles in the form of point, line, or polygon, where all of them consider the polygonal characteristics of the *NavMesh* representation structure. All in all, these works discuss techniques that support the modification of the represented terrain topology. However, they mostly focus on the analysis of small-scale and almost always local terrain changes while many real-world simulations ought to be based on large-scale terrain areas in which the topology of the represented world could change significantly over short periods of time.

3 THE SIMULATION SYSTEM ARCHITECTURE

The proposed architecture for the simulation system (Fig. 1) is based on a distributed service-oriented multi-agent approach in which simulation services are used by simulation systems' clients over the web (Byrne, Heavey et al. 2010). At its core, there is a Terrain Simulation Server that receives as input real-world GIS-based terrain maps to create a virtual model for the various levels of represented terrain information. This is a process that generates the various navigation maps for the simulated real-world terrains.

To develop dynamic terrain simulation services to equip this simulation proposal, this work focuses on the representation and prompt modification of the structure of the terrain navigation maps used to guide the agent-based simulations. While the terrain representation structure by itself can be indexed by different representation layers, such layers have the overall aim of capturing the targeted terrain features for the simulation purposes. Then these features are hierarchically represented to optimize the computations of the simulation exercises. With that, the status of the altered terrain map attributes is distributed to/shared by the Simulation Clients, where terrain scenarios can be visualized in different forms by these client implementations.

The architecture is composed of core components supporting the large-scale terrain representation and the basic functionalities for the terrain map manipulation. The Hierarchical Terrain Representation component, which is implemented as a QuadTree, is responsible for the basic elements that compose the terrain navigation map. The hierarchical and irregular grid topology of the QuadTree allows the optimized representation and computation of large-scale maps. In it, the deeper the node is, the higher the level of terrain representation detail is. The use of this topology allows refining the representation structure to capture the terrain features of interest for the simulations, where alternative features can be considered: the shapes of the terrain landscape, the steepness of these terrain features, and the location of rivers and other bodies of water, among others.

The component capturing the Terrain Simulation Algorithms is responsible for providing the basic algorithms to the management of the elementary parts of the terrain structure, i.e. the nodes that represent the hierarchical terrain map. These algorithms are used to implement specific Simulation Services that provide large-scale changes in the terrain map, such as blocking/releasing nodes affected by river floodings and wildfires. In the proposed approach, the alteration of terrain representation structure is computed according to a set of reusable flood fill-based operations. It means that the computing techniques regarding graph flooding are explored to guide the analysis and alteration of the represented areas (nodes) of the terrain map. Such procedures are directed by the QuadTree refinement, resulting in the expansion and retraction of the hierarchy levels. Similarly, terrain restoration operations are implemented, aiming to return the QuadTree structure to its previous state. The restoration procedures can also be concerned with the maintenance of the altered QuadTree nodes, which are saved into and retrieved from a Dictionary-Based Memory.

The Dictionary-Based Memory manages terrain information that is redundant to the QuadTree.

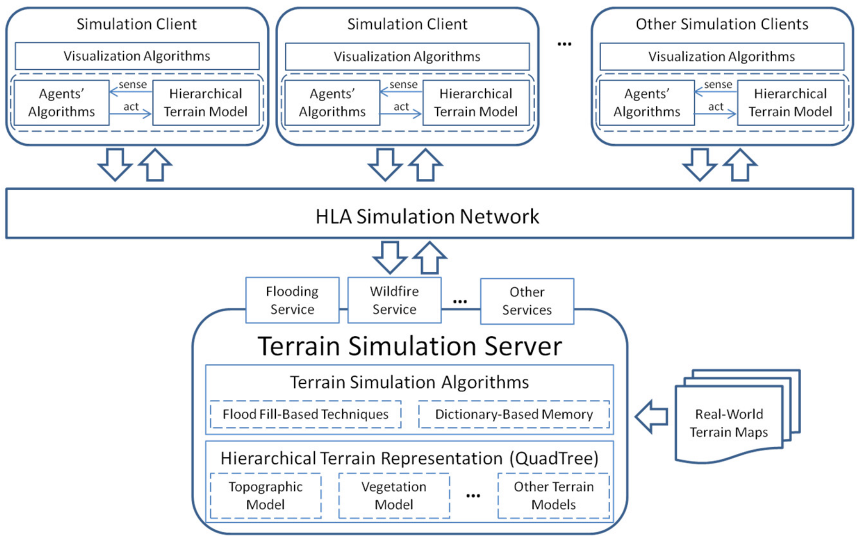


Figure 1: A distributed service-oriented multi-agents' simulation architecture.

