

# Univector Field Method based Multi-Agent Navigation for Pursuit Problem

Hoang Huu Viet, Sang Hyeok An, and TaeChoong Chung\*

Artificial Intelligence Lab, Department of Computer Engineering, Kyung Hee University  
1-Seocheon, Giheung, Yongin, Gyeonggi, 446-701, South Korea

## Abstract

This paper presents a new approach to solve the pursuit problem based on a univector field method. In our proposed method, a set of eight agents works together instantaneously to find suitable moving directions and follow the univector field to pursue and capture a prey agent by surrounding it from eight directions in an infinite grid-world. In addition, a set of strategies is proposed to make the pursuit problem more realistic in the real world environment. This is a general approach, and it can be extended for an environment that contains static or moving obstacles. Experimental results show that our proposed algorithm is effective for the pursuit problem.

**Key words:** Pursuit problem, Predator agents, Prey agent, Univector field.

## 1. Introduction

The pursuit problem is a well-known class of test problems for the study of cooperative behavior in Distributed Artificial Intelligence (DAI) systems. This problem was first introduced by Benda et al. [1], and has recently been received a great deal of attention from the research community due to its important role in a wide range of applications, especially in some areas such as robotics and computer games. In Benda's formulation, a group of four predator agents try to capture a prey agent by surrounding it from four directions on a finite grid-world. Agent movements are limited to either a horizontal or a vertical step per time unit. The prey agent moves randomly, and two predator agents are allowed to occupy the same location. He proposed a solution to this problem by using computations of the center of gravity of the agents.

Gasser et al. [2] introduces an approach to this problem in which the grid-world is divided into four quadrants by using diagonal lines crossing at the prey agent's position. Each of the predator agents tries to occupy a position within their assigned quadrant to attain the "Lieb configuration". Once the predator agents have attained the "Lieb configuration", they use a set of rules called the "Lieb rules" that determine the movements of the predator agents to capture the prey agent. However, no experimental results were presented for the performance of this approach.

Stephens and Merx [3] present a version of the pursuit problem that does not allow two predator agents to occupy the

same position. The predator agents alternated moves and the prey agent moves randomly. In their approach, a centralized control mechanism is used to implement a scheme similar to the Lieb rules of [2] for the predator agents.

Korf [4] proposes a greedy algorithm to solve the pursuit problem in a finite grid of 100×100 cells. In his approach, each predator agent measures the distances between coordinates of its empty neighboring cells and the prey agent's coordinate, and then moves to a cell that minimizes distance to the prey agent's position. Three versions of the pursuit problem are presented. The first version is called the orthogonal game that corresponds to the original version, where agent movements are horizontal or vertical. He claims that a discretization of the continuous world with only horizontal and vertical movements is a poor approximation. Therefore, he develops the diagonal game that allows eight predator agents to move orthogonally, diagonally, and then surrounds the prey agent. Finally, the hexagonal game is implemented on an environment of hexagonal grid, where each cell has six neighboring cells and there are six predator agents in this game. In the first and second versions, a max norm distance metric is used by the predator agents to choose their next steps. However, the Euclidean distance metric is used instead of the max norm distance metric in the third version. He also improves the performance of the algorithm by adding an "attractive forces" between the predator agents and the prey agent, and "repulsive forces" between the predator agents. He assumes that the predator agents have the knowledge of existence of all the other predator agents and the prey agent. As a result, every single predator agent computes the "resultant force" to choose their next move.

In Korf's model, agents are able to see the complete world at once. Therefore, Reverte et al. [5] extend the Korf's ideas to improve his model. First, they propose a simple extension of Korf's fitness function and consider the problems related to a partial view of the world. Second, they propose a communication protocol to partially overcome them. They conclude that with these two simple extensions, Korf's ideas

---

Manuscript received Oct. 5, 2011; revised Feb. 8, 2012; accepted Mar. 3, 2012.

\* Corresponding author: TaeChoong Chung(tcchung@khu.ac.kr)

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (2010-0012609).

© The Korean Institute of Intelligent Systems. All rights reserved.

get results comparable to most machine learning approaches. However, simulations are limited in a finite grid of  $30 \times 30$  cells.

Kyung-Dal Cho [6] presents an approach to the pursuit problem for virtual robots based on an artificial neural network and a genetic algorithm. He claims that each robot needs to have both learning and evolution ability to adapt to a dynamic environment. Therefore the artificial neural network is used as a behavior decision controller. The input of this network is the existence of other robots and the distance to the other robots. The output of this network is the moving directions of robots at each instant time. The connection weight values of the artificial neural network are encoded as genes, and the fitness individuals are determined by using a genetic algorithm. In his proposed system, a group of 20 predator robots tries to pursue and capture a prey robot by surrounding it from four directions in a finite grid of  $10 \times 10$  cells. The robot movements are limited to either a horizontal or a vertical step per time unit. The predator robots and prey robot are supposed to be initialized at different spots on the grid to avoid the collision among robots.

