Grid Based Sensing Model for Autonomous Ship Agent

Pan Mingyang*, Masaaki Inaishi **
Zhao Xiaoyu*** and Akira Kawaguchi****

Abstract

The cost of computation becomes a bottleneck for an agent based marine traffic simulation. This is especially true when investigating interferences of a large number of ships and static objects placed in the sea. In general, a collection of ship agents built with a simple sensing model mandates a quadratic time complexity of $O(n^2)$ for sensing each other. The runtime cost required for sensing the environment sharply increases and the system performance rapidly deteriorates, as the number of environment elements increases. This paper presents a grid based sensing model, a new approach for building an autonomous ship agent. The aim of this research is to address the issue of time and space complexities. The proposed model reduces the cost of time and space for running the marine traffic simulation system implemented with the previous work of a ship cluster behavior model. Analysis of the grid based sensing model and its overall effects based on simulation experiments are discussed.

Keywords: Marine traffic simulation, ship agent, grid based sensing model

1. Introduction

Traffic is commonly viewed as a complex system. The formal method for modeling complex systems, such as cellular automata and agent, is considered as a primary means for capturing traffics on the road and sea, in which a large number of agents behave like vehicles and ships and their interactions yield traffic flows in turn.

A series of researches on marine traffic simulation were carried out in the past for the analysis of a ship cluster behavior model (1) - (3). This cluster behavior model is indeed a complex system made up of reactive agent. For the full investigation of the characteristic movement of ship clusters in the congested waterways, an intelligent marine traffic simulator has been developed (4). This simulator has incorporated an agent-based model to realize a captain’s decision-making process with abilities for each ship to maneuver the navigation in narrow areas without colliding with other ships or grounding in the sea. Further refinement has led to the building of a more autonomous, agent-oriented maritime traffic simulator. Each ship agent is now enabled to perceive the status of surrounding environment and to adjust into the optimal course to accomplish the passage plan (5). All of these efforts rely on the complex system, especially, agent-based methodology.

The cost of computation becomes a bottleneck for the agent based marine traffic simulation when a large number of ships and static objects are placed in the sea. The ship agent model has three equally important functions: sensing, decision-making, and effecting. Decision-making is a vital part of the simulation, thus drawing most attentions to its functional efficiency. However, a sensing function, if poorly designed, grabs a large amount of computation cost. For instance, simple sensing is based on the pair-wise interactions between the ships and static objects in the system. The time and space required for this kind of sensing sharply increases and the overall performance rapidly deteriorates as the number of the elements in the system increases.

*Student member Dalian Maritime University, 116026, 1 Linghai Road, Dalian, panmingyang@sina.com
**Member, ***Student member Tokyo University of Marine Science and Technology
****Non-member The City University of New York
This paper presents a grid based sensing model, a new approach for building an autonomous ship agent. The aim of this research is to address the issue of time and space complexities of the sensing process of ship agent. The proposed model reduces time and space cost for running the marine traffic simulation system implemented based on the previous work of a ship cluster behavior model. This new model does not compromise the accuracy of sensing. Analysis of the grid-based sensing model and its overall effects based on the simulation experiments are discussed.

2. Simple sensing model

The environment or framework of agent models provides a basis for the interactions of a collection of autonomous elements defined in terms of parameters to express their behaviors and characteristics. The defined autonomous individuals, also referred to as intelligent agents, continuously perform the following functions: (1) perception of dynamic conditions in the environment, (2) reasoning to interpret the perceptions, draw inferences, and determine actions, (3) and acting to affect conditions in the environment. The ship agent defined for the agent based marine traffic simulation repeats the same kind of sense-decide-act cycle, that is, perceive the environment through sensors, making decision, and acts upon environment through effectors. A simulated sensor scans the environment around the ship, similar to the work done by the real sensors such as radar, ARPA, and AIS, but in a limited range of surrounding area to detect other ships and objects that could influence the own behavior of the agent.

A simple sensing model (SS-model for short) for the ship agent is to calculate the distances between the own ship and other elements located in the simulated sea area. It thus finds all the ship agents and objects existing within a radius. This is the model adopted for the previous work of the marine traffic simulation system built on top of the ship cluster behavior model. Obviously, the SS-model is not tailored for the optimal performance. The model is rather primitive and costly due to the expensive calculations of distances to all the elements in the environment. A great amount of computation is involved even finding out only a few elements potentially affective to the agent behavior.

