Testing Reliability of Replay-based Imitation for StarCraft

In-Seok Oh and Kyung-Joong Kim*
Dept. of Computer Science and Engineering, Sejong University, Seoul, South Korea
ohinsuk@naver.com, kimkj@sejong.ac.kr

***Abstract*— For StarCraft, it's easy to download lots of replays from gaming portals. Using simple tools, it's possible to extract all the gaming events stored in the replays. At each frame, it can tell us the human player's decision making given game states. Instead of making hard-coded AIs, it's promising to imitate the human player's decision recorded in the replays. In this study, we propose to create an AI bot imitates human player's high-level decisions (attack or retreat) on a group of units from replays. As a first step, we tested the reliability of the imitation system using replays from portals. We reported the ratio of apparent mistakes from the imitation system and the way to reduce the error.**

I. Introduction

In RTS (Real Time Strategy) games, artificial intelligence (AI) has to quickly decide which commands to execute because small delay can make big difference on the final outcome. The constraint makes it very difficult to design AI bot for StarCraft. Also, the bots must perform lots of roles at the same time including unit control, building construction, unit production, scouting, combat and so on. Because of the challenges, it has attracted many researchers to investigate RTS games such as StarCraft. In this community, the development of BWAPI has played a key role in RTS AI research [1].

In our early study, we developed an imitation system to find the most similar frame from replay database and mimics the actions taken at the moment for unit controls of AI player [2]. It's based on the ideas of case-based reasoning and all the information from each frame is regarded as a case. In the case retrieval, all the low-level information from the frame was transformed into an influence map, numerical representation of the influence at each position. It's expected to find the best case based on the influence map similarity. When the system finds the most similar case, it matches the current units to the one on the case and mimics the unit-level commands and orientation to control units.

In this study, we attempted to imitate group-level decisions (attack or retreat) instead of unit-level actions. After the top-level imitation, the system will generate the individual unit's actions to achieve the high-level goals. Because the system is highly dependent on the replays from the portals, it's important to behavior reliably in the existence of unfiltered bad cases. The replays were played by different levels of players and also recorded mistakes or bad behaviors the imitation system should reject. However, it's not a trivial problem to filter them automatically. It requires lots of expertise to design the filtering system. In this work, we tested the reliability of the imitation system without any filtering. It's important to know the ratio of acceptable decisions from the imitation system and the way to reduce the errors.

II. Reliability of Imitation System

Our imitation system is totally based on the replays to control units. Currently, the system only uses the replays from the same map because it has not yet implemented the terrain analysis to calculate case similarities from different maps. It downloads replays from the portals and extracts all the actions/game states from them using software tools. When the game starts, the AI bot repeats the case retrieval and case reuse. In the case retrieval, it uses hashing techniques to speed up search and influence maps to compare spatial features. In combat situations, the player needs to make quick decision whether our units are better to attack opponents or retreat. Usually, the strength of the group of units has several factors: Positioning, Health of units, Configuration of Units, Terrain, Player's unit control skills and so on. It's not easy to formulate a simple equation to predict the outcome of the combat even for the professional players. In this work, we attempted to imitate the human player's decision (attack or retreat) on the similar situation.

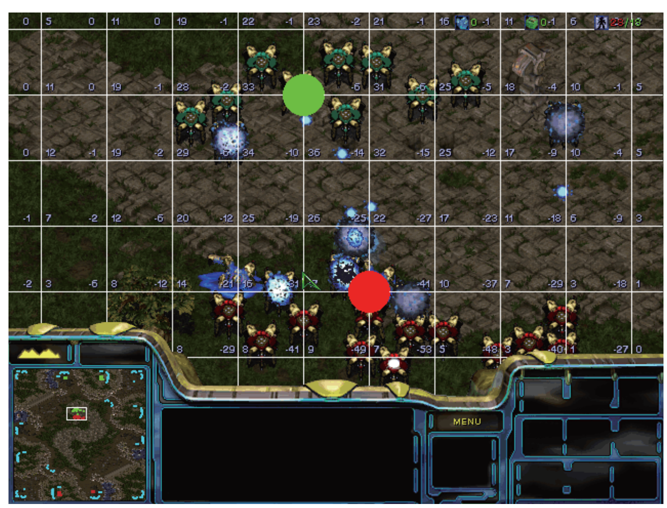

Figure 1. Influence map representation. For each cell, left-bottom number shows our force; right-bottom number is for enemy. The green and red circles point the location with the highest influence.

