Adaptive strategic decision point extraction from Influence maps in games

Namo Podee [#1], Vishnu Kotrajaras [*2], Boonserm Kijsirikul [#3]

[#] *Department of Computer Engineering, Chulalongkorn University*
254 Phayathai Road, Pathumwan, Bangkok, Thailand 10300

[1] namo.p@student.chula.ac.th

[2] vishnu@cp.eng.chula.ac.th

[3] boonserm@cp.eng.chula.ac.th

***Abstract*—One of the main challenges facing artificial intelligence (AI) is the vast amount of data, which contains both helpful and unhelpful information from the perceived environment mixed together. It is often difficult for an agent to understand relationships amongst data, and this can lead to ineffective learning and decision-making. Many studies have proposed solutions to this problem by using machine learning. However, recent results in applying machine learning to AI are insufficient to encourage uptake outside academia. Its unpredictable behaviour combined with its indecomposable structure make it difficult to use in practical applications.**

Our research tackles how an agent learning process in games can be improved so that it is more robust and faster when retrieving information from its surrounding. The influence map, which is a representation of the agents' influences, will help the AI gain a greater understanding of the environment, without requiring detailed information of the map. Therefore the vast amount of data from the map is transformed into information that can be efficiently processed. Moreover, using the influence map as the input to a machine learning system leads to improvement in the efficiency of learning process and allows the system to generate the map that indicates the strategic value of each area. Our AI then uses such strategic values to decide its action. This allows higher-level analysis of game situations, leading to more accurate and faster decision-making. In addition, since programmers have more direct control over the AI's decision, the AI behaviour becomes more predictable.

I. Introduction

Artificial intelligence (AI) in game is becoming more important than before because of the increase in available computational time. Also, players expect AIs that are challenging, unpredictable and rational [1]. In order to accomplish these tasks we need AI that can compute large amount of data in short time and exhibit intelligent behavior to players. In complex environment games such as Real-time strategy games (RTS), the large amount of game data can be simplified using influence maps. Unlike existing approaches where influence maps were used to directly train AIs' behavior, our research explored the potential of using influence maps to generate environment-related strategic information. Such information was learned using neural networks. AIs could then be written to make use of such information.

Section II discussed related works. We explained our approach in detail in Section III. Experimental results were given in section IV. Section V concluded our paper.

II. Related Work

A. Influence Maps

An influence map [2] was a grid-map that represents influences of agents or objects on their surroundings. It was created based on a concept that each agent or object in the system influenced one another. By propagating the influence of each agent, an overall state of the system became visible. It simplified the complex system state data, allowing AIs to analyze data more effectively. The AIs were able to summarize the situation, memorize a previous situation, and even predict a possible future situation.

Two influence maps are shown in Fig. 1, each representing positions of a team of agents. By converting position data into the influence maps, a game state became much more understandable. AIs could also make use of such data.

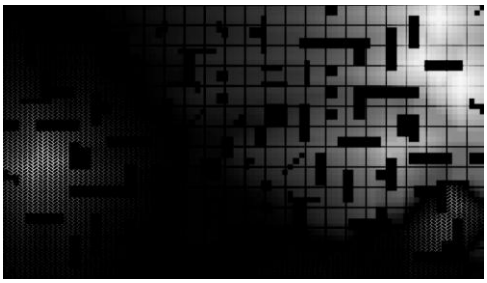

Fig. 1. Two influence maps, each representing a influence of a team of agents.

B. AI Creation Using Influence Maps

Su-Hyung [3] [4] had a turn-based game AI of the game 'Conqueror' learn its behavior by feeding a neural network with influence maps. Multi-layered influence map, which consisted of troops influence map and building influence map were computed (see Fig. 2). The maps were employed as input for a neural network. The neural network then decided agents' behavior using the given data. The methodology resulted in significant improvement of game AI.

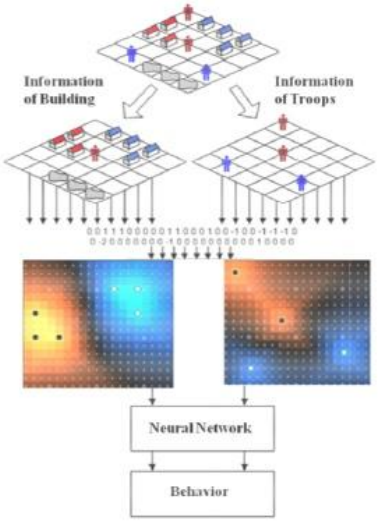

Fig. 2. Influence maps AI in 'Conqueror' [3].

III. METHODOLOGY

Our research used an already developed game called "Capture the flag" [5] (Fig. 3). It was an AI battle simulation system that runs two AI commanders. Each commander was able to issue commands to its soldiers.

We used the position data of our own soldiers combined with enemy soldiers' last known position information to generate an influence map. The game did not allow us to obtain enemy soldiers' data directly. Therefore the influence map was not quite perfect. But it was enough for our experiment.

When an important event, such as one of our soldiers made a kill or get killed, took place, the following data was collected:

- Influence map values of nine grids around and including the soldier's position.
- A rough or scaled Influence map values of nine grids (bigger grids) around and including the soldier's position.
- Distance between the soldier and its enemy base.
- The soldier's orientation in the map.