Specifically, the SS-model mandates a quadratic time complexity of \(O(n^2)\) for \(n\) agents to sensing with each other. It needs \(O(mn)\) time complexity to sensing static objects. In the paper, \(n\) and \(m\) are the numbers of ship agents and static objects. A class of moderately large-scale simulation analyzes the environment of those elements at least in the magnitude of hundreds. Because of this kind of pair-wise sensing mechanism imposed to each of the ship agents and static objects, the overall cost for sensing grows into a dominant part of the simulation computation.

3. Grid based sensing model

3.1 Spatial data structure

The improvement of the efficiency of the sensing model comes from the use of one type of spatial data structures for holding the elements in the simulated environment. The revised model is called a grid based sensing model (GS-model for short). The GS-model extends the discrete data structure of spatial lattice of cellular automaton traffic model (CA-model for short) as detailed in this section.

See Fig. 1 for the spatial data structure utilized in the GS-model. Similar to the CA-model, the simulation area is divided into square lattice. Notice that the lattice will accelerate the sensing of agents, not restrict the flexibility of the agent behaviors. Each point in the lattice is called a grid, as marked “Grid-i” in the figure. One grid can represent more than one ship, in contrast to a single element requirement in the CA-model. Furthermore, segments of the static objects of the environment are also placed in the grid.

![Fig.1 Spatial data structure](image-url)
As illustrated in Fig.1, each grid maintains a list of indices to speed up the access of the ship agents and static objects existing in the grid region, which differs from the representation of a finite number of states used in the CA-model.

This spatial data structure enhanced with element indices allows the swift retrieval of the grid from the element viewpoint, vice versa, the retrieval of the ship agents and static objects from the grid viewpoint.

(1) Computing ship list in grid

The simulation system built with the GS-model establishes a functional mapping from ship agents into distinct grids based on the initial positions of the ships. This process occurs in the beginning of the simulation process. The right part of Fig.2 shows the ship lists maintained in the three grids: Grid-i holds one ship agent; Grid-j holds no ship agent; Grid-k holds two ship agents. The list in each grid needs to be revised for every simulation step to reflect the new positions of the ships resulted from their continuous movement. The simulation proceeds with the following sequence of steps:

① Erase old contents of the ship list maintained in each grid.
② Locate a grid associated to the new position of each ship agent, and update the index value of the ship list maintained in that grid.

(2) Computing object list in grid

While no multiple ship agents can appear in more than one grid, one static object may occupy several grids. Consider the next algorithm to maintain the list of static objects:

① For every polygon representing the static object in the environment, construct a minimum bounding box and obtain the box points expressed by the spatial lattice (see Fig.2).
② For each grid touching the rectangular segment of the object, find the geometry relationship between the grid’s square and the object’s polygon. The object is indexed into the list if the relation of containment or intersection appears.

For instance in Fig.2, Object-1 covers grids with right diagonal; Object-2 covers grids with left diagonal. The Grid-i contains one object; the Grid-j contains two objects; the Grid-k has no object.

The cost of producing the object lists might be observed larger than the cost of computing the ship lists. However, the system would benefit from the fact that the object calculation is done only once at the beginning of simulation, instead of every simulation clock, thus achieving a significant performance gain over the SS-model.

![Fig.2 Schematics of GS-Model](image-url)
3.2 Sensing operation

Fig. 2 also illustrates the way the sensing around the area of ship agent is carried out in the GS-model. Given the current position of the ship and a radius of sensing, the system inspects the contiguous grids (filled with point-filling method in Fig. 2) in the spatial lattice around the ship, and builds up a merged list of ships and objects found at the search process. For example, at the bottom of Fig. 2, six ships and one object are found by the sensing operation of the given ship.

From the fact that the system let the ship agent search the grids only nearby its location, instead of having to examine the entire environment or interact with all other elements, it is highly likely that the lattice-based GS-model could achieve a good amount of cost reduction at the sensing operation of the local environment.