Although all the approaches mentioned above might potentially solve the pursuit problem. However, in our opinion, these approaches are insufficient to implement in the real world environment for mobile robots. They are only applicable for the simulated pursuit games. Because when robots are used instead of agents in the pursuit problem, the orientation of the robots at each position of the environment need be considered. Further, the approaches mentioned above are only implemented in a finite grid-world. Univector field is a method for fast mobile robot navigation that considers to the position and orientation of a robot in its moving time. Based on the univector field method, we propose a relatively simple but effective approach for the pursuit problem that can be applied to the real world environment in the area of robotics. In our approach, a group of eight predator agents works together to find suitable moving directions and follows the univector field of the environment to pursue and capture a prey agent by surrounding it from eight directions in an infinite grid-world. In addition, strategies of the predator agents and prey agent are defined to test our proposed approach. The experimental results prove that the proposed approach guarantees to find a solution of the pursuit problem in the finite moving steps of the predator agents. Moreover, our approach provides a general solution since with slight modifications it can be applied to multi-agent real-time pursuit (MAPS) problem in a partially observable environment with obstacles.

The rest of this paper is organized as follows: Section 2 describes a short review of the univector field method. The proposed approach is described in Section 3. Section 4 shows the experimental results. Finally, we conclude our work in Section 5.

## 2. Univector Field Method

Univector field method is an improvement of the potential vector field which is designed for fast mobile robots [7, 8]. In the univector field method, the magnitude of each vector is unity at all positions, so a vector field contains only directional information. The univector field is classified into two categories. The first category is concerned with a robot going at the desired posture. The other is concerned with a robot avoiding obstacles. Since we only solve the pursuit problem in an obstacle-free environment, so the first category is considered. A univector field  $F(s)$  at the position  $s(x_s, y_s)$  is defined as in (1), where  $n$  is a positive constant,  $g(x_g, y_g)$  is the desired goal position,  $r(x_r, y_r)$  is a guidance point, and the symbol  $\angle$  denotes the angle of a vector mapped onto the range  $(-\pi, \pi]$ . The value of the univector field  $F(s)$  at the position  $s(x_s, y_s)$  is equivalent to the desired heading angle of the robot at the position  $s(x_s, y_s)$ . For instance, while implementing the action ‘‘shoot a ball’’ in a soccer robot game, the position  $s(x_s, y_s)$  will be the current position of the robot, the desired goal position  $g(x_g, y_g)$  will be the position of the ball, and the guidance point  $r(x_r, y_r)$  will be a point on the desired direction of the ball.

All univector fields constitute a univector field space of the environment. Fig. 1 depicts the univector field space with  $n = 10$ , where each tiny circle represents a position and the straight line attached to it represents the moving direction of the robot.

$$F(s) = \angle \overline{sg} - n\alpha, \text{ where } \alpha = \angle \overline{sr} - \angle \overline{sg},$$

$$\angle \overline{sr} = \arctan \frac{y_r - y_s}{x_r - x_s}, \text{ and } \angle \overline{sg} = \arctan \frac{y_g - y_s}{x_g - x_s} \quad (1)$$

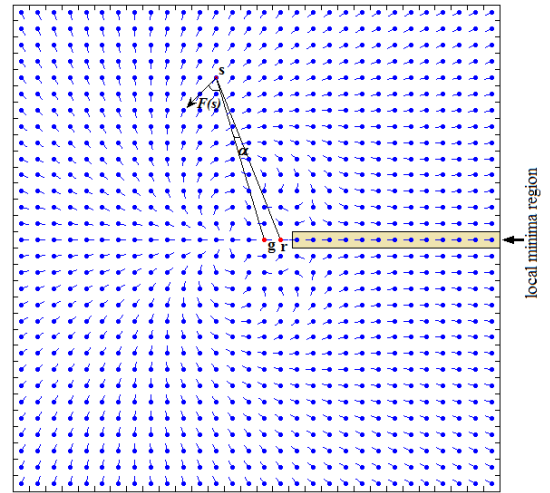


Fig.1. The univector field space of the environment.

It is clear that the larger  $n$  is, the smaller the  $F(s)$  is at the same robot position. Thus, if  $n$  increases, the univector field spreads out to a larger area, making the path to be traversed by the robot in reaching its goal larger. The univector field  $F(s)$  of the robot at the position  $s(x_s, y_s)$  changes according to the parameter  $n$  and the length of the line  $\overline{gr}$ .

By using the univector field method in an obstacle-free environment, from the most positions of the environment, a robot can reach to the goal position with a desired posture by following the univector field, except positions of the robot in the local minima region (i.e., the region on the direction of the vector  $\overline{gr}$ ). As shown in Fig. 1, if the robot locates in the local minima region, its guidance point is the position  $r(x_r, y_r)$ , and its task is to reach to one of eight neighboring positions of the goal position  $g(x_g, y_g)$ , except the point  $r(x_r, y_r)$ . By following the moving direction, when the robot is at the guidance point  $r(x_r, y_r)$ , there is no direction to move forward except the backward direction. However, the moving direction of the coming point (i.e., the right point of  $r(x_r, y_r)$ ) is also backward. As a result, the robot will move forward and backward between two these points (i.e., the point  $r$  and the right point of  $r$ ). This phenomenon is called the local minima problem. Note that the local minima region of the univector field method depends on the guidance point of the robot.