However, it allows the implementation of simulation terrain services that permit executing efficient expansion and retraction operations on the hierarchical terrain map representation. In practice, the terrain information leaves the QuadTree when the changes in the terrain map nodes are materialized. So the degree of terrain detail represented in the QuadTree is reduced. Despite the fact the simulated terrain map ends up being represented in a lower granularity, the modified terrain map information is maintained in the dictionary for future (re)use. In later computations with it, the changed terrain information doesn't need to be searched in the QuadTree. This memory is implemented as a *HashTable* organized as tuples (key, value), where a query to it can be executed. This query retrieves the represented QuadTree nodes (or entire subtrees) changed due to the simulated natural phenomena. That restoration also permits the reconstruction of the QuadTree in different forms.

Several Simulation Clients can access the Simulation Services provided by the above-described server via a HLA Simulation Network (e.g. (Lees, Logan et al. 2006)). At each client, a multi-agent simulation engine operates over the locally stored hierarchical representation of the simulated terrain map, sensing and acting on it. The simulated agents implement different types of algorithms, e.g. global and local path planning, which need to collect data from the simulated environment. Then they select the best course of actions to perform a given realistic agent behaviour in the targeted simulations. In practice, a Hierarchical Terrain Model component captures a view of the terrain map data at the clients' systems, and uses the Simulation Services provided by the Terrain Simulation Server to perform large-scale changes in the hierarchical structure that compose the map. These changes affect the perception of the environment sensed by the simulated agents, thus also their actions. They can also guide the visual appearance of the simulated environment which is presented by the alternative Visualization Algorithms implemented in each Simulation Client.

The structural changes on the hierarchical terrain representation analyzed in this work are motivated by the advance and retreat of floods along riverbanks and the terrain changes occurring due to wildfires. Those changes are computed in different representation layers of a simulated terrain. In practice, while the simulations are running in this dynamically changing environment, for instance, the simulation instructor can insert these events into the simulation exercises. Such action can cause agents' evasive actions to avoid the flooded areas, the construction of river dams, and the development of other mitigation tasks in the simulations. As approached in this work, similar agents' re-planning computations can also be required due to large-scale wildfires. From the dynamic nature of the used simulation scenarios, practical experience with simulations indicates that users of the implemented simulation systems can be subjected to multi-agent tactical and strategic simulation problems to improve their decision-making capabilities.

4 REPRESENTATION AND COMPUTATION OF LARGE-SCALE TERRAIN CHANGES

In the construction of the QuadTree, the terrain height feature can be used in the process of subdividing the hierarchical representation in its different levels can be explored (see examples of it in Fig. 1-3). There, relief heights are captured in different levels of detail. In doing so, there is an "inheritance" of terrain height values from the father to the child nodes, where the child nodes just refine the maximum and minimum terrain heights represented at the father node. A new level in the QuadTree hierarchy is built whenever one of these terrain features is identified in the analysis of the used real-world GIS-based maps. This process is repeated until the final QuadTree refinement condition is met. In the end, the irregular representation of the terrain is expressed by QuadTree "leaf" nodes.

To model the terrain map structure, a directed graph is captured as a *Tree <n, maxLevel, root>*. There, a QuadTree node *n* consists of *n <parent, children, level, isLeaf, refined>* such that *n.parent* and *n.children* are pointers used by the hierarchy; *n.level* indicates the level of the hierarchy to which the node *n* belongs to; *n.isLeaf* indicates whether the node is a leaf, and *n.refined* indicates that new nodes were generated from the node *n* on a temporary basis. *Tree.maxLevel* indicates the maximum hierarchical level where the irregular grid is captured by the QuadTree leafs. Attributes required for the computation of terrain effects due to river floods and wild fires are modeled as *n <minHeight, maxHeight, neighbors, hasRiver, walkable, flooded, vegetationDensity, burned>* such that *n.minHeight* and *n.maxHeight* capture the minimum and maximum terrain height values inside the terrain region represented by the node *n*; *n.neighbors* consist of a list of neighbor nodes of node *n*, permitting to capture the connection between the adjacent nodes; *n.hasRiver* indicates the presence of rivers in the terrain area covered by node *n*; *n.walkable* indicates whether the node *n* is navigable by simulated agents; *n.flooded* indicates whether the node *n* was subjected to terrain alterations as a result of the flood fill-based computations;. *n.vegetationDensity* indicates the amount of trees inside the terrain region represented by the node *n*; and *n.burned* indicates whether the node *n* was subjected to the fire alterations.