* Corresponding author

A. Imitation System

Influence map (IM) shows unit's power in the current map and has been used to analyze the spatial influence of units. It also allows to enhance the speed of comparison by avoiding low-level unit-by-unit similarity check. In IM, the influence was calculated at each location based on the near units' health power. If the location is populated with several powerful units from our team, it's good place to stay. Figure 1 is an example of two players' influence values and the peak points. It's interesting that the strongest place is not the center of the units.

Our hashing technique uses four keys: types of unit (Zealot & Dragoon) and average of x and y position. At first, the key values from the current game are inputted to the case retrieval module. For each query, it returns different number of cases and adjusts the number by changing tolerance of matching. The N cases retrieved are transformed into N influence maps. The last step is to find the closest influence map from the N candidates. It's possible to select just one case to imitate or select M cases from the N cases ($M \leq N$). In case of the M cases, the final imitation selects the decision by majority. It's interesting to see whether the imitation based on multiple cases can enhance the reliability.

B. Reliability Test

In the imitation system, there are several reasons to make it unreliable. They're 1) quality/quantity of replays 2) mistakes or undesirable actions by human players 3) hidden/undetectable factors in human decision making and 4) imperfect case retrieval. In this study, we measure the reliability based on the apparent mistakes by the imitation system. Based on expert player's opinion, if the gap between two IMs is more than 500, the decision is easily one of "attack" or "retreat." If one player has +500 than opponent, the decision is clearly "attack." If the gap is smaller than 500, there are several factors to be considered in decision making. We count the correct/incorrect decisions by the imitation system when the gap is big. For example, if the player A's units power is 500 points more than player B's units, it's definitely "attack" for player A. In the situation, it's rarely possible to win even if the player has professional micromanagement skills. If our imitation system concludes "attack" for the player A at the situation, it's correct decision. If not, it's counted as "incorrect" one.

III. Experimental Results

We collected 216 replays from one of StarCraft maps named as "Python". From the replays, we extracted about 200,000 cases and loaded them into main memory for real-time imitation. In the experiments, we controls the number of cases (M) to be considered for imitation. For each value of M, we played 500 games. During the testing, two players are designed to attack the opponent's highly influenced location. In the game play, the imitation system also makes decisions for each player given the situation. The reliability was measured based on the correctness of decisions. For each game, the number of Dragoon and Zealot units is randomly chosen.

- Map : Python
- Match : Protoss vs Protoss
- Dragoon Units : 3 ~ 21
- Zealot Units : 2 ~ 14
- # of Games Executed: 500

Table 1 shows the reliability of our imitation system. The results show that the imitation system can behave correctly about 80% on the apparent situations. Although we don't do any preprocessing (filtering cases or replays), it's highly accurate decision making. The reliability is maximum if the number of cases participate in the final decision making is 100. In the setting, the system always uses 100 close cases to derive the final decision.

Table 1. Summary of imitation system's reliability with different settings

# of Cases for Decision (M)	Correct Decision	Incorrect Decision	Reliability
1	328,182	69,553	82.4%
5	320,062	74,966	81.0%
10	315,813	73,384	81.1%
20	313,418	80,515	79.6%
50	293,393	64,075	82.1%
100	232,671	44,638	83.9%

IV. Conclusions and Future Work

Although the replays are important resource to design imitation-based AI for the RTS games, there are still several challenges to hinder the progress. In this study, we attempted to evaluate the reliability of imitation bot's decision making (attack or retreat) on apparent situations. The results show that the imitation system behaves about 80% correctly. Because it uses replays as it is without any treatment, it's the baseline performance of the system. It's possible to increase the reliability up to 84% using decision making by majority. The next step is to find the reason of misbehavior and develop techniques to reduce the errors. For example, the system can delete the case retrieved during the game playing if its imitation is not successful. Because the quality of case is dependent on the current gaming situation, it's desirable to discard them during the play.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (2013 R1A2A2A01016589).

References

[1] BW API : An API for Interacting with StarCraft: Brood War, http://code.google.com/p/bwapi/

[2] I.-S. Oh, H.-C. Cho and K.-J. Kim, "Imitation learning for combat system in RTS games with application to starcraft." *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*. IEEE, 2014.