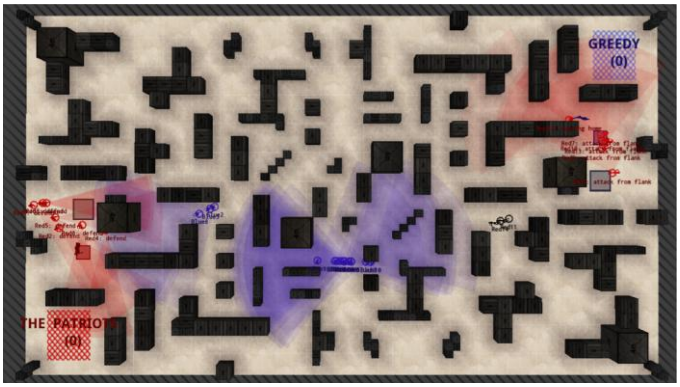

Fig. 3. Screenshot of 'Capture the flag'.

A neural network was trained offline with the collected data to obtain strategic values of each map position. The strategic values of all positions formed a suitability map. The whole process is as shown in Fig. 4. Information in the suitability map is illustrated in Fig. 5.

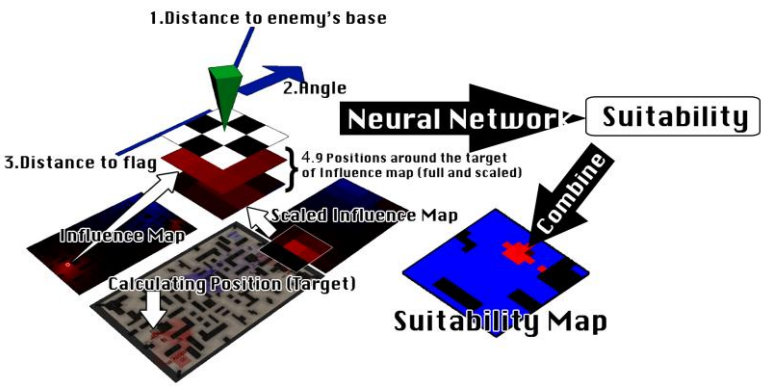

Fig. 4. Obtaining Suitability Map.

The suitability map provided information for our commander's AI. For each agent, the AI randomized 20 positions within an area of 15x15 grids surrounding the agent's flag. The position with the best strategic value was then chosen for the agent to move to. The agent's most effective orientation for the chosen position was also decided by the AI based on trained data. Each agent shot at

one of enemies in sight (agents had limited vision range).

Fig. 5. Example of a suitability map, dark colored areas have low suitability while lighter colored areas have high suitability.

IV. Experiment

We experimented and compared our methodology with a random positioning AI, which was used to collect training data for our methodology. The random positioning AI had exactly the same shooting behavior as our AI, but it selected positions (from the 15x15 grids surrounding the flag) and orientation for soldiers randomly.

The 5000 data sets were divided into 3500 training sets and 1500 validation sets. Agents' kill over death ratio were used as efficiency indicator.

The experiment was done by collecting data from 32 simulations of both random positioning AI and our AI. The average result of kill over death ratio of random positioning AI was 0.405, while it was 2.762 for our AI. It was quite obvious that our AI performed much better.

V. Conclusion

The efficiency of the obtained suitability map was tested by using the map to command our soldiers in the game. The result showed that our AI was much better than the basic AI that did not utilize the suitability map. Our method therefore showed strong potential for AI development. The method was also fast enough to be used in real-time.

We successfully utilized trained map data in game AI development. Our methodology can also be applied to similar data such as crime in cities, areas with social problems, or even product distribution in various regions. For our future work, we would like to improve the way decision-making system uses a suitability map. We also want to calculate a suitability map more accurately by combining it with other information.

Acknowledgment

We would like to thank Chulalongkorn University for providing facilities and resources for our research.

References

[1] Sweetser, P., Johnson, D., Sweetser, J., Wiles, J., 2003, Creating engaging artificial characters for games, *Proceeding ICEC '03 Proceedings of the second international conference on Entertainment computing*, p. 1-8.

[2] Hansson, N., 2010, Influence Maps I, *Game School Gems*, http://gameschoolgems.blogspot.com/2009/12/influence-maps-i.html (March 15, 2013)

[3] Su-Hyung, J., 2008, Evolving Neural NPCs with Layered Influence Map in the Real-time Simulation Game 'Conqueror', *CIG '08 Evolving Neural NPCs with Layered Influence Map in the Real-time Simulation Game 'Conqueror'*, p. 385-388.

[4] Su-Hyung, J., 2009, Optimal Strategy Selection of Non-Player character on Real Time Strategy Game using a Spectated Evolutionary Algorithm, *CIG '09 Evolving Neural NPCs with Layered Influence Map in the Real-time Simulation Game 'Conqueror'*, p. 75-79.

[5] Champandard, A.J., Champandard-Pail, P., Dunstan, P., Kogelnig, R., Brandstetter, M.F., 2012, Capture The Flag, *The AI Sandbox*, http://www.aisandbox.com (March 15, 2013)