In our proposed pursuit problem, if two predator agents enter the local minima region and the guidance points of these two agents are symmetric through the prey agent's position  $g(x_g, y_g)$ , then eight predator agents cannot surround the prey agent from eight directions because these two predator agents will get stuck at one of these two guidance points. To avoid the local minima problem and to improve the successful capture rate of the predator agents, eight predator agents should exchange their guidance points among them.

### 3. The Pursuit Problem

In the classical pursuit games [1], a group of four predator agents try to capture a prey agent by surrounding it from four directions in a finite grid-world. In this paper, we assume that there are eight predator agents and a prey agent within the system where the agents move to any empty cells of eight neighboring cells, including horizontal move, vertical move and diagonal moves per time unit in an infinite grid-world. All agents are randomly initialized at the distinct positions. The solution for the pursuit problem is that the predator agents completely surround the prey agent as shown in Fig. 2. In this section, we propose strategies of agents and a Univector Field method based Multi-Agent Navigation (UFMAN) algorithm to solve this problem.

#### 3.1 Strategies of predator agents

**a) Greedy strategy:** It is assumed that each predator agent knows only the position of the prey agent at each instant time. In this strategy, each predator agent measures the distances from its position to empty neighboring cells of the prey agent's position, and choose the direction to move to a cell that minimizes the distance from its position. For instance, Fig. 3(a) depicts a situation of this strategy in which the predator agents 1, 2, 3, 5 and 7 choose a neighboring cell of the prey agent for each of them that minimizes distance from the neighboring cell

of the prey agent. Therefore, the predator agents 4, 6 and 8 have to choose a cell from the empty neighboring cells that has the minimum distance from its position to those cells.

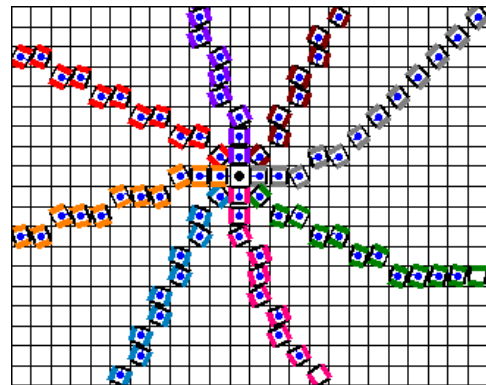
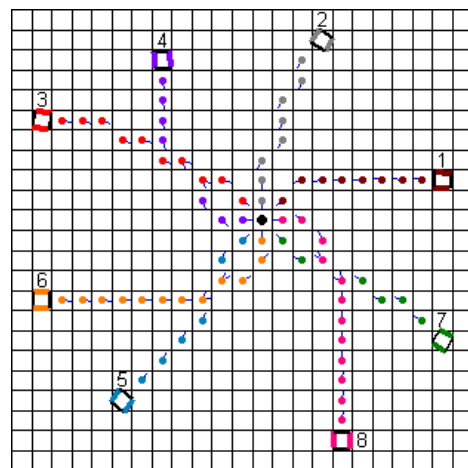
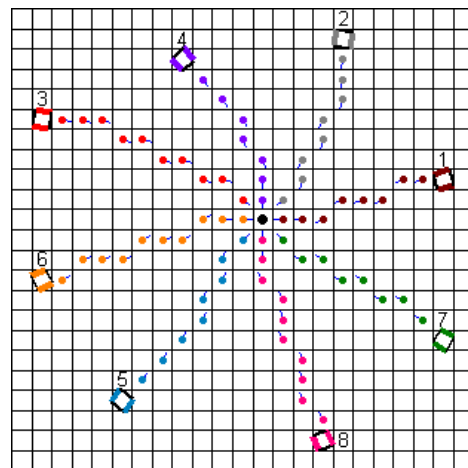


Fig. 2. A solution of the pursuit problem based on the univector field method, where the circle is the prey agent.



(a) Greedy strategy



(b) Cooperation strategy

Fig. 3. The capture directions of the predator agents based on the greedy strategy and the cooperation strategy.

**b) Cooperation strategy:** The cooperation is one of the most important strategies in making an agent community more

intelligent to achieve a common goal. The goal of the cooperation strategy is to share short-term benefits of agents to increase efficiency of overall agents for a task. Accordingly, we assume that each predator agent communicates to each other to know positions and orientations of the other predator agents and the prey agent at each instant time. The predator agents cooperate to determine a strategy that has the minimum sum of distances from all predator agents to eight neighboring cells of the prey agent. Since there are 8 predator agents in the system, so the predator agents have to choose on an optimal solution with the minimum sum of distances from eight factorial solutions. Fig. 3(b) shows a situation that the predator agents occupy positions as Fig. 3(a), but by using cooperation strategy, the predator agents share directions to obtain an optimal solution for surrounding the prey agent.

From Fig. 3, we can see that by applying the univector field method for the pursuit problem, the directions of predator agents 1, 4, 6 and 8 are changed when the capture direction of these agents changes.

