**Proceedings of the 2001 IEEE/RSJ**
**International Conference on Intelligent Robots and Systems**
**Maui, Hawaii, USA, Oct. 29 – Nov. 03, 2001**

# Collision Avoidance Method for Multiple Autonomous Mobile Agents by Implicit Cooperation

ABE Yasuaki, MATSUO Yoshiki
Graduate School of Science and Engineering
Tokyo Institute of Technology

*Abstract*—**This paper proposes a collision avoidance method called the Cooperative Collision Avoidance (CCA) method for an environment consisting of multiple autonomous mobile agents. Especially, we assume a herd of agents where all agents have the same shape and the same algorithm. Furthermore, we also aim at high performance for tasks such as map exploration and object collection, in addition to collision avoidance. Considering implementation, we insist that the method should be based on local measurements and local planning at every sampling time in real-time. The proposed method is an extension of a conventional Velocity Obstacle (VO) method which provides collision detection among moving agents using the concept of Velocity Obstacle. As the VO method provides only collision detection and has no limitation to velocity selection, this method is appropriate to our objective, because freedom of velocity selection makes task operation easier. Besides, the proposed method utilizes uniformity of agents which enables agents share information of collisions without explicit communication. Sharing collision information allows agents to cooperate implicitly each other to avoid collision using a Common Velocity Obstacle (CVO). Furthermore, we introduce a velocity index which evaluates velocities to achieve good task performance and collision avoidance simultaneously. The velocity index is defined as a weighted sum of a task velocity index and a CVO velocity index, where the task velocity index gives a better evaluations to a velocity being expected to achieve good task performance, while the CVO velocity index penalizes velocities leading to future collision using the CVO. Therefore, agent can achieve good task performance and collision avoidance by selecting a velocity with the best evaluation of the velocity index. Finally, an efficiency of the Cooperative Collision Avoidance method is demonstrated by analyses and simulations on examples.**

*Keywords*—**Multiple Mobile Agents, Cooperative Task, Collision Avoidance, Velocity Obstacle, Cooperative Collision Avoidance**

## I. INTRODUCTION

Collision avoidance is an essential issue for mobile agents. This paper proposes a collision avoidance method for an environment consisting of multiple autonomous mobile agents. Especially, we introduce the Cooperative Collision Avoidance method, assuming uniformity of agents' algorithm. As this paper targets on implementation of a collision avoidance maneuver in dynamic environments, a proposed method must be real-time executable. To achieve real-time operation, we restrict a proposed method to being based on local planning at every sampling time, for global planning or time accumulating planning increases a calculation cost. Further, as global measurements need a global sensor system or a coordination of local measurements among agents, we assume agents can only obtain local measurements. In addition to collision avoidance, performance of task such as map exploration, object collection is also important. Therefore, agents must achieve tasks without a collision. However, as previous works deal collision avoidance and task performance separately or sequentially ([9]), collision avoidance influences task performances and vice versa. To achieve good task performance with collision avoidance, our method should consider both collision avoidance and task performance simultaneously.

Although there are many approaches for collision avoidance between moving obstacles, the most usual approach is the configuration space-time method([3], [7]). This method is extended from the configuration space method which is well discussed or applied for static obstacles. However, this method needs off-line search which is not suitable for our problem. Fiorini introduced the Velocity Obstacle (VO) method based on a relative velocity between agents ([1]). This method introduced a VO map showing a future collision in a velocity space. Assumption that other agents never change their velocity limits this method to environments where this assumption is reasonable –changes of other agents' velocities are small enough for the sampling time of the measurement. Prassler implemented this method into a robotic wheelchair ([6]). To ensure the assumption of other obstacles' constant velocities during the sampling period, this system has a laser range-finder as the measurement system. This sensor is so expensive that it does not suitable for multiple agent systems. The another problem of the VO method is that the VO method only gives a region leading to future collision and has no index for velocity selection. Although Fiorini proposed some strategies called "maximum velocity strategy", "to goal strategy" and etc. ([2]), these strategies first narrow freedom of velocity selection then select velocities, which discards merits of the VO method allowing unlimited velocity selection. In the case of focusing environments are limited to autonomous agents with same algorithm, specific approaches can be introduced. Detailed reviews of these researches are shown in [5]. These methods can be categorized into communicative methods and rule-based methods. A communicative method has problems on communication, cost, identification of agents and localiza-

