Potential Flows for Controlling Scout Units in StarCraft

Kien Quang Nguyen*, Zhe Wang*, and Ruck Thawonmas*
*Intelligent Computer Entertainment Laboratory, Graduate School of Information Science and Engineering, Ritsumeikan University

***Abstract*—Real-Time Strategy (RTS) games typically take place in a war-like setting and are accompanied with complicated game play. They are not only difficult for human players to master, but also provide a challenging platform for AI research. In a typical RTS game, such as StarCraft, WarCraft, or Age of Empires, knowing what the opponent is doing is a great advantage and sometimes an important key to win the game. For that, good scouting is required. As subsequent work for improving the scouting agent in our StarCraft AI bot–IceBot–the winner of the mixed division in Student StarCraft AI Tournament 2012, this paper proposes a method that applies potential flows to controlling scout units in StarCraft. The proposed method outperforms an existing scouting method as well as a modified version of this existing method and is comparable to scouting by human players.**

I. Introduction

Potential fields are often used for finding a path while avoiding obstacles. One of its first applications is robot navigation. In this application, the goal is given an attractive potential while all obstacles are given repulsive potentials [1]. The gradient which is created by the combination of these two kinds of fields leads the robot from the start location to the goal. A problem with this configuration is that the robot can become attracted to a local minimum of the potential field and will thus get stuck in that location, failing to reach the goal. A solution to this problem is to utilize potential fields that satisfy the Laplace equation whose solutions are called harmonic functions, which do not have local minima [2]–[4]; such potential fields are called potential flows.

With a very complicated game-play, the RTS game StarCraft is not only a challenging game for human players but also a challenging platform for AI research. This paper discusses an application of potential flows to scouting in StarCraft, a crucial task in the game. Potential flows are used to control scout units for collecting information from the opponents while avoiding opponents' attacks and obstacles. In addition to basic potential flows, such as the vortex flow, uniform flow and doublet flow, we propose a new flow called needle flow. Each or a combination of these flows is associated with a different object or opponent unit in the game. This research aims at improving the scouting agents in our RTS project, a StarCraft AI bot–Icebot–which has won the first place of the mixed division in Student StarCraft AI Tournament 2012 (SSCAI 2012)[1].

The contributions of this paper are as follows:

1) To our knowledge, the first application of potential flows to the scouting task in RTS games, such as StarCraft,
2) A new potential flow, needle flow, suitable for being associated with each of the opponent's attacking units, and
3) A method that associates each flow with the corresponding object or opponent unit and combines them for effectively controlling a scout unit.

In the next section we give the outline of scouting in StarCraft and the mathematical background of potential flows and describe related work. Section III presents the methodology for controlling a scout unit with potential flows, while Section IV describes our experiments and the results. Conclusions are then drawn and future work is stated.

II. Background

A. Scouting in StarCraft

StarCraft is one the most popular RTS games in the recent decade, developed by Blizzard Entertainment in 1998. Its complicated game play not only is difficult for human players to master, but also provides many challenging problems for AI research. The game starts with a main base and a few workers. A player must send them to gather resources and produce more workers, then build more expansions and train army units to destroy their opponents. This process requires not only a lot of skills, but also plenty of expert knowledge.

Scouting is an essential activity in the StarCraft game-play. This is due to the existence of the fog-of-war, which covers the majority of the game map. For each player, only a certain range is visible where their units (army units or buildings) stay. Therefore, without sending reconnaissance units to opponents' territories for gathering information, such players will be blind and thus difficult to come up with a tactical plan to conquer/counteract their opponents.

Similar to many other RTS games, a way to overcome the uncertainty in such imperfect environments is to keep sending certain units to gather information about the opponents, which is also considered as one of the most significant factors in winning the game. However, on one hand, a scout unit is usually weak and can be easily killed by its opponent's attacking units. On the other hand, staying too far from its opponent's territory will result in a scouting failure, not able to acquire the opponent's important information. A mechanism is therefore necessary that navigates a scout unit properly and keeps it alive until its mission is completed.