The use of the flood fill-based technique along with this QuadTree representation occurs as follows. An initial node is removed from a control queue. Then the flooding test is performed on it. This permits to identify (un)blocked nodes that satisfy a "flooding condition". Such nodes can be blocked for the terrestrial agent movement, where the condition for it is that (a) the terrain node represents a relief height below a certain flood-prone height (i.e. a river flood limit) and (b) the node captures a terrain region that is located next to the course of rivers. In doing so, the algorithm considers the maximum and minimum terrain relief heights. These are there in each QuadTree node as shown in Fig. 2 (this figure illustrates some of the steps of the proposed solution through an example of a 5m river flood).

The neighbouring nodes of a flooded one are analyzed. When it happens, the flood limit can be a) above the maximum terrain height, b) below the minimum terrain height, and c) between the maximum and minimum terrain height. The nodes with maximum terrain heights below the flood limit are identified and added to the control queue. A node is removed from this queue and the algorithm checks whether its parent also meets the flooding condition. If so, further parent nodes upwards in the hierarchy are tested.

This process continues until a QuadTree node not meeting the flooding condition is found. With the non-leaf node satisfying the condition, the subtree rooted on it is removed from the QuadTree. Then it is added to the dictionary. The flooded node is transformed into a leaf one. That ends expressing a larger flooded terrain area in the QuadTree. This process stops when the control queue is empty. As an example, in Fig. 2, the subtree rooted on node 2 is memorized into the dictionary and the QuadTree structure is simplified after this process.

The restoration of the terrain structure consists of rebuilding the QuadTree to a targeted state. For instance, the reconstruction allows simulating the return of the flooded water to the original river bed. When analyzing an altered node, the algorithm checks whether it was previously refined. For example, large terrain areas could have been subdivided in smaller pieces, where only a few of these subareas were flooded. To do so, the alteration algorithm checks the "refined' attribute represented in the QuadTree node. If the node was refined, the restoration procedure carries out the (re)grouping of the terrain representation nodes generated due to the QuadTree refinement.

The terrain structure can also be returned to a partial representation state, permitting to compute the partial flood retreat, for instance. This occurs when the restoration condition allows a portion of

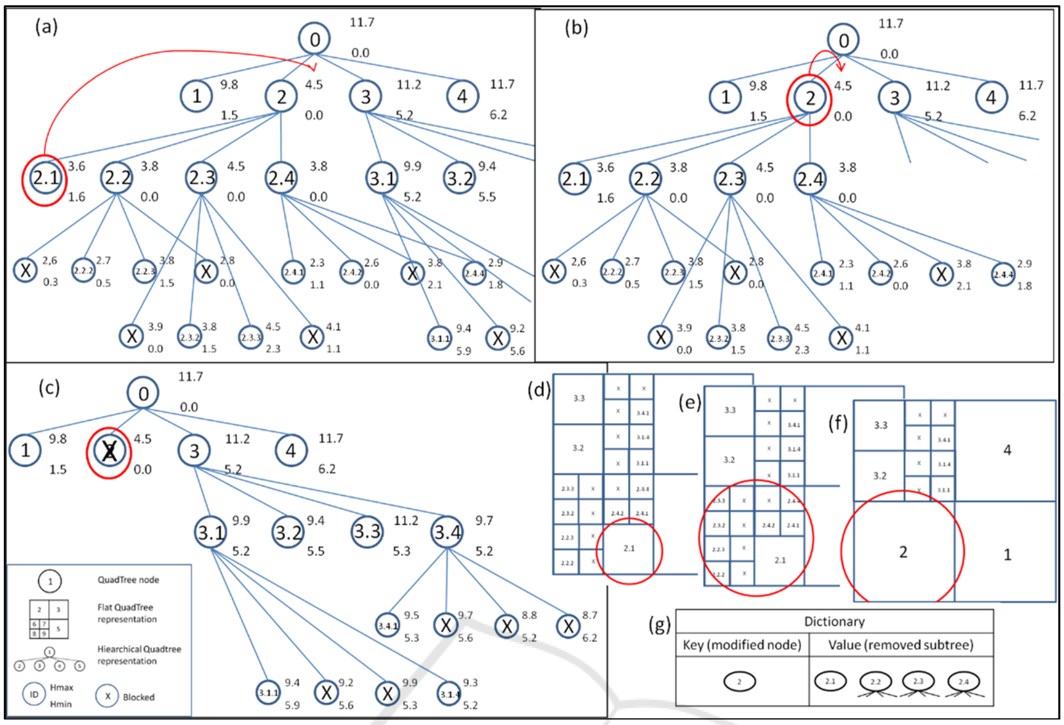


Figure 2: River flood alteration steps computed in the QuadTree: (a) and (d) the 5.0m river flood is above the relief height represented by the node; (b) and (e) this flood is also above the height represented by the parent node; (c) and (f) all the terrain area covered by the parent node is flooded – a single larger size and blocked terrain area is represented in the QuadTree; (g) the flooded subtree is saved in the dictionary.

the terrain structure not to be restored, while other parts of it are kept into the dictionary.