**c) Mixed strategy:** In this strategy, we consider the pursuit problem related to a partial observation of the environment and propose a mixed strategy of the greedy strategy and the cooperation strategy. It is assumed that each predator agent knows the position of the prey agent at each instant time, but it only communicates its position and direction with some predator agents in a predefined radius. If a predator agent cannot communicate its information to the other predator agents, the greedy strategy is applied for this predator agent. Otherwise, the cooperation strategy is applied for predator agents in such a manner that they can communicate to each other.

### 3.2 Strategies of prey agent

**a) Random strategy:** It is assumed that prey agent does not know about positions and directions of the predator agents. At each instant time, the prey agent moves randomly to an empty neighboring cell from the set of its eight neighboring cells.

**b) Greedy strategy:** it is assumed that the prey agent knows about positions and directions of all predator agents at each instant time. In this strategy, the prey agent measures the distances to eight predator agents and moves an empty neighboring cell from the set of its eight neighboring cells that maximizes the distance to the nearest predator agent. However, if each predator agent moves at the same speed as the prey agent and the positions of eight predator agents are initialized at the same side of the prey agent's position, the predator agents cannot capture the prey agent. Therefore, we assume that the prey agent will move with a probability of  $q_0$  ( $0 < q_0 < 1$ ) and it will remain stationary with the probability of  $(1 - q_0)$ .

**c)  $\epsilon$ -greedy strategy:** The combination of the random strategy and the greedy strategy makes an  $\epsilon$ -greedy strategy. That is, with a small probability of  $\epsilon$  ( $0 < \epsilon < 1$ ), the random strategy is applied for the prey agent, and with the probability

of  $(1 - \epsilon)$ , the greedy strategy with  $q_0 = 1$  is applied for the prey agent.

### 3.3 The proposed algorithm

It is assumed that  $p$  (or  $p(x,y)$ ) is the position at the coordination  $(x,y)$  in the plane,  $A(k,p)$  is the current position of the  $k^{th}$  predator agent,  $g(x,y)$  is the current position of the prey agent,  $D(k,p)$  is the guidance point of the  $k^{th}$  predator agent, and  $F(k,p)$  is the value of the univector field at the position  $A(k,p)$ . The task of the  $k^{th}$  predator agent is to follow the univector field to surround and capture the prey agent. Since the agents are assumed to move to one of the eight directions, so the values of the univector field are divided into eight angular regions  $R_i$  ( $i = 1, 2, \dots, 8$ ) as in (2). Accordingly, when the value of the univector field  $F(k,p)$  belongs to the region  $R_i$ , the  $k^{th}$  agent takes the action  $a = i$ , where  $i = 1, 2, 3, 4, 5, 6, 7$ , or  $8$  is corresponding to the eight moving directions of the agents including East, North-East, North, North-West, West, South-West, South, or South-East as shown in Fig. 4. In general, the action selection rule of the  $k^{th}$  predator agent at the position  $A(k,p)$  is defined as in (3).

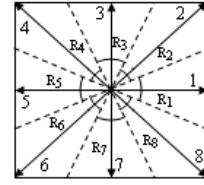


Fig. 4. Eight regions and the action selection rule at the position  $A(k,p)$ .

$$\begin{cases} R_1 = (-\pi/8, \pi/8] \\ R_2 = (\pi/8, 3\pi/8] \\ R_3 = (3\pi/8, 5\pi/8] \\ R_4 = (5\pi/8, 7\pi/8] \\ R_5 = (-\pi, -7\pi/8] \cup (7\pi/8, \pi] \\ R_6 = (-7\pi/8, -5\pi/8] \\ R_7 = (-5\pi/8, -3\pi/8] \\ R_8 = (-3\pi/8, -\pi/8] \end{cases} \quad (2)$$

$$\text{If } F(k,p) \in R_i, \text{ then } a = i \ (i = 1, \dots, 8) \quad (3)$$

With the univector field method and strategies of agents shown above, we propose an algorithm for solving the pursuit problem in an infinite grid-world as in Algorithm 1. Parameters of this algorithm are described in Table 1 and strategies of the agents are inferred from Table 2.

$N(4)$	$N(3)$	$N(2)$
$N(5)$	$g(x_g, y_g)$	$N(1)$
$N(6)$	$N(7)$	$N(8)$

Fig. 5. The set of eight neighboring cells of the prey agent's position  $g(x_g, y_g)$ .

**Table 1.** Notations used in UFMAN algorithm

$p(x,y)$	the position at the coordination $(x,y)$ of the grid-world.
$A(k,p)$	the current position of the $k^{th}$ predator agent ( $k = 1, \dots, 8$ ).
$F(k,p)$	the value of the univector field at $p(x,y)$ of the $k^{th}$ predator agent.
$g(x,y)$	the position at the coordination $(x,y)$ of the prey agent.
$D(k,g)$	the guidance point of the $k^{th}$ predator agent (i.e., a neighboring cell of the $g(x,y)$ ).
$\{N(i), i=1,2,\dots,8\}$	the set of eight neighboring cells of the prey agent's position $g(x,y)$ as shown in Fig. 5.
$\{J_N\}$	the subset of the set $\{1,2,\dots,8\}$ of which for any $i \in J_N$ , the cell $N(i)$ is unassigned to any predator agent.
$C(k,g)$	the neighboring cell of the prey agent's position that minimizes distance to the $k^{th}$ predator agent.
$r$	the radius parameter of the mixed strategy.
$\varepsilon$	the parameter of the $\varepsilon$ -greedy strategy.
$q_0$	the parameter of the greedy strategy of the prey agent.