[1] http://sscaitournament.com/index.php?action=2012

978-1-4673-5311-3/13/$31.00 ©2013 IEEE

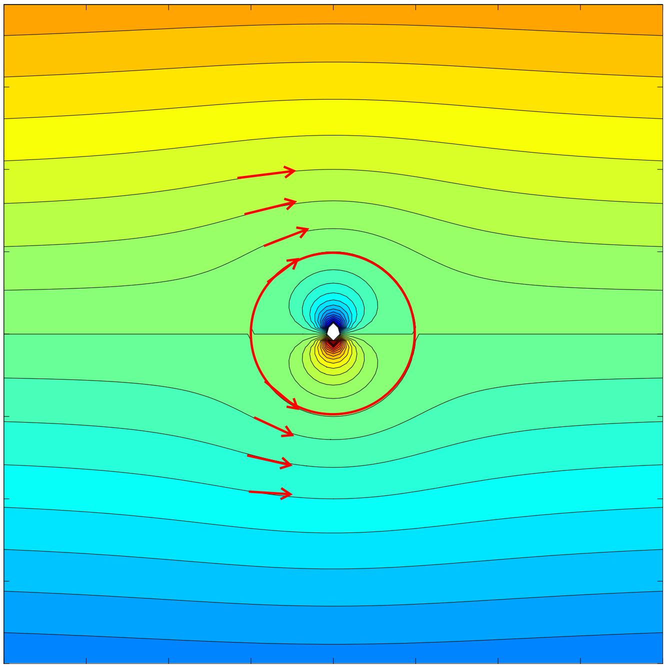


Fig. 1. A potential flow, whose representative streamlines are arrowed, caused by an obstacle inside a uniform flow.

B. Potential Flow

A potential flow can be derived by solving the Laplace equation on (x, y) plane below.

$$\bigtriangledown^2 \varphi = \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = 0$$

Solutions to this equation are known as harmonic functions. In addition, φ is also known as the velocity potential because the its gradient is the velocity vector field (u, v), which defines the streamlines, in other words, the trajectories for a particle (or a scout unit in our case) to follow.

$$\begin{aligned} u &= \frac{\partial \varphi}{\partial x} = \frac{\partial \theta}{\partial y} \\ v &= \frac{\partial \varphi}{\partial y} = -\frac{\partial \theta}{\partial x} \end{aligned} \tag{1}$$

where θ is another harmonic function conjugate to φ and is known as the stream function.

The velocity potential and the stream function are together represented as a complex potential, known as potential flow,

$$f = \varphi + i\theta$$

along with the complex position $z = x+iy$. As Eqn. (1) shows, in order to derive the speed vector (u, v) for a scout unit of interest, either the real part or the imaginary part of f can be used.

Care must be taken for obstacle avoidance. In the goal/obstacle configuration of potential fields, an obstacle is often represented as a repulsive field, which does not ensure that the controlled unit does not penetrate the obstacle; this is another problem with this configuration besides having local optima, as mentioned earlier. With respect to potential flows, to represent a single circular obstacle in a generic potential flow, the circle theorem is employed. This theorem gives a new stream function for a flow $f(z)$ when a cylinder is placed into that flow. The result is a potential flow bellow (Fig. 1).

$$F = f(z - z_s) + \bar{f}(\frac{a^2}{\bar{z}} - z_s)$$

where a, z_s, $\bar{f}$ are the radius of the obstacle, the source of the flow $f(z)$, and the conjugate function of complex potential f, respectively. The conjugate $\bar{f}(\frac{a^2}{\bar{z}} - z_s)$ can be considered as a representation of an obstacle of interest in the potential flow.

Note that the stream function is identically zero on the boundary of the obstacle, indicating that the boundary is on a stream line.

$$\begin{aligned} F|_{\|z\|=a} &= f(z - z_s) + \bar{f}(z - z_s) = 2\varphi \\ &\Rightarrow \theta = 0 \end{aligned}$$

This also implies that the velocity field actually flows around the obstacle without penetrating it. As seen in Fig. 1, a feature known as a doublet is formed inside the obstacle.

C. Related work

Recent AI research on StarCraft focuses mainly on micro-management and opponent modeling, including reinforcement learning for small-scale combating [5], a Monte-Carlo planning method for micromanagement [6], and heuristic search [7] for micro controlling units in combat scenarios. Ben Weber et al. also proposed a method using a particle model to estimate opponent units' positions [8], while Bayesian model was applied to both unit micromanagement and opponent planning prediction [9],

Park et al. applied various machine learning algorithms to prediction of the opponent's plans based on the scouted information [10]. In their work, an algorithm for navigating a scout unit was also proposed using a number of heuristic rules. With respect to Potential fields, not potential flows, they have been applied to navigation of units during a combat [11].

In this paper, we compare our method with the scouting method proposed by Park et al. It would be debatable that we should compare our method to the potential-field approach in [11]. However, because that paper mainly focuses on combating not scouting, modifying the method therein for scouting and then comparing it with the proposed method here is beyond the scope of this paper.

III. Proposed Method

In StarCraft, because we mainly control each scout unit in a fixed area (such as its opponent's base), we need some mechanisms for navigating a scout unit around its assigned area in order to obtain as much information as possible. With the approach based on potential fields using only repulsive and attractive fields, it is hard to design such fields to make a scout unit move around a targeted area and not stumble on a local minimum. On the contrary, the potential-flow approach uses harmonic functions which do not contain local minima. In addition, the vortex flow, a basic potential flow, can be used as the main force to navigate the scout unit around the restricted area naturally. For these reasons, we propose a method for

Algorithm 1 $scouting(unit)$

▷ This algorithm controls a scout unit using potential flows. Here $unit$ is the scout unit of interest.