The restoration can also result in the generation of new terrain subareas (i.e. new nodes in the QuadTree). That is because the computed flooding may have caused the grouping of these terrain nodes, where this grouping ended expressing a single larger terrain area in the QuadTree representation. To restore terrain subareas from the dictionary, the node that rooted the affected subtrees is used. It is kept in the QuadTree even after the terrain changes were computed, guiding the search of a desired terrain subarea stored in the dictionary.

As illustrated in Fig. 3, the algorithm analyzes the minimum and maximum heights of each restored node from the dictionary. If the maximum height recorded in these nodes is below the flood retreat level, the nodes remain blocked. Otherwise, if the minimum height is above the retreat level of the flooding, the nodes are defined as unblocked.

In the process of releasing flooded nodes, when the node under review has a subtree stored in the dictionary, that subtree is restored and included in the QuadTree. After restoring the node in which the flooding retreat level is between the maximum and minimum terrain relief heights, it is necessary to refine that node to examine its possible children. These terrain subareas can have a maximum height below the flooding retreat level, indicating that they should be blocked for terrestrial navigation. And if they have a minimum height above the flooding retreat level, they are set up as unblocked. Such refinement is also developed as long as the maximum QuadTree refinement level is not reached.

The aimed vegetation characteristics of a simulated real-world terrain can also be represented in the QuadTree. Similar to the terrain relief representation, the hierarchical representation is used in the indexed generation of the vegetation layer for the simulated terrain. The computation of changes in the terrain representation structure as to express the effects of wildfires makes use of the numerical value representing vegetation density characteristics in the QuadTree. When a parent node representing a terrain area has a certain vegetation density, its children nodes may not capture the same density represented by its parent. That is because such vegetation can be localized in a smaller subarea of the entire area covered by the parent node. Therefore, to change the blocking status of an internal QuadTree node due to wildfire effects, it is necessary to analyze all children nodes from the current one.

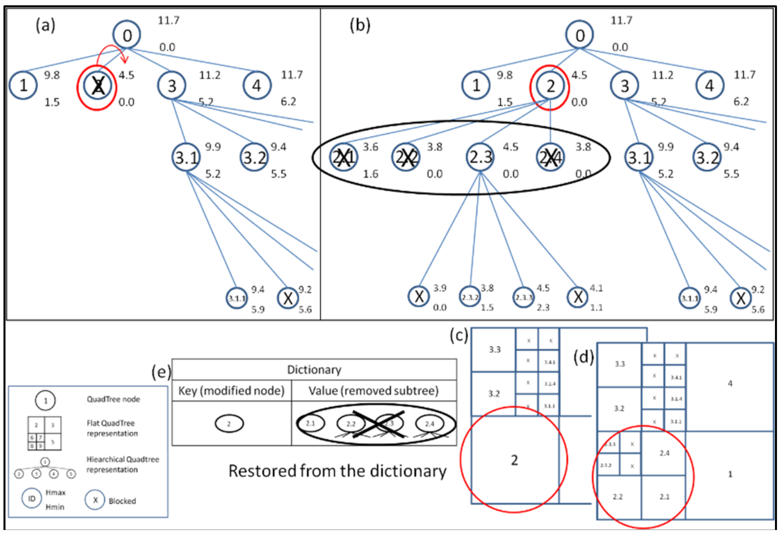


Figure 3: Restoration steps in the QuadTree: (a) and (c) the 1.0m river flood retreat is below the relief height represented by the flooded node; (b) and (d) the flooded subtrees are restored from the dictionary: one of these subtrees returned to its original state, and the other three remained flooded – they represent larger-size and blocked terrain areas in the QuadTree.