**Table 2.** Strategies of the agents in UFMAN algorithm

Predators \ Prey	Greedy strategy	Cooperation strategy	Mixed strategy
Random strategy	$(r = 0)$ and $(\varepsilon = 1)$ and (any $q_0$ )	$(r = \infty)$ and $(\varepsilon = 1)$ and (any $q_0$ )	$(0 < r < \infty)$ and $(\varepsilon = 1)$ and (any $q_0$ )
Greedy strategy	$(r = 0)$ and $(\varepsilon = 0)$ and $(0 < q_0 < 1)$	$(r = \infty)$ and $(\varepsilon = 0)$ and $(0 < q_0 < 1)$	$(0 < r < \infty)$ and $(\varepsilon = 0)$ and $(0 < q_0 < 1)$
$\varepsilon$ -greedy strategy	$(r = 0)$ and $(0 < \varepsilon < 1)$ and $(q_0 = 1)$	$(r = \infty)$ and $(0 < \varepsilon < 1)$ and $(q_0 = 1)$	$(0 < r < \infty)$ and $(0 < \varepsilon < 1)$ and $(q_0 = 1)$

At each iteration  $t$  of UFMAN algorithm, each predator agent finds its community in a predefined radius  $r$ . If a predator agent does not find any the other predator agents in the radius  $r$ , it will find an empty neighboring cell  $C(k,g)$  of the prey agent's position that minimizes the distance to it. Otherwise, the predator agents within the radius  $r$  cooperate to determine a set of empty neighboring cells  $C(k,g)$  that have the minimum sum of distances from them to empty neighboring cells  $C(k,g)$ . Once all predator agents have determined their cells  $C(k,g)$ , the guidance points  $D(k,g)$  of agents are determined by symmetry  $C(k,g)$  ( $k = 1, \dots, 8$ ) through the prey agent's position  $g(x,y)$ . After having the guidance points  $D(k,g)$ , the values of univector field  $F(k,p)$  are determined and all predator agents then move to an empty neighboring cell of each predator agent based on its  $F(k,g)$ . For the prey agent, with a probability of  $\varepsilon$ , it moves to an its empty neighboring cell at random. Otherwise, with a probability of  $q_0$ , it moves to an empty neighboring cell that maximizes the distance to the nearest predator agent. The algorithm will end if the prey agent is captured by eight predator agents or after a predefined number of iterations.

**Algorithm 1.** UFMAN algorithm for the pursuit problem

1:	Initialize $A(k,p)$ arbitrarily for all $k = 1, \dots, 8$ .
2:	Initialize $g(x,y)$ arbitrarily
3:	$stop \leftarrow$ false;
4:	$t \leftarrow 0$ ;
5:	Repeat
6:	For $k = 1$ to 8 do
7:	$C(k,g) \leftarrow \emptyset$ ; (i.e., $C(k,g)$ is empty cell)
8:	If (the $k^{th}$ predator agent cannot find a predator in $r$ ) then {find the nearest empty $C(k,g)$ , i.e., greedy strategy}
9:	$j \leftarrow \arg \min_i (\ A(k,p) - N(i)\ ), i \in J_N$ ;
10:	$C(k,g) \leftarrow N(j)$ ;
11:	Endif
12:	Endfor
13:	Find empty cells $C(k,g)$ for predator agents at $A(k,g)$ within the radius $r$ based on the cooperation strategy.
14:	Determine the guidance points $D(k,g)$ of the predator agents by symmetry $C(k,g)$ through the prey agent's position $g(x,y)$ .
15:	For $k = 1$ to 8 do
16:	Compute the value of univector field $F(k,p)$ based on (1).
17:	The $k^{th}$ predator agent moves to a next empty cell based on $F(k,p)$ .
18:	Endfor
19:	If (eight predator agents capture the prey agent) then
20:	$stop \leftarrow$ true;
21:	Else
22:	If (a probability of $\varepsilon$ ) then
23:	The prey moves to an empty neighboring cell at random
24:	Else
25:	If (a probability of $q_0$ ) then
26:	The prey agent moves to the cell $N(i)$ ( $i = 1, \dots, 8$ ) that maximizes the distance to the nearest $A(k,p)$ ( $k = 1, \dots, 8$ ) (i.e., the greedy strategy).
27:	Endif
28:	Endif
29:	Endif
30:	$t \leftarrow t + 1$ ;
31:	Until ( $stop$ ) or ( $t < MAX\_ITER$ ).