**0-7803-6612-3/01/$10.00©2001 IEEE**

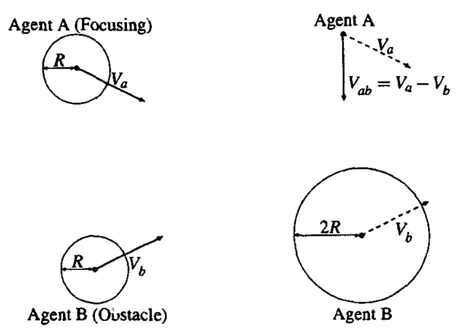
tion etc. A rule-based method is influenced by on-line conflicts between agents.

In this paper, noticing that the VO method firmly gives collision condition and no limitation to velocity selection, we extend the VO method and introduce the Cooperative Collision Avoidance (CCA) method. Assuming uniformity of other agents, the CCA method can remove the limitation on the VO method to allow other agents to change their velocities. It uses the Cooperative Velocity Obstacle (CVO) map to denote a collision. While each agent constructs the CVO map independently, the CVO map is shared among agents involved in the same collision through the scheme of the CCA method. The CVO map is also based on local measurements of agents and communications between agents is not needed. Although this method can be thought as a rule-based method, there exists only one rule –each agent follows a solution of the CVO map, which makes a conflict between agents never occurs. Further, we introduce a velocity index to evaluate velocities. The velocity index consists of a task velocity index and a CVO velocity index. Selecting a velocity with the highest evaluation achieves a collision avoidance and task operations simultaneously. In this paper, we demonstrate that the CCA method is more effective than the VO method in the considered environment, using both analyses and simulation. In addition, efficiency of the CCA method for longer sampling time is also shown in simulations. This implies that the proposed method is suitable for implementing on practical systems.

## II. Review of the Velocity Obstacle method

The conventional VO method is proposed in [1] and employs a velocity set in the velocity space to determine future collisions from the relative velocity to other agents. As this set can be thought as an obstacle in the velocity space, it is called the **Velocity Obstacle**.

Fig. 1-a shows general placements of two agents A and B. Derivation of VO of Agent A ($VO_a$) to Agent B is as



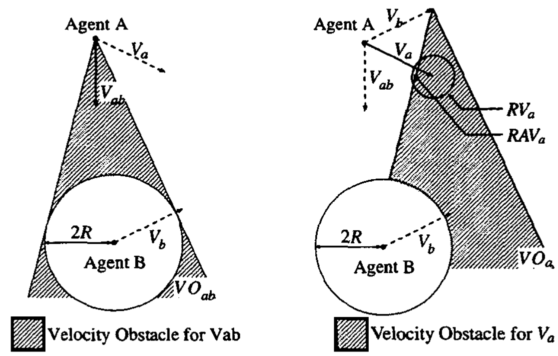

1-a:General placement of agents. 1-b:Equivalent condition for collision.

Fig. 1. Derivation of Velocity Obstacle

shown below. It is assumed that agents can measure only other agent's relative position, relative velocity and radius.

***Step 1*** Adding the radius of Agent A to the radius of Agent B and subtracting $V_b$ (the velocity of Agent B) from both of $V_a$ and $V_b$, the problem can be rewritten as a simpler problem shown in fig. 1-b –Agent A of radius 0, velocity $V_{ab} = V_a - V_b$ and Agent B of radius $2R$, velocity 0.

***Step 2*** The VO for $V_{ab}$ ($VO_{ab}$) is displayed as a hatched region in fig. 2-a. It is clear that if $V_{ab}$ lies inside of the $VO_{ab}$, Agent A and Agent B will collide in the future.