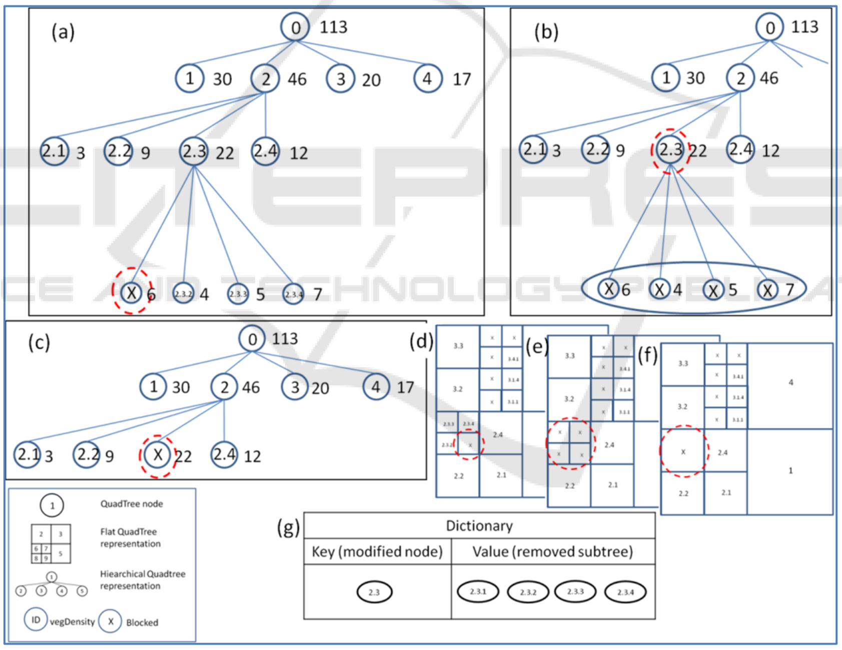


Figure 4: Wildfire alteration steps computed in the QuadTree: (a) and (d) a vegetation density (e.g. 10) permitting the current wildfire propagation is higher than the density represented on the leaf node (e.g. a fire ignition point); (b) and (e) this density is not higher than the one represented by the parent node; (c) and (f) all the terrain area covered by the parent node is blocked due to the wildfire; (g) the subtree rooted on the parent node is saved in the dictionary.

Different from the river flooding computations, the QuadTree alterations due to wildfire effects are implemented as a bottom-up procedure. The algorithm starts at a certain terrain location representing an initial fire ignition point (see Fig. 4). Then it examines the neighbouring nodes of the first node affected by the fire. Nodes that satisfy the wildfire condition are added to the queue that controls the algorithm execution. A node is removed from the queue, and it is marked as burned. Then, the algorithm checks whether all siblings of the current node have been marked as burned. If so, the node along with its siblings are removed from the QuadTree, and their parent node is marked as burned. Then this parent is turned into a leaf node. In addition, it is analyzed whether all the siblings of the node turned into a leaf are burned. When these siblings capture larger terrain areas, these nodes are refined and the smaller ones are added into the QuadTree. It happens whenever the maximum QuadTree refinement level is not reached. If all these children nodes satisfy the fire propagation condition, the parent node is marked as burned and the QuadTree refinement is undone. Otherwise, only the children nodes satisfying the wildfire condition are blocked.

The analysis continues until the algorithm does not find a node in which its siblings have been subjected to fire. Unlike the retreat of river floods, we have not implemented a short-term natural phenomenon regarding the restoration of the terrain structure from the wildfire effects. However, it is possible to restore the terrain representation map to the state prior to the computation of the wildfire changes, restoring all the performed QuadTree refinements.