4. Experiment and discussion

To investigate the efficiency of the GS-model, experiments with a various numbers of ship agents and static objects and different sizes of grid structures are performed in the context of the ship group navigation simulation. All the experiments are done in a Dell OptiPlex GX620 PC with Intel Pentium 4 CPU 3.20GHz, DDR 1GB memory and Windows XP Pro SP2 OS. The analysis is made based on the metrics of total running time and maximum memory increment, as well as the derived measures of time and space complexities. The precision of detecting computation time is about 15 [nautical mile]. The sensing radius for all ship agents is set 5 [nautical mile]. All the data used to show the computation time and memory increment is obtained from the execution of 1000 steps of simulation loop.

For the process of interactive sensing between ship agents, a comparison of the SS-model and the GS-model is summarized in Fig. 3 and Fig. 4. The result indicates that the GS-model with an appropriate grid size outperforms the SS-model in time and space. The SS-model requires a pair-wise interaction between all ship agents in the system which results in a quadratic time complexity of $O(n^2)$. In contrast, the GS-model avoids such excessive communications by restricting search space only within the neighboring grids. For the interactive sensing of ship agents, the computation cost of the GS-model is divided into two parts: the ship-by-ship cost for searching nearby grids to obtain sensing result, and the cost for updating the ship lists of all grids in the spatial data structure. Let $k$ be the average number of grids within the sensing range of ship agents. The time complexity of the interactive sensing is $O(kn)$. For a fixed size of grid, $k$ becomes constant, degenerating the time complexity into $O(n)$. Both SS-model and GS-model can be implemented within a space complexity of $O(n)$. Therefore, GS-model with a suitable range of grid sizes outperforms SS-model.

![Fig. 3 Time complexities of interactive sensing between ship agents](image)

![Fig. 4 Space complexities of interactive sensing between ship agents](image)
For the process of sensing static objects, Fig.5 and Fig.6 summarize comparative performances of SS-model and GS-model. To investigate the influence of the number of static objects, experiments are done with a fixed number of ship agents (800).

With GS-model, the time complexity of sensing static objects is $O(kn)$, with fixed grid size and ship agent quantity, the time complexity becomes to $O(1)$. With SS-model, the time complexity is $O(mn)$, and becomes to $O(m)$ with fixed quantity of ship agents. The static object lists never require the refresh of their contents at each simulation step. Thus, the cost for sensing static object only includes searching nearby grid in the spatial data structure, which is independent of the number of static objects. The SS-model and GS-model with suitable grid size both can be implemented with a space complexity of $O(1)$ with fixed ship agent quantity. GS-model outperforms SS-model without the need of the complex computation of spatial relationship between ship agents and static objects.

The influence of grid size to the time and space complexity of interactive sensing between ship agents and sensing static objects is also analyzed. The results are gathered into Fig7 through Fig10. The analysis of grid size to the overall efficiency of GS-model indicates an optimal range of grid sizes. With assumption of the sensing radius of 5 [nautical mile], both for interactive sensing between ship agents and sensing static objects, the time and space complexities are lowest when the grid size is set between 0.5 and 5
and its overall effects were discussed based on the simulation experiments. The set of experiments confirmed that the GS-model with an appropriate range of grid sizes outperforms the SS-model in time and space, yielding far efficient processing of the sensing mechanism in ship agents.

The grid-based spatial data structure also would be used to a future research about auto route plan for autonomous ship agent, then a grid-based intelligent marine traffic simulation system with high computation performance.

References
Question and Answers:
Hiroki Iwasaki (Oshima National College of Maritime Technology):

Please teach me the ship agent’s generating interval.

Pan Mingyang:

Thank you for your question. The ship agent's generating interval is set up on the basis of the traffic data of the Uraga Suido Traffic Route.

Hiroki Iwasaki (Oshima National College of Maritime Technology):

I think, to describe the influence of object to the computation cost, the perimeter of objects would be more suitable than the amount of objects, how do you think?

Pan Mingyang:

In fact, for the GS-model with suitable grid size, there is no influence of the complexity of objects to the computation performance. For the SS-model, the perimeter and amount of total objects both can represent the complexity of objects to a certain extent, and the amount is selected because I think it is more straightforward and would not influence the result of analysis.