2-a:Velocity Obstacle for $V_{ab}$ 2-b:Velocity Obstacle for $V_a$

Fig. 2. Derivation of Velocity Obstacle

***Step 3*** Assuming constancy of $V_b$, adding $V_b$ to $V_{ab}$ and $VO_{ab}$ doesn't corrupt a condition of collision. Thus, fig. 2-b shows $VO_a$. Selecting $V_a$ pointing out of $VO_a$ can prevent a future collision. Namely, Agent A should select its velocity outside its $VO_a$ to avoid a collision.

***Step 4*** However, as every agent has limitations to the velocity selection according to it's kinematics and/or dynamics, agents cannot select velocities arbitrarily. In fig. 2-b, the velocity limitation is shown as an area called **Reachable Velocity (RV)**. Further, **Reachable Available Velocity (RAV)** is defined as RAV $=$ RV $\cap$ $\overline{\text{VO}}$ which enables the agent to select its velocity to avoid a collision within its mobility.

## III. The Cooperative Collision Avoidance method

### A. The CCA method

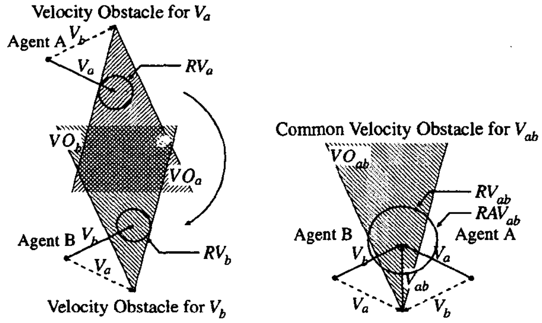
In this section, we propose the **Cooperative Collision Avoidance (CCA) method** by refining and suiting the VO method for a multiple agents environment, This method can remove the assumption of a constancy of other agent's velocities required in the VO method by implicitly sharing the information of a collision.

A highlight of this method is as follows:

- Generate the Common Velocity Obstacle map (the CVO map) between Agents involved in a collision.
- Determine the unique solution of the CVO map which guarantees a collision avoidance. Detailed description is shown in section IV.

The CVO map expresses overall of a collision. In the CCA method, it's shared among all agents involved in one collision. We show derivation of the CVO map below:

***Step 1*** The VO maps for two agents $VO_a$ and $VO_b$ are shown in fig. 3-a.




3-a:Velocity Obstacle for each agent

3-b:Velocity Obstacle for both agents

Fig. 3. CVO map configuration

***Step 2*** The one VO can be superposed on the another VO after rotated by 180 degrees, as these shapes are the same. Fig. 3-b shows $VO_a$ overlapped on $VO_b$, thus $VO_a, VO_b$ are denoted as $CVO_{ab}$. We call this map **the Common Velocity Obstacle (CVO) map**, as it is shared between two agents. It's worth noting that $RV_{ab}$ for the $CVO_{ab}$ becomes the additive set of $RV_a$ and $RV_b$ and broader. It is obvious by comparing $CVO_{ab}$ with $VO_a$ or $VO_b$ that collision avoidance on $CVO_{ab}$ means collision avoidance between two agents. Therefore the solution on the CVO map yields collision avoidance.

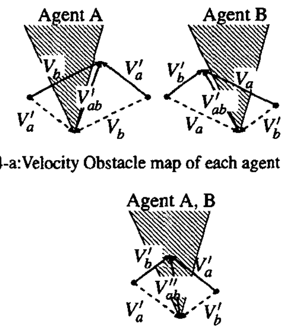
As each agent can build the CVO map from local measurements separately, no communication is needed.

### *B. Comparison between the two method*

We compare the proposed CCA method with the conventional VO method to illustrate that the CCA method can solve problems where the VO method cannot find any collision avoidance velocities –empty set of RAV– and where agent's collision avoidance velocities conflict and a collision avoidance is not achieved.

*Example 1:* As the VO method assumes constancy of velocity of the other agent, avoidance behaviors of individuals may conflict. However, by the CCA method, if only agents use the same decision algorithm, conflict of avoidance will never occur.