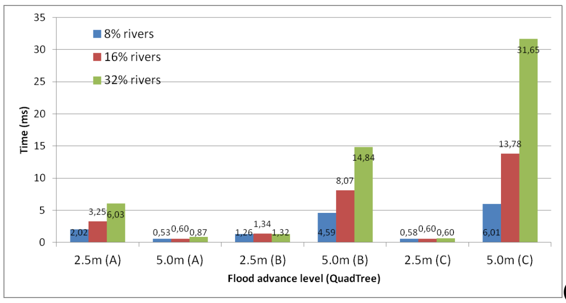
5 EXPERIMENTS AND RESULTS

The goal of the experiments was to analyze the effort to compute structural alterations in the QuadTree used in the representation of different terrains. The following terrains were used: terrain (A) with 64 km² and a QuadTree with 10 levels, terrain (B) with 256 km² and a QuadTree with 11 levels and terrain (C) with 1024 km² and a QuadTree with 12 levels. In all tested terrains, the deepest "leaf" nodes in the QuadTree represent a real-world terrain region of approximately 64 m². For comparison purposes, versions of each terrain were represented with the use of a regular grid structure with 64 km², 256 km², and 1024 km², where grid cells capture terrain regions with approximately 1024 m². The terrain structures used in the tests have an increasing number of areas that are susceptible to changes caused by river floods and wildfires. In this case, 8%, 16%, and 32% of the total area of these terrains have areas covered by rivers, while 16%, 31%, and 64% of their areas are covered by forests. Different levels of floods were used in the experiments: 2.5m and 5.0m above the level of the represented rivers. Similarly, different wildfire burning effects were computed in these terrain structures. Finally, the relief characteristics and the distributions of fire-prone areas (i.e. forests) in the tested terrains are different, where the larger-size terrains are not simply larger versions of the smaller ones. This should be noted when drawing conclusions from the test results. The algorithms were executed 50 times in each test scenario. The tests were executed in the Unity 3D engine (Unity Technologies 2018). With many relevant industrial applications, that is the engine used in the development of computer games and simulation systems. An Intel I7 4790 processor with 10 GB RAM was used to perform the experiments.

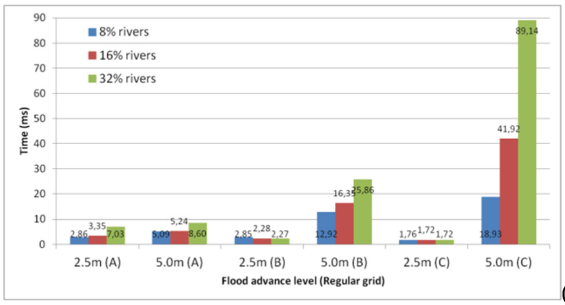
Regarding the advance of the river floods, the highest computing time to execute the terrain structure alteration algorithms was 31.65ms with the QuadTree and 89.14ms with the regular grid (Fig. 5). Thus the execution time of the algorithms with the QuadTree was much lower than the time required to performing similar alterations with the regular grid. This is mostly due to the blockage of internal QuadTree nodes as a consequence of the floods, causing the blockage of a larger number of "leaf" nodes (when blocking a root node of the QuadTree, its children are also blocked). This happens when all children nodes represent terrain areas with relief elevations below the flood level being computed.

Regarding the retreat of the floods, the highest computing time was 9.64ms with the QuadTree and 2.48ms with the regular grid (Fig. 6). Although the retreat of the floods with the QuadTree presented a higher computing time than with the regular grid, the computation of the flood retreat consequences was much slower than the computations for the advance of the floods.

First, tests were performed to assess the river flood advance computations. For the 2.5m and 5.0m flood advance computations (Fig. 5), the time to compute the 2.5m flood consequences was smaller than the time for computing 5.0m flood effects. That was observed when both the QuadTree and the regular grid structures were used. In case of small amplitude floods (i.e. 2.5m), the execution of the implemented algorithms show that most of the time the nodes analyzed are adjacent to the ones representing the rivers, where small-size affected areas do not require the analysis of the internal nodes of the QuadTree (i.e. the nodes representing larger pieces of terrain). Thus the floods affecting small areas of the terrain structure do not result in a thoughtful exploration of the QuadTree hierarchy, increasing the execution time of the algorithms. That did not happen when the 5.0m floods were computed in the terrains (B) and (C). However, there are exceptions to this pattern: due to the topography of the terrain (A), the time to compute 5.0m flood effects in it was smaller than the time to compute similar 2.5m floods.

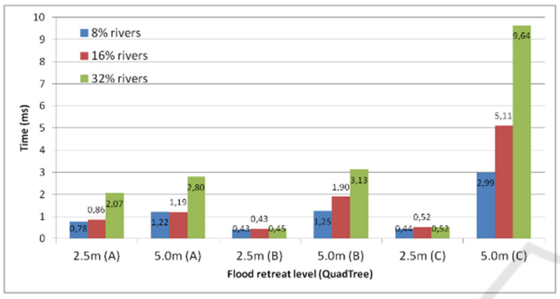


(a)

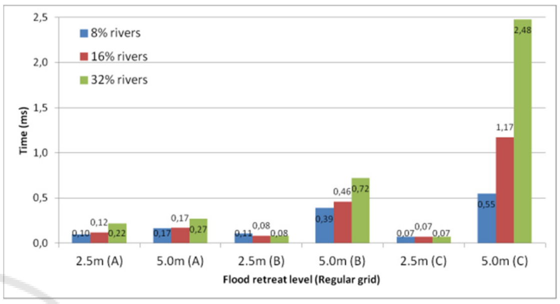


(b)

Figure 5: Execution time results of river flood advance computations for terrains (A), (B) and (C).