$f \leftarrow regionFlow(unit)$
▷ Get the flow generated by the opponent's region.
$f \leftarrow f + borderFlow(unit)$
▷ Get the flow generated by the opponent's region border.
$f \leftarrow f + obstaclesFlow(unit)$
▷ Get the flow generated by obstacles.
$f \leftarrow f + attackFlow(unit)$
▷ Get the flow generated by the opponent's attacking units.
$v \leftarrow getVelocity(f)$
$moveScoutUnit(v)$
▷ Move the scout unit according to the velocity obtained.

controlling a scout unit using the combination of several kinds of potential flows (Fig. 2) below:

- Vortex flow: $V(z) = iUlog(z - z_{start})$
- Source/Sink flow: $S(z) = Ulog(z - z_{start})$
- Doublet flow: $D(z) = \dfrac{a^2}{z - z_{start}}$
- Needle flow:

$$N(z) = Um(z - z_{start})log(z - z_{start})$$

where

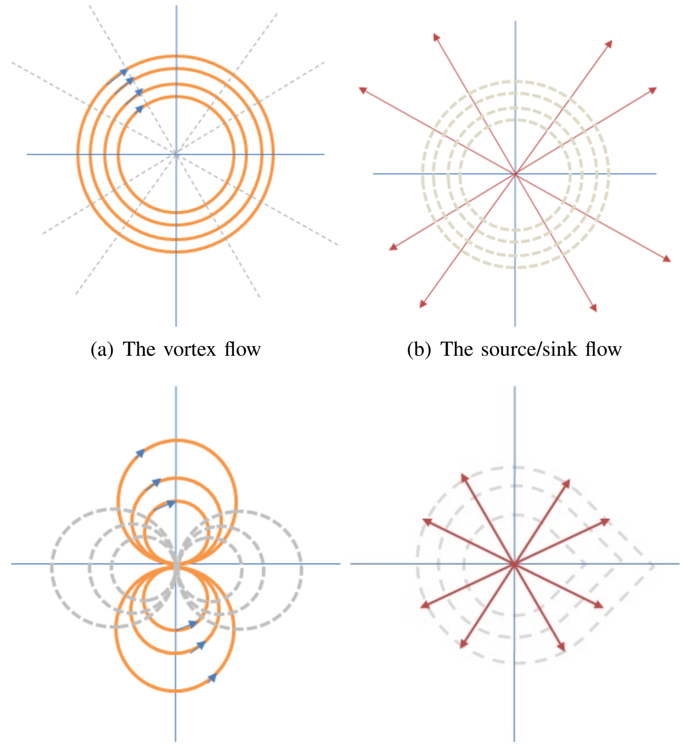
$$m(z) = \begin{cases} \dfrac{1}{cos(\beta - \theta)} & \text{if } \theta \in [0; \beta] \\ \dfrac{1}{cos(\beta + \theta)} & \text{if } \theta \in [-\beta; 0] \\ 1 & \text{otherwise} \end{cases}$$

The first three above are basic potential flows while the last one is originally developed by the authors for representing the potential flow generated by each attacking unit of an opponent. In addition, $z = re^{i\theta}$; U, z_{start}, and β are the magnifier parameter, the source position of the flow, and the open angle threshold (used to create a needle flow) respectively. From the combination of flow functions and Eqn. (1), we can derive the velocity (u, v) which in turn will be used to control a scout unit (Alg. (1)).

In the next sub-sections we discuss how we apply potential flows to individual objects in the terrain of StarCraft. For more convenience, from now on we denote that $z' = z - z_{start}$ and $p_{1,2...,5}$ are weight parameters.

A. *Potential flow generated by the opponent's region*

For scouting in StarCraft, the more a scout unit moves around its opponent's region the easier it gathers more information. A natural way to keep a scout unit going around the region is to assign a vortex flow starting from the region center. However, using only the vortex flow will result in the scout unit only circulating around the region with a fixed distance from the center. To avoid this and encourage the scout unit to explore also the inner part of the region, where most of its opponent's buildings are lying, we also add a source/sink flow (source for



(a) The vortex flow (b) The source/sink flow (c) The doublet flow (d) The needle flow

Fig. 2. Potential flows, used in this work, represented by the stream lines (solid lines) and velocity potentials (dot lines).

pushing while sink for pulling) to the region center. The result is the potential flow generated by the following function

$$R(z) = \begin{cases} p_1V(z) + p_2S(z) & \text{if } \|z'\| > d_{r\_thres} \\ p_1V(z) - p_2S(z) & \text{otherwise} \end{cases} \quad (2)$$

Here d_{r_thres} is the distance at which the source/sink flow will be switched (Fig. 3). With this, the scout unit will be drawn to the region center to a certain distance then be pushed back if it comes too close to the region center.