In fig. 4-a, each agent selects velocity to avoid collision on their own the VO map according to the conventional VO method, where "′" stands for a corrected velocity selected by agent aiming at collision avoidance. On the other hand, fig. 4-b shows conflict of collision avoidance according to the proposed CCA method, nevertheless each agent is aiming to avoid collision in fig. 4-a. This conflict is caused by the assumption that other agents will never change their velocity. In reality, this leads to cancellation of each agent's avoidance operation at the next sampling time. In addition to this situation, an existence of an amplification of avoidance may happen.



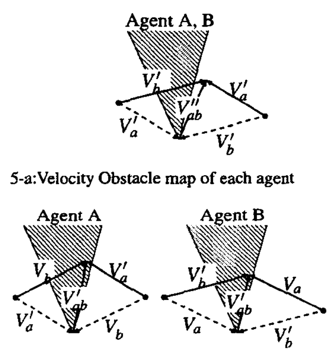

4-a:Velocity Obstacle map of each agent

4-b:Common Velocity Obstacle map

Fig. 4. Comparison between VO and CVO 1

*Example 2:* When the collision avoidance velocity exists on the CVO map, there may be no feasible solution on the VO map of each agent. That is to say, there exist cases that the proposed CCA method can find solutions which the conventional VO method cannot find.

It is assumed that $V_a'$ and $V_b'$ on the CVO map lead to $V_{ab}''$. Fig. 5-b shows $V_a'$ and $V_b'$ on the VO map given on the CVO map in fig. 5-a. As the relative velocity $V_{ab}'$ is lying inside of each agent's VO map, $V_a'$ and $V_b'$ cannot be regarded as collision avoidance velocities according to the conventional VO method. However, in fig. 5-b, these velocities lead to collision avoidance relative velocity $V_{ab}''$ and cannot be regarded as a collision avoidance velocity. In short, the CCA method can find solutions which the VO method dismisses.




5-a:Velocity Obstacle map of each agent

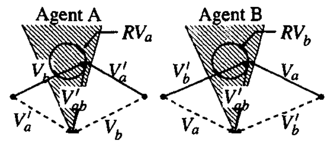
5-b:Common Velocity Obstacle map

Fig. 5. Comparison between VO and CVO 2

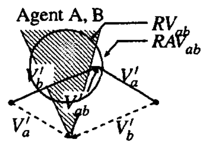
*Example 3:* If RAV is empty set, both the VO method and the CCA method cannot avoid collision. However, as the CCA method has a broader RAV, $RAV_{ab}$ may not be empty set, even if $RAV_a$ and $RAV_b$ are empty set. That is to say, the CCA method can solve this problem, while the VO method cannot find any solutions. This example is a special case of example 2.

Fig. 6-a shows that $RAV_a$ and $RAV_b$ are empty set, which means the conventional VO method cannot yield collision avoidance velocities in this sampling period.

On the other hand, the proposed CCA method yields collision avoidance solutions according to the broader $RAV_{ab}$.




6-a:Velocity Obstacle map of each agent




6-b:Common Velocity Obstacle map

Fig. 6. Comparison between VO and CVO 3

These examples show that the proposed CCA method can prevent misleading of collision avoidance and derive broader solutions than the conventional VO method can.

## IV. Implementation of the CCA method

The principle of the CCA method is introduced in the last section. Additional issues on the CCA method are discussed in this section.

### A. The Velocity Index

As the CCA method provide a set of collision avoidance velocities, agents need an index to select a unique collision avoidance velocity from the set. To provide an index of velocity selection for agents, we introduce the concept of velocity index which evaluates feasible velocities. Using the velocity index, task oriented navigation and collision avoidance are achieved simultaneously, because both of them are denoted as velocity indices and can be layered each other. In addition, also an RV can be denoted as a velocity index, planning of navigation is only solving a CVO map defined by the velocity index. Detail of a velocity index for an RV is discussed in the next subsection.