(a)



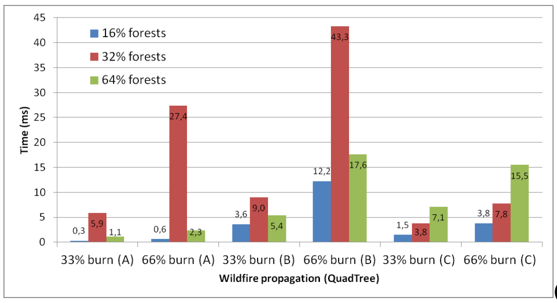
(b)

Figure 6: Execution time results of river flood retreat computations for terrains (A), (B) and (C).

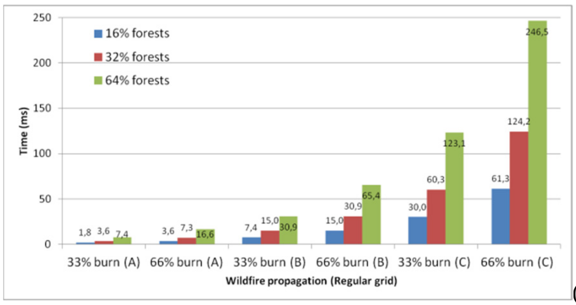
Fig. 5 shows that the increase in the percentage of rivers from 8% to 32% in the terrains (B) and (C) did not present a relevant increase/decrease in the time required to compute the 2.5m flood effects in these terrains with the QuadTree and the regular grid structures. Conversely, the increase of the percentage of rivers increased the time to compute the flooding consequences when the terrain (A) was used. For the 5.0m computations with the QuadTree and the regular grid structures, the time to compute the flood effects increased with the increase of the percentage of rivers in the terrain. Similarly, the time for this kind of flood advance computations also increased with the increase of the size of the terrains, even considering that the tested terrains have different topographic characteristics. That increase in the computing time was not related to the increase of the terrain size when the 2.5m flood advances were executed in the terrain structures.

Second, tests were performed to assess the performance of the flood retreat algorithms. In doing so, a previously computed flooding was applied to the tested representation structures. The results presented in Fig. 6 shows that the time to execute such flood retreat consequences in the QuadTree was higher than the time to compute similar aspects in the regular grid. For the retreat of the 5.0m floods, the time for these computations increased with the increase of the percentage of rivers in the terrains when using the QuadTree and the regular grid structures. That increase of the computing time with the increase of the percentage of rivers was also observed when 2.5m flood retreats were executed in the terrain (A) representation structure even though it was not observed in the terrains (B) and (C) with the QuadTree and the regular grid representations. For the 5.0m flood retreat computations, the computing time increased with the increase of the size of the terrains from (A) to (C). For similar 2.5m flood retreat computations, the computing time with the terrain (A) was higher than with the terrains (B) and (C). In each individual terrain, the time to compute the effects of the flood retreat in the QuadTree and the regular grid structures increased when the flood retreat varied from 2.5m to 5.0m. The 5.0m flood retreat computations were slower than the similar 2.5m flood retreat computations.

Experiments of wildfire propagation effects were also conducted. The highest computing time was 43.3ms with the QuadTree, while it was 246.5ms with the regular grid. There are several clearings inside the forests represented in the hierarchical QuadTree structure of the terrain. To faithfully represent these areas, a larger number of "leaf" nodes in the QuadTree were used, generated at deeper hierarchical levels. This larger number of "leaf" nodes is analyzed by the fire propagation algorithm, increasing the execution time of the structural terrain altera tions. In this case, it is often necessary to refine the large-size nodes representing these clearings, where



(a)



(b)

Figure 7: Execution time results of wildfire propagation computations for terrains (A), (B) and (C).

this refinement implies a better detail of the burning/fire consequences simulated. Despite these facts, the computing times were much higher when the terrains were represented with the regular grid structure in all tested scenarios.

Regarding the use of regular grids (Fig. 7), a pattern of increasing computing times was related to the increase of the burned area from 33% to 66%, the increase of fire-prone areas from 16% to 64%, and the size of the terrains from (A) to (C). However, a similar pattern was not observed when the terrains were represented with the QuadTree. Terrains (A) and (B) with the QuadTree representation have a similar computing time pattern related to the increase of fire-prone areas from 16% to 64%. In these terrains, the time to compute the fire propagation effects was higher when 32% of the burning area of the terrains (A) and (B) with the QuadTree were analyzed. That pattern appeared even when the burning areas was varied from 33% to 66% in these terrain structures. An exception to it was the terrain (C), where the computing time increased with the increase of the fire-prone areas of this terrain from 16% to 64% and the burned areas from 33% to 66%.