B. *Potential flow generated by the opponent's region border*

The further the scout unit is from the opponent's region center, the smaller the center's source/sink flow affects it. Also, while navigating the scout unit around the region, we need to prevent the scout unit from colliding with the border. To solve these two problems, we assign a vortex flow and a source flow to each game node forming the border line (border node). Thereby, in the case where the scout unit stays too far from the region center, the two forces from the border will keep it moving while not colliding with the border. We can calculate the border's flow as

$$B(z) = \begin{cases} p_3V(z) + p_4S(z) & \text{if } \|z'\| < d_{b\_thres} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Here d_{b_thres} is the distance at which we activate the border's flow (Fig. 4). With this, if the scout unit does not come to a certain distance from a border node of interest, this flow will not be activated. This is done to prevent the border's flow from interfering the region flow. In addition, the source flow here prevents the scout unit from coming too close to the border

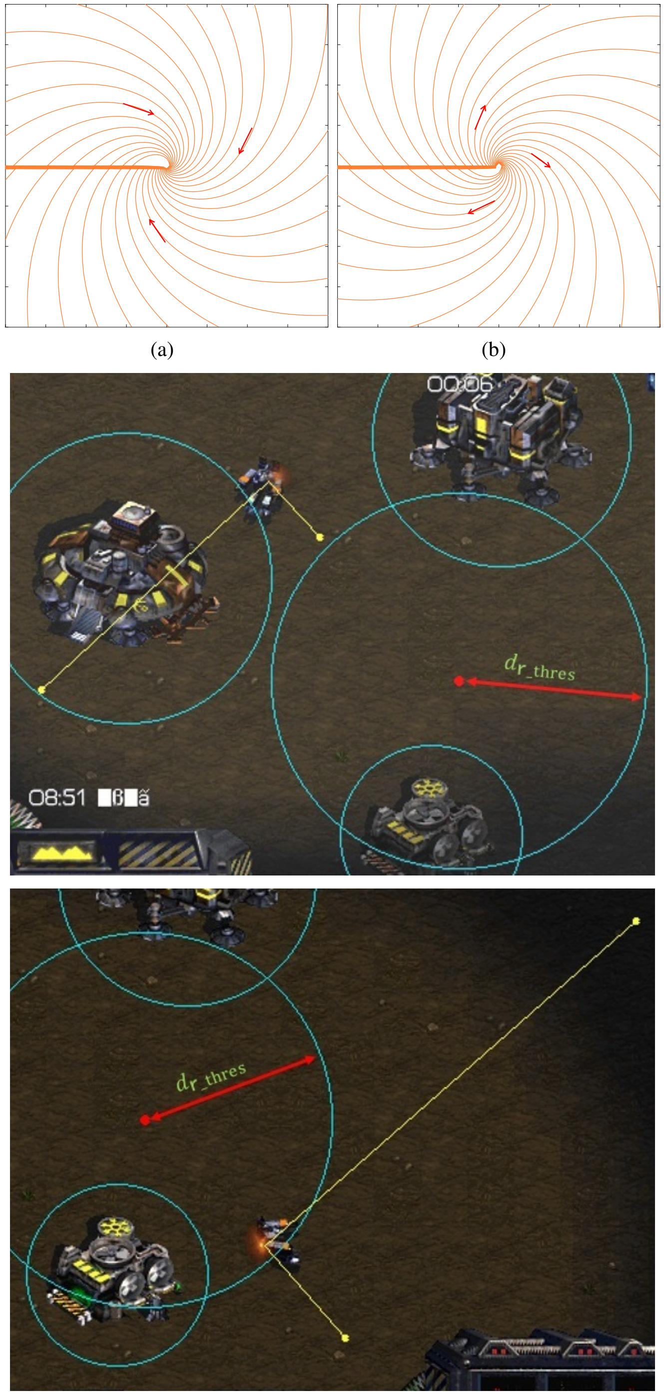


Fig. 3. Potential flows generated by the region center in cases where the scout unit is pulled toward the region center 3(a) and is pushed away from the region center 3(b). Examples of the two forces affecting the scout unit due to this flow type are shown in yellow in the upper map (the former case) and the lower map (the latter case).

because scouting positions nearer the border have less room to escape under attack. If there are multiple activated flows, all of them will be accumulated.