### 4. Experiments

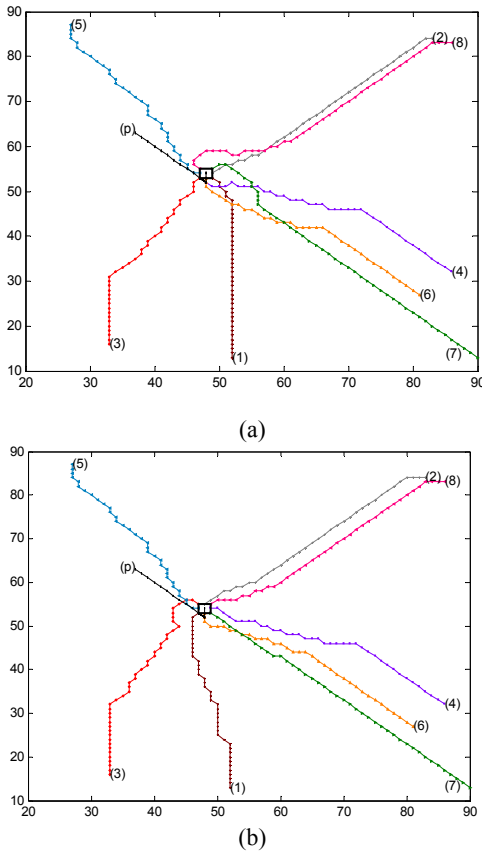
In this section, simulations using the Matlab software are implemented to estimate the efficiency of UFMAN algorithm for the pursuit problem. The environment of these simulations is represented by the cells of a uniform infinite grid-world.

First, we make 100 simulations to determine the successful capture rate of predator agents. In these simulations, all predator agents are randomly initialized at the distinct positions within the radius of 100 units from the prey agent's position. The coefficient of the univector field in equation (1) is set  $n = 5$ . The results of these simulations are shown in Table 3 with the parameter  $MAX\_ITER = 1,000$  and  $MAX\_ITER = 3,000$ . It is clear that if the predator agents use the cooperation strategy or mixed strategy, the successful capture rate of predator agents is

higher than that using the greedy strategy. Actually, when predator agents surround the prey agent within the radius  $r$ , the mixed strategy will become the cooperation strategy. If we increase the maximum iteration  $MAX\_ITER = 4,000$ , the successful capture rate of the predator agents using greedy strategy and prey agent using the greedy strategy remains unchanged (97%). The error found in this case is due to the local minima problem of the univector field method. Fig. 6(a) shows an example of this problem in which the prey agent is currently surrounded by seven predator agents, the 4<sup>th</sup> predator agent and the 6<sup>th</sup> predator agent get stuck in the local minimum trap. However, if the predator agents use the cooperation strategy instead of using the greedy strategy, this error can be solved as shown in Fig. 6(b).

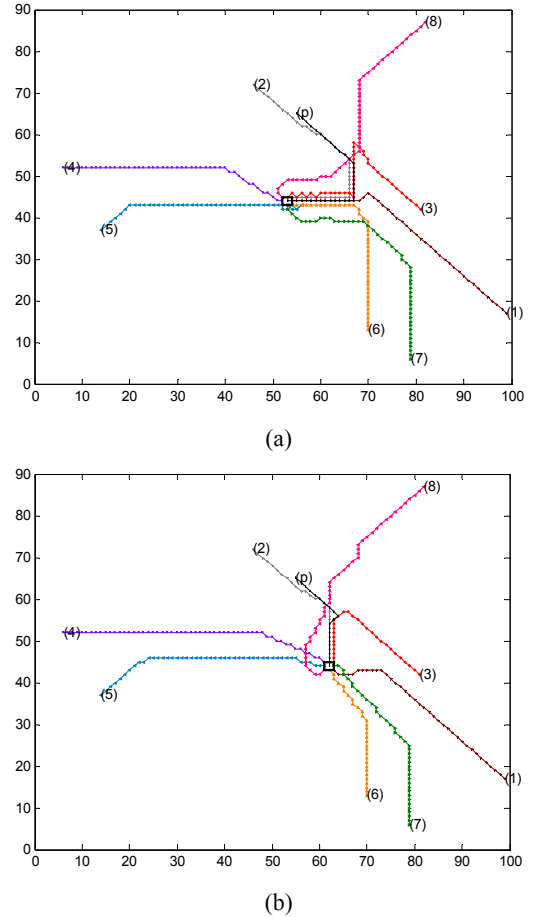
**Table 3.** The successful capture rate of predator agents after 1,000 iterations and 3,000 iterations.

Prey \ Predators	Greedy strategy	Cooperation strategy	Mixed strategy
	( $r = 0$ )	( $r = \infty$ )	( $r = 50$ )
<i>MAX_ITER = 1,000</i>			
Random strategy	100%	100%	100%
Greedy strategy ( $q_0 = 0.9$ )	91%	96%	96%
$\epsilon$ -greedy strategy ( $\epsilon = 0.05$ )	98%	99%	99%
<i>MAX_ITER = 3,000</i>			
Random strategy	100%	100%	100%
Greedy strategy ( $q_0 = 0.9$ )	97%	100%	100%
$\epsilon$ -greedy strategy ( $\epsilon = 0.05$ )	99%	100%	100%



**Fig. 6.** Trajectories are found by eight predator agents and prey agent. The small rectangle shows the prey agent's position.