In summary, the increase of the percentage of rivers and the percentage of fire-prone areas in the tested terrains did not necessarily increase the time to change the terrain structures due to the flood advance and wildfire-propagation computations executed with the QuadTree. However, the increase of the computing time for these terrain structure alterations was related to the amplitude of the required floods and percentage of the burned areas, considering the tested 2.5m and 5.0m flooding and the 33% and 66% burning consequences. A similar pattern was observed when the effects of the flood retreat were computed in the tested terrain structures. Not surprisingly, the size of the terrains also had a contribution to the required computing time to change these terrain structures accordingly.

6 CONCLUDING REMARKS

The use of ABMS is a valuable asset to low-cost, low-risk training and instruction activities in different application fields. In the context of a distributed simulation architecture in which simulation services are used to produce dynamic terrain maps for agent-based simulations, this work proposes the hierarchical QuadTree representation for the modelling and computation with large-scale, real-world terrains. The approach permits to efficiently use the terrain information captured in the representation structure, and to make computations with it to express the terrain changes in the hierarchical terrain model. This work shows how to do it with the use of flood fill-based techniques, which are applied to the analysis and alteration of the QuadTree along with the exploration of a dictionary-based memory. Based on two different case studies, ways of working with the hierarchical structure are discussed, and the effectiveness of the proposed techniques is demonstrated. In addition to tests with other terrains, future works can involve the use of the proposed techniques in the computation of terrain map effects due to the occurrence of other natural phenomena.

ACKNOWLEDGEMENTS

We thank the Brazilian Army Strategic Program ASTROS for the financial support through the SIS-ASTROS GMF project (898347/2020).

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: 11.
Brondani, J. R., L. A. de Lima Silva, E. Zacarias and E. P. de Freitas. 2019. Pathfinding in hierarchical representation of large realistic virtual terrains for simulation systems. *Expert Systems with Applications* 138: 112812.
Byrne, J., C. Heavey and P. J. Byrne. 2010. A review of Web-based simulation and supporting tools. *Simulation Modelling Practice and Theory* 18(3): 253-276.
Cocaud, C. and A. Jnifene. 2010. *Environment mapping using probabilistic quadtree for the guidance and control of autonomous mobile robots*. International Conference on Autonomous and Intelligent Systems, AIS 2010, Povoa de Varzim, Portugal, IEEE.
Glassner, A. 1990. Graphics Gems. San Francisco, CA, Morgan Kaufmann Publishers Inc.
Hawe, G. I., G. Coates, D. T. Wilson and R. S. Crouch. 2012. Agent-based simulation for large-scale emergency response: A survey of usage and implementation. *ACM Computing Surveys (CSUR)* 45(1): 8.
Lees, M., B. Logan and G. K. Theodoropoulos. 2006. Agents, games and HLA. *Simulation Modelling Practice and Theory* 14: 752–767.
Liang, Q., G. Du, J. W. Hall and A. G. Borthwick. 2008. Flood inundation modeling with an adaptive quadtree grid shallow water equation solver. *Journal of Hydraulic Engineering* 134(11): 1603-1610.
Macal, C. 2016. Everything you need to know about agent-based modelling and simulation. *Journal of Simulation* 10: 144–156.
Nascimento, E. J., E. Zacarias, D. M. Doebber, E. P. de Freitas and L. A. de Lima Silva. 2019. *Flooding-Driven Modifications of a Hierarchical and Irregular Navigation Grid Structure for Large Virtual Terrains used in Simulation Systems*. 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), IEEE.
Nielsen, J. 1994. Usability engineering, Elsevier.
Pfeiffer, K. D. and T. Tamash. 2014. *Measuring the impact of natural environment representation on combat simulation outcomes*. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), Orlando, FL.
Rui, X., S. Hui, X. Yu, G. Zhang and B. Wu. 2018. Forest fire spread simulation algorithm based on cellular automata. *Natural hazards* 91(1): 309-319.
Smelik, R., F. van Wermeskerken, R. Krijnen and F. Kuijper. 2018. Dynamic synthetic environments: a survey. *The Journal of Defense Modeling and Simulation*.
Unity Technologies, I. 2018. "Software Unity®."
Wouter G. van Toll, Atlas F. Cook IV and R. Geraerts. 2012. A navigation mesh for dynamic environments. *Computer Animation and Virtual Worlds* 23(6): 535-546.