A task oriented velocity index is to give a better evaluations to a velocity being expected to achieve the task better, and a CVO oriented velocity index penalizes evaluations inside CVO. Adding a task velocity index to a CVO velocity index yields a total velocity index. An agent selects a velocity with the best evaluation and can achieve both a good performance task operation and a collision avoidance.

An example of the CVO velocity index is shown in fig. 7. Although an actual CVO map consist of 4 dimensions, fig. 7 depicts only 2 dimensions where the $x$ axis and the $y$ axis indicate velocity variances where the origin of the axes means that the agent keeps the current velocity. The $z$ axis indicates the velocity index. In this example, the CVO velocity index gives lower penalty to a later collision. This prevents task oriented navigation from being influenced by later less possible collisions.

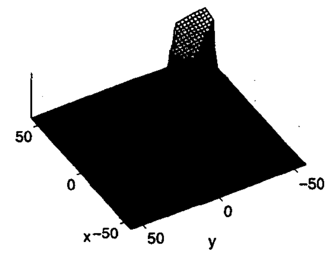



Fig. 7. The velocity index of the CVO map

Now, we consider a task velocity index for a simple task. The task is navigation to a destination in a shorter time and with lower energy consumption. Then an evaluation function $E(\Delta V_x, \Delta V_y)$ can be defined as a weighted sum of a time oriented evaluation $E_t(\Delta V_x, \Delta V_y)$ and an energy oriented evaluation $E_e(\Delta V_x, \Delta V_y)$ with respect to velocity variances in both directions $\Delta V_x$ and $\Delta V_y$, where the $x$ axis is set to the heading direction of the agent and the $y$ is set orthogonal to the $x$ axis. $E_t$ denotes the ratio between reduction of the distance to the destination and the maximum reduction of the distance (in the case of going straight to the destination with maximum velocity) both in one sampling period. $E_e$ denotes the maximum decrease and consumption of energy. $E_t$ and $E_e$ are in the range of [0, 1] and weighted by $\alpha$ and $1-\alpha$, respectively, where $\alpha$ is a constatant which determines a weight of two evaluations. To simplify the notation, we use $v$, $l$, $\theta$ and $\Delta V$ insted of $\Delta V_x$, $\Delta V_y$,

$$v = \sqrt{(V_x+\Delta V_x)^2 + (V_y+\Delta V_y)^2}, \quad l = \sqrt{tx^2+ty^2}$$

$$\theta = \tan^{-1}\frac{ty}{tx} - \tan^{-1}\frac{V_y+\Delta V_y}{V_x+\Delta V_x}, \quad \Delta V = \sqrt{\Delta V_x^2 + \Delta V_y^2},$$

where $v$: the absolute value of the velocity; $\theta$: the angle to the destination; $\Delta V$: the absolute value of the velocity variance; $l$: the distance to the destination; $(tx, ty)$: the destination; $(V_x, V_y)$: current velocity.

Then,

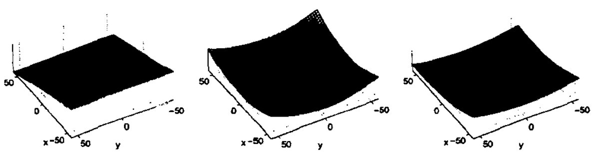
$$E(v,\theta,\Delta V) = \alpha E_t(v,\theta) + (1-\alpha)E_e(\Delta V)$$

$$E_t(v,\theta) = \frac{V_{\max} - l + \sqrt{v^2+l^2+2vl\cos\theta}}{2V_{\max}}$$

$$E_e(\Delta V) = \frac{\Delta V}{A_{\max}^2},$$

where $V_{\max}, A_{\max}$ are the maximum velocity and the maximum acceleration of the agent.

Fig. 8 depicts examples of $E_t$, $E_e$ and $E$ using velocity indices. As the destination is set on the $+x$ dirction, $E_t$ in fig. 8-a gets smaller according as $x$ increases. This motivates agents to select a velocity faster in the $x$ direction. On the other hand, $E_e$ in fig. 8-b prevents agents from selecting velocities with large changes. Therefore, the best velocity evaluated with $E$ in fig. 8-c navigates the agent to the destination without huge energy comsumption in one sampling period. As there is no limitation to the velocity in $E$, the agent selects faster velocities as sampling time advances. Limitations by agent's kinematics will be introduced in the following.



8-a:$E_t$ 8-b:$E_e$ 8-c:$E$

Fig. 8. Task velocity index

*B. Reachable Velocity*

An RV represents agent's limitation in the velocity field considering kinematics and/or dynamics. The conventional VO method defines an RV as a region in which the agent can select its velocity. Although the proposed CCA method also defines an RV as a region, as the CVO map is expressed as a velocity index, an RV can be also described by a velocity index. We will explain two examples to demonstrate that an RV can be expressed as a velocity index.

First, in the case of an omni-directional agent with constant limitations on both the velocity and the acceleration, the resulting RV is easily obtained and shown in fig. 9-a.

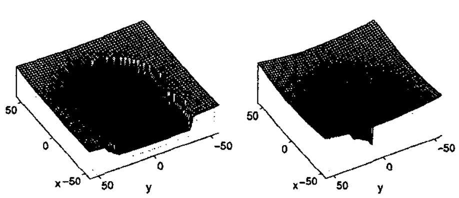
Second, in the case of a two-wheel differential-drive agent with its radius $D$, the RV is described as

$$\begin{aligned} &\text{if} \quad \mathrm{V_l = V_r} \quad \text{then} \quad \mathrm{(V_x, V_y) = (V_l = V_r, 0)} \\ &\text{else} \quad \mathrm{(V_x, V_y) = (l \sin\theta, l(1-\cos\theta))} \\ &\text{where} \quad \mathrm{l = \frac{D}{2}\frac{V_r + V_l}{V_r - V_l}}, \quad \theta = \mathrm{\frac{V_r - V_l}{D}}, \end{aligned}$$

where $V_l$, $V_r$ and $(V_x, V_y)$ are the velocities of the left wheel, the right wheel and the center of agent. $l$ is the distance to the center of rotation, and $\theta$ is the rotating angle. Therefore, the RV is a region bounded by

$$(V_x, V_y) = \begin{cases} (V_x, V_y) | (V_l = V_{l_{\max}}, V_{r_{\min}} \le V_r \le V_{r_{\max}}) \\ (V_x, V_y) | (V_{l_{\min}} \le V_l \le V_{l_{\max}}, V_r = V_{r_{\max}}) \\ (V_x, V_y) | (V_l = V_{l_{\min}}, V_{r_{\min}} \le V_r \le V_{r_{\max}}) \\ (V_x, V_y) | (V_{l_{\min}} \le V_l \le V_{l_{\max}}, V_r = V_{r_{\min}}). \end{cases}$$

The resulting RV using velocity index is shown in fig. 9-b.

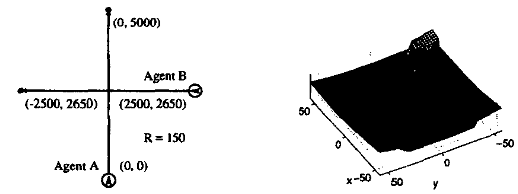


9-a:Omni-directional 9-b:Differential-drive

Fig. 9. Velocity indices of RVs

## V. Validation of the proposed CCA method

In this section, we apply the VO method and the CCA method to an example shown in fig. 10-a and validate effectiveness of the CCA method. The VO method uses "to goal strategy", and the CCA method uses a time and energy based velocity index introduced in section IV-A.



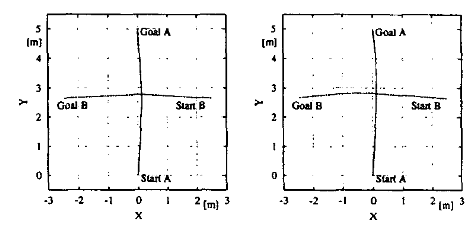

10-a:Geometric setup 10-b:Total velocity index

Fig. 10. Simulation setup