Second, we make simulations to evaluate the sum of moves of eight predator agents until the prey agent is captured. Intuitively, predator agents using the cooperation strategy can surround and capture the prey agent faster than using the greedy strategy. In other words, the sum of moves of eight predator agents using the cooperation strategy is smaller than that of the predator agents using the greedy strategy. However, simulations show that this conclusion is misleading since during the navigation time, the prey agent always changes the moving direction depending on moving directions of eight predator agents. For instance, Fig. 7(a) depicts trajectories found by the predator agents using the cooperation strategy and the prey agent using the greedy strategy. In this simulation, eight predator agents need 365 moves to capture the prey agent. Fig. 7(b) depicts trajectories found by the predator agents using the greedy strategy and the prey agent using the greedy strategy. Eight predator agents only need 311 moves to capture the prey agent.



**Fig. 7.** Trajectories are found by eight predator agents and the prey agent. The small rectangle shows the position where the prey agent is captured.

Next, we make simulations in which eight predator agents are initialized from the same side of the prey agent's position. The parameters of these simulations are set the same as those of the first simulations. The results of these simulations show that

the prey agent is totally captured by eight predator agents. Fig. 8 depicts trajectories found by eight predator agents using the greedy strategy and the prey agent using: (a) random strategy, (b) greedy strategy, and (c)  $\epsilon$ -greedy strategy. The sum of moves of eight predator agents with respect to three cases (a), (b), and (c) is 412, 474, and 606 moves.

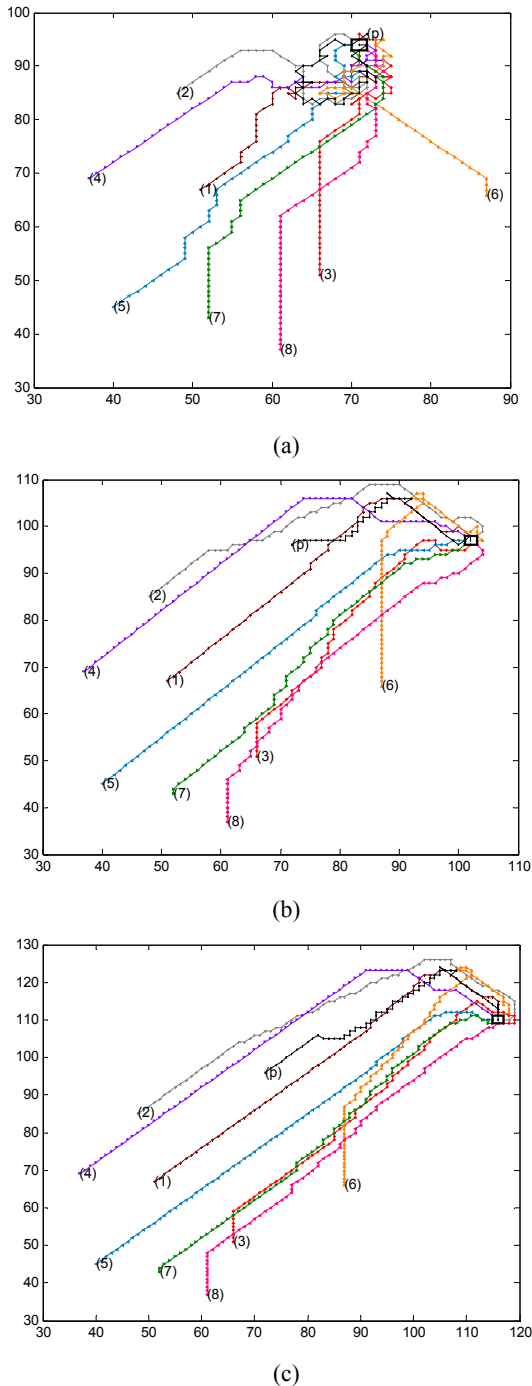


Fig. 8. Trajectories are found by eight predator agents and the prey agent. The small rectangle shows the position where the prey agent is captured.

Finally, we implement simulations to evaluate the different values of the coefficient  $n$  of equation (1). Specifically, 100

simulations are implemented to test the value of  $n = 10$  and 100 other ones are implemented to test the value of  $n = 15$ . When  $n$  increases, the path traversed by each predator agent in reaching its goal is larger. Therefore, the maximum iteration of these simulations is increased to  $MAX\_ITER = 5,000$ . The value of the other parameters and the results of these simulations are shown in Table 4. The simulation results show that although the coefficient  $n$  is changed, the successful capture rate of predator agents almost remains unchanged in comparing with the results of Table 3. The results of these simulations prove that the proposed approach is efficient for the pursuit problem in the infinite grid-world. Furthermore, if predator agents use the greedy strategy, they will get stuck in the local minimum trap and the unsuccessful capture rate of predator agents is about 11%. However, if predator agents apply the cooperation strategy or mixed strategy, the local minima problem of the univector field method is almost solved and the unsuccessful capture rate of predator agents is only about 3%.

In addition, the impact of other parameters on our proposed approach is analyzed as follows. First, if the value of  $q_0$  ( $0 < q_0 < 1$ ) is increased, the successful capture rate of predator agents will be decreased because the prey agent has many opportunities to move faster in its greedy strategy. Second, if the value of  $\epsilon$  ( $0 < \epsilon < 1$ ) is increased, the successful capture rate of predator agents will be increased because the prey agent has few opportunities to escape from predator agents in its  $\epsilon$ -greedy strategy. Finally, the successful capture rate of predator agents is not too dependent on the value of  $r$  ( $0 < r < \infty$ ) in the mixed strategy. Because when predator agents move close to the prey agent's position, the mixed strategy will become the cooperation strategy. Consequently, the successful capture rate of predator agents in these two strategies is approximately alike.