C. Potential flow generated by obstacles

We consider all unlifted buildings, non-attacking workers, resources, and other non-movable objects in the scout unit's view range as obstacles (Fig 5). Here we model each obstacle as a circular object. Thereby, its associated potential flow can be calculated by using the aforementioned circle theorem, which ensures that the flow does not penetrate the corresponding obstacle. By ignoring the border flow which rarely affects the obstacles, for most cases, we can safely assume that all flows flowing around an obstacle consist only of the region's vortex flow and source/sink flow. Given the region center z_c, the formulas for the obstacle in each of these flows can be represented as

$$\begin{aligned} O_v(z) &= -iUlog(\frac{a^2}{z'} - \bar{z'_c}) \\ O_s(z) &= Ulog(\frac{a^2}{z'} - \bar{z'_c}) \\ z'_c &= z_c - z_{start} \end{aligned}$$

Here $O_v(z)$ and $O_s(z)$ are the obstacle's doublet flows for the vortex flow and the source/sink flow, respectively. By using the same recipe as the region flow (Eqn. (2)), the obstacle flow becomes

$$O(z) = \begin{cases} p_1 O_v(z) + p_2 O_s(z) & \text{if } \|z'_c\| > d_{r\_thres} \\ p_1 O_v(z) - p_2 O_s(z) & \text{otherwise} \end{cases} \quad (4)$$

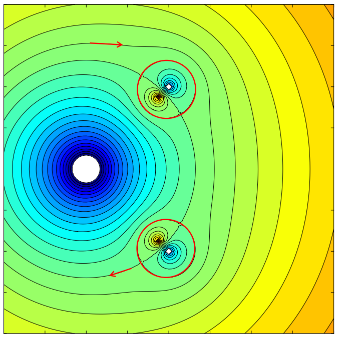
If there are multiple obstacles in sight of the scout unit, their flows will be accumulated.

Fig. 4. An example of forces (orange vectors) affecting the scout unit and their sources (red points) due to potential flows by the corresponding border nodes.

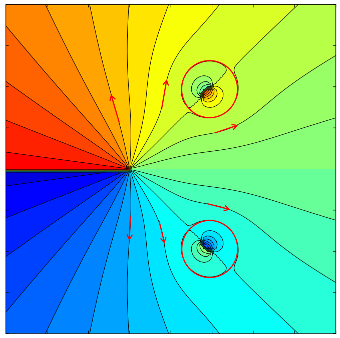
D. Potential flow generated by the opponent's attacking units

In the targeted scouting region, the opponent often reserves some attacking units to defend its base. A typical method to assign the potential flow to each of such units is that of using the source flow or repulsive force field. However, because an attacking unit of the opponent can only attack our scout unit if it faces our unit inside its attack range, we do not need to run away from those attacking units if they are not aware of our scout unit. To incorporate this thought into the potential flow, we use the needle flow, which emphasizes a certain direction, representing the current direction of the attacking unit of interest.

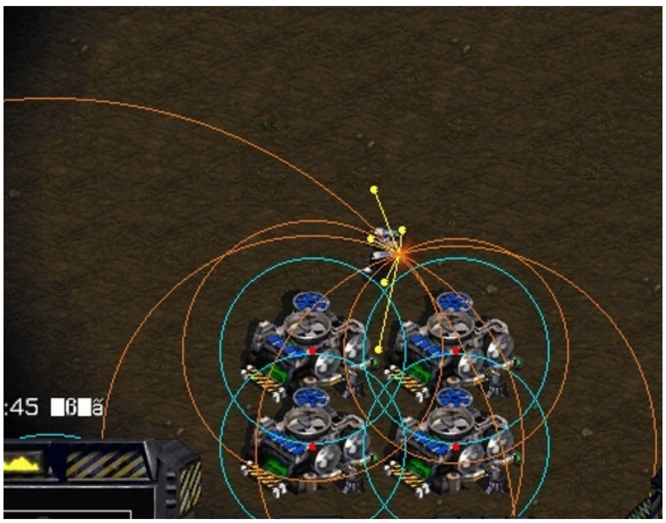
$$E(z) = p_5 e^{-i\alpha} N(z) \quad (5)$$



(a) Obstacles in the vortex flow



(b) Obstacles in the source flow



(c)

Fig. 5. Potential flows generated by obstacles in the region flow (5(a) and 5(b)). Examples of forces (yellow vectors) affecting the scout unit and their sources (red points) due to this flow type are shown in Fig. 5(c).

Here α is the direction of the opponent's attacking unit. The open angle of the needle flow can be calculated by using its attack range r_{e_atk} in the following function

$$\beta = \arccos(\frac{1}{r_{e\_atk}})$$

The emphasized force in the opponent's facing direction encourages our scout unit to stay away from the opponent's attacking direction while still being able to maintain a proper distance to go behind the opponent's unit (Fig. 6). Note that in this paper, we only consider the potential flow generated by an attacking unit in sight of our scout unit. If there are multiple such attacking units, their flows will be accumulated.

IV. Evaluation

We evaluated the performance of the proposed method with three different scenarios or maps (Fig. 7). Because in StarCraft each scout unit is often dispatched to only one region of the opponent, we designed the test-case scenarios with only one region for scouting that resembles the opponent's start region on popular competition maps. The opponent team is controlled by StarCraft built-in AI (specifically, the Campaign Expansion Difficult AI script) in all scenarios, whereas the race in each scenario (Terran, Zerg, and Protoss) is different. In each

Fig. 6. Examples of forces (yellow vectors) affecting the scout unit and the opponent's attack range (orange circles) due to potential flows by the opponent's attacking units.

scenario, we put a scout unit (Terran SCV), the opponent's army units, workers and buildings in a limited space with a start resource for the opponent. Effective scouting should meet the following three criteria:

- Never stop moving – the scout unit should not just hide in a safe corner.
- Avoid being attacked while scouting – the scout unit must stay alive as long as possible to collect information.
- Explore places hiding in fog-of-war as many as possible – this should be done when the scout unit's safety is secured.

As time goes by, the built-in AI is also able to produce more army units and construct more buildings. We recorded the survival time of the scout unit and the number of discovered buildings. The result is then compared with the scout unit using the navigation method proposed by Park et al [10] and the scout unit controlled by human players. Note that because we placed some of the opponent's attacking units on the maps beforehand, it is harder for a scout unit to survive than in normal games where the opponent has to manually build required buildings before it can produce attacking units.

The parameters of each flow are for creation of a fine total flow. These parameters do not depend on a map, except the magnifier parameter U. By using BWAPI[2] and one of it add-ons BWTA[3] to implement our bot and assuming that the shape of a region is nearly a circle, the magnifier parameter, for regulating the circulating speed, is calculated from the region perimeter p_r, which can easily be obtained, as

$$U = 0.4\lfloor\frac{p_r}{2\pi}\rfloor$$

In the experiments, the rest of the parameters are set as follow

$$\{p_1, p_2, p_3, p_4, p_5\} = \{1, 0.25, -1, 0.75, 0.75\}$$

[2]https://code.google.com/p/bwapi/

[3]https://code.google.com/p/bwta/

and

$$\{d_{r\_thres}, d_{b\_thres}\} = \{128, 128\}$$

Note that the lengths of these two thresholds are heuristically set to about four times the diameter of the scout unit.

We ran 100 games for each of the three maps with the opponent's buildings and army units being placed as shown in Fig. 7. In Park et al. [10], the navigation method only considers the opponent's building as primary objects ignoring the opponent's army units. This results in a very short live time when the opponent's army unit (which can attack our scout unit) comes out. We thus modified Park's original method to allow the scout unit to consider also army units in the same fashion as buildings.

A. *Survival time of the scout unit*

Figure 8 shows the survival time of the scout unit of each method, where Go Circle (Bldg), Go Circle (All), and Human, indicate Park's original method [10], Park's extended method, and the performance of five human players, who are members in our RTS project with an average RTS playing skill, respectively. It can be clearly seen that the proposed method (Potential Flow) outperforms both Go Circle (Bldg) and Go Circle (All). In addition, the proposed method's survival time is comparable to that of human players. It is superior to human players in the case of against Terran, because of all of Terran's units in use were ranged units, combined with the appearance of fast units like Vulture(s), it is quite hard for a human player to maintain a good scouting; in this case, however, our method, which can calculate the exact range of enemies, has a clear advantage.

Although the standard error of the proposed method is still quite high, we have found that the main reason is that in some cases, the opponent can produce stronger units (Vulture, Hydralisk, Dragoon...) in a short time. These units can easily kill our SCV which has a shorter sight range and/or moves slower. However, to achieve this, the opponent has to slow down its economy development.

B. *Number of the discovered buildings*

One of the main purposes of scouting is to reveal the opponent's buildings, both finished or under construction. The information on the opponent's buildings can give human players a rough image of what the opponent's strategy is or whether or not the opponent is planning to rush. This is a crucial part of the game play. In this experiment, we counted how many opponent's buildings each scout unit can discover before being destroyed. We placed eight, five, and nine buildings in the map against Terran, Zerg, and Protoss, respectively.

The average number and maximum number of discovered buildings of each scouting method are shown in Fig 9. From Fig 9(b), we can see that the scout unit using Park's original method in some cases, those games its scout unit achieved the maximum number of discovered buildings, can discover all of the start buildings, but because it can only survive for a limited time, it cannot discover any of newly constructed building.

Being able to survive longer, our scout unit can not only discover most of the time the opponent's start buildings, but also continue gathering information about newly constructed buildings. This property is useful for predicting the opponent's strategy. Our scout unit only gives up when the opponent can produce stronger units such as Dragoon or Vulture, in which it is also hard for a human player to continue scouting using just a worker unit as done in this work.

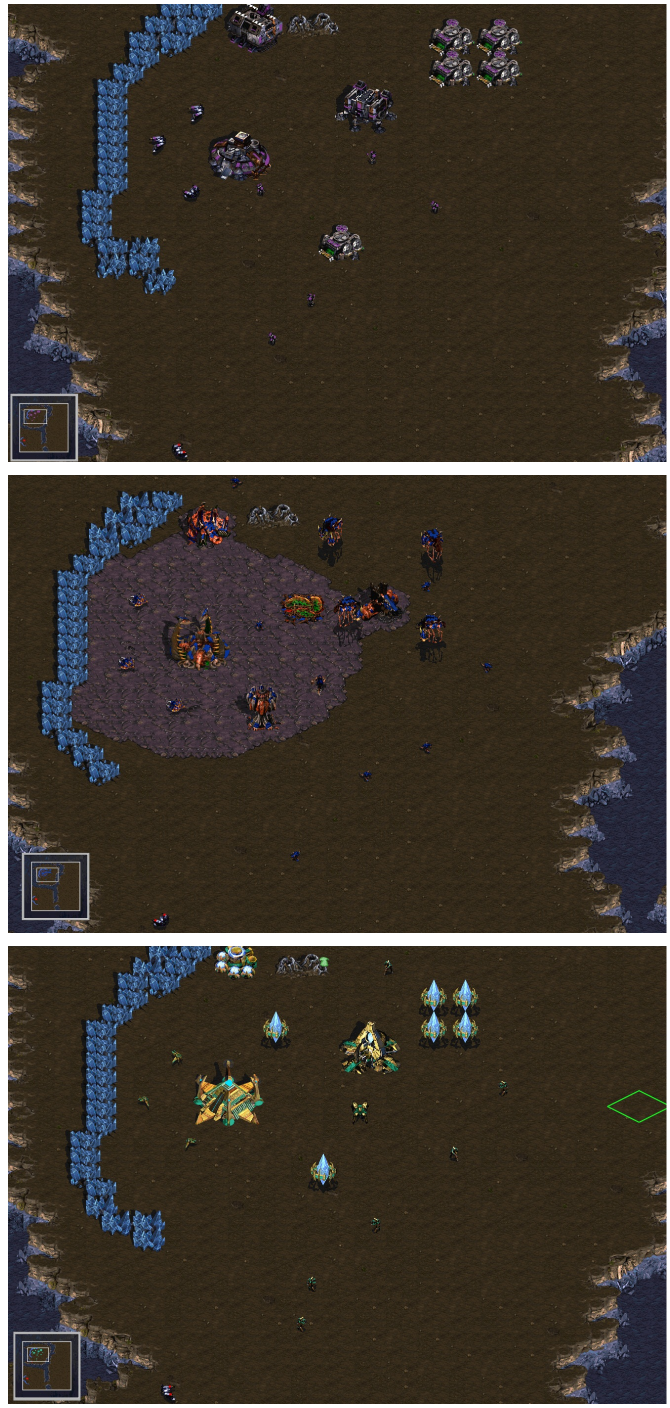


Fig. 7. Scenarios used to evaluate the performance of the scout unit, where all of the opponent's starting army units, workers and buildings are shown. On the bottom-left of each scenario there is a thumbnail of the whole map and a SCV (Space Construction Vehicle) used as our scout unit. The top, middle, and bottom figures show the scenarios in case the enemy race is Terran, Zerg, and Protoss, respectively

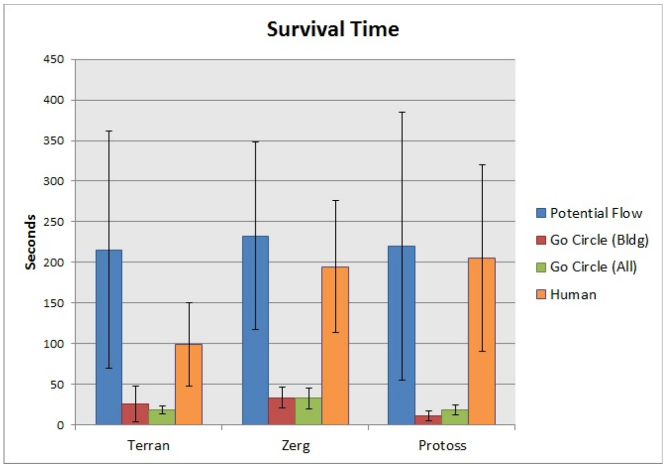


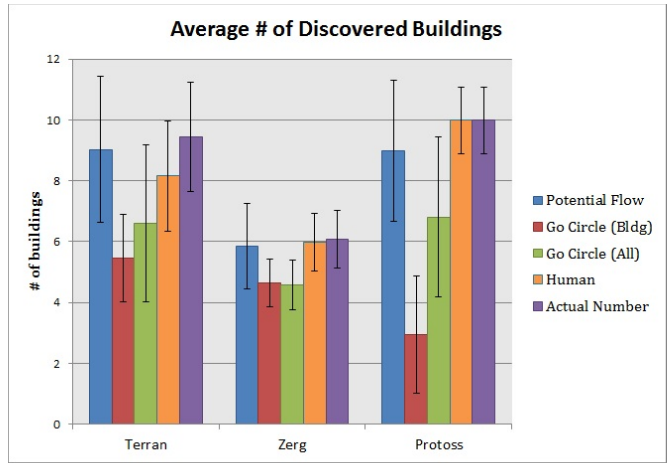
Fig. 8. The survival time of each method when scouting against difference races.

V. Conclusions and Future Work

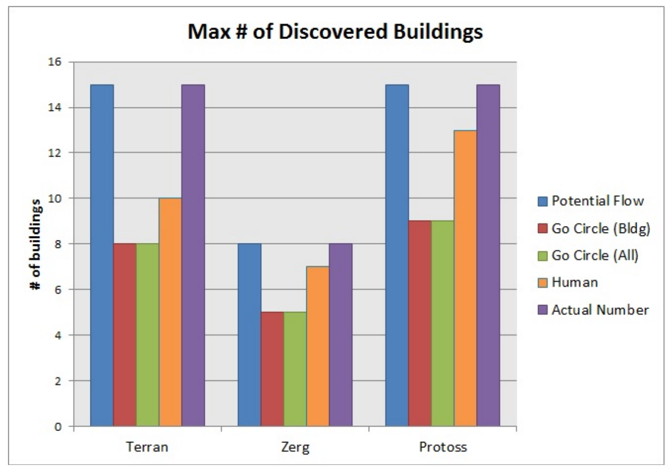
In StarCraft, a player with a capable scouting skill can completely dominate their opponents. With the information from scouting, such a player can easily guess their opponents' strategies and prepare effective counter strategies. As a result, scouting is an essential part in StarCraft's game play.

This paper proposed a method that applies potential flows to the scouting task in StarCraft and newly introduced a potential flow which is suited for representing the opponent's attacking unit. We examined three typical scenarios whether or not our scouting method is capable of battling against different races in StarCraft. The results show that our method outperformed an existing scouting method [10] as well as its modified one and that the proposed method was comparable to scouting by human players who are of average level. Because of the use of potential flows with no local minima, we can expect that the proposed scouting method would perform better than scouting methods based on potential fields, where local minima exist.

Our future work includes incorporation of effective and efficient search algorithms into scouting. This is to improve the scouting performance in a situation where the scout unit is initially located far from its targeted opponent's region. We also plan to extend this work to scouting of multiple opponents' regions and will work on how to apply the scouting results to predicting and countering opponent AIs.



(a)



(b)

Fig. 9. The average and max number of the discovered buildings, until the scout unit is killed, in each scenario.

References

[1] Oussama Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *Int. J. Robotics Research*, vol. 5, No. 1, pp.90-98, March 1986

[2] Sadao Akishita, Sadao Kawamura and Kei-ichi Hayashi, "New navigation function utilizing hydrodynamic potential for mobile robot," *Intelligent Motion Control, 1990. Proceedings of the IEEE International Workshop*, vol. 2, pp.413-417, 1990

[3] Stephen Waydo and Richard M. Murray, "Vehicle motion planning using stream functions," *2012 IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA'03.*, Vol.2, pp.2484-2491, 2003

[4] R. Daily and D.M. Bevly, "Harmonic potential field path planning for high speed vehicles," *American Control Conference, 2008*, pp.4609-4614, 11-13 June 2008

[5] Stefan Wender and Ian Watson, "Applying Reinforcement Learning to Small Scale Combat in the Real-Time Strategy Game StarCraft:Broodwar," *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, pp.402-408, 11-14 Sept. 2012

[6] Wang Zhe, Kien Quang Nguyen, Ruck Thawonmas, and Frank Rinaldo, "Using Monte-Carlo Planning for Micro-Management in Starcraft,," *Proc. of the 4th Annual Asian GAME-ON Conference on Simulation and AI in Computer Games (GAMEON ASIA 2012)*, Kyoto, Japan, pp. 33-35, Feb. 24-25, 2012.

[7] David Churchill, Abdallah Saffidine, and Michael Buro, "Fast heuristic search for RTS game combat scenarios," *2012 Proceedings of AIIDE,(pre-print available at www. cs. ualberta. ca/mburo/ps/aiide12-combat.pdf)*, 2012

[8] Ben G. Weber, Michael Mateas, and Arnav Jhala, "A particle model for state estimation in real-time strategy games," *2011 Proceedings of AIIDE*, pp.103-108, 2011

[9] G. Synnaeve and P. Bessiere, "Bayesian model for RTS units control applied to StarCraft," *2011 IEEE Conference on Computational Intelligence and Games (CIG)*, pp.190-196, Aug. 31 2011-Sept. 3 2011

[10] Hyunsoo Park, KwangYeol Lee, Ho-Chul Cho and Kyung-Joong Kim, "Prediction of early stage opponent strategy for StarCraft AI using scouting and machine learning," *Proceedings of the Workshop at SIGGRAPH Asia*, ACM, 2012. p. 7-12.

[11] Johan Hagelback, "Potential-field based navigation in StarCraft," *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, pp.388-393, 11-14 Sept. 2012