Table 4. The successful capture rate of predator agents after 5,000 iterations.

Prey \ Predators	Greedy strategy	Cooperation strategy	Mixed strategy
	( $r = 0$ )	( $r = \infty$ )	( $r = 50$ )
<b><math>n = 10</math></b>			
Random strategy	100%	100%	100%
Greedy strategy ( $q_0 = 0.9$ )	92%	98%	98%
$\epsilon$ -greedy strategy ( $\epsilon = 0.05$ )	98%	99%	99%
<b><math>n = 15</math></b>			
Random strategy	100%	100%	100%
Greedy strategy ( $q_0 = 0.9$ )	89%	97%	97%
$\epsilon$ -greedy strategy ( $\epsilon = 0.05$ )	97%	99%	99%

### 5. Conclusions

In this paper, we propose a new algorithm, UFMAN, to solve the pursuit problem in an infinite grid-world based on the univector field method. In addition, strategies of predator agents and prey agent are defined to test the proposed approach. For a predefined strategy, at each instant time, each predator

agent determines the value of the univector field at its position, and then follows the univector field to surround and capture the prey agent. The simulations show that the pursuit problem can be solved by a simple greedy strategy of predator agents without requirement for the cooperation strategy or the mixed strategy. However, if predator agents use the cooperation strategy or the mixed strategy, they always capture the prey agent in the finite moving steps. By experimental evaluation, we have successfully shown the application of UFMAN algorithm to the pursuit problem. However, we have only solved the well-known pursuit problem in an obstacle-free environment. In the future, we plan to extend the proposed approach to the pursuit problem in an environment with obstacles.

### References

- [1] Benda, M., Jagannathan, B., Dodhiawalla, R., "On optimal cooperation of knowledge sources," *Technical Report BCS-G2010-28*, Boeing AI Center, Boeing Computer Services, Seattle, Wa., July 1986.
- [2] Gasser, L., Rouquette, N., Hill, R. W., and Lieb, J., "Representing and using organizational knowledge in DAI systems," *Distributed Artificial Intelligence*, vol. 2, pp. 55-78, 1989.
- [3] Stephens, L., and Merx, M., "Agent organization as an effectors of DAI system performance," *Proc. of the Ninth Workshop on Distributed Artificial Intelligence*, Eastsound, Washington, pp. 263-292, 1989.
- [4] Korf, R.E., "A simple solution to pursuit games," *Proc. of the 11th International Workshop on Distributed Artificial Intelligence*, Glen Arbor, Mich, pp. 183-194, 1992.
- [5] Reverte, J., Gallego, F., Llorens, F., "Extending Korf's Ideas on the Pursuit Problem," *International Symposium on Distributed Computing and Artificial Intelligence 2008 - DCAI 2008*, Advances in Intelligent and Soft Computing, vol. 50, pp. 245-249, 2009.
- [6] Kyung-Dal Cho, "Learning of Emergent Behaviors in Collective Virtual Robots using ANN and Genetic Algorithm," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 4, no. 3, pp. 327-336, 2004.
- [7] Kim, J. H., Kim, D. H., Kim, Y. J., Seow, K.T, *Soccer Robotics*, Springer-Verlag, 2004.
- [8] Kim, Y. J., Kim, J. H., "Univector field navigation method for fast mobile robots in dynamic environment," *Proc. of the 2002 FIRA Robot World Congress*, pp.165-170, 2002.



**Hoang Huu Viet** received the B.S. degree in Mathematics from Vinh University, Nghean, Vietnam, in 1994, and the B.S. and M.S. degrees in Computer Science from Hanoi University of Technology, Hanoi, Vietnam, in 1998 and 2002, respectively. He is now working toward a Ph.D.

degree at the Artificial Intelligence Laboratory, Department of Computer Engineering, Kyung Hee University, Republic of Korea. His current research interests include Artificial Intelligence, Reinforcement Learning, and Robotics.

E-mail: viethh@khu.ac.kr



**Sang Hyeok An** received the B.S. and M.S. degrees in Department of Computer Engineering from Kyung Hee University, in 1999 and 2002, respectively. He is now working toward a Ph.D. degree at the Artificial Intelligence Laboratory, Department of Computer Engineering, Kyung Hee

University, Republic of Korea. His current research interests include Machine Learning, Heuristic Search, and Mobile Robots.

E-mail: ash@khu.ac.kr



**TaeChoong Chung** received the B.S. degree in Electronic Engineering from Seoul National University, Republic of Korea, in 1980, and the M.S. and Ph.D. degrees in Computer Science from KAIST, Republic of Korea, in 1982 and 1987, respectively. Since 1988, he has been with Department of

Computer Engineering, Kyung Hee University, Republic of Korea, where he is now a Professor. His research interests include AI, Machine Learning, Meta Search, and Robotics.

E-mail: tcchung@khu.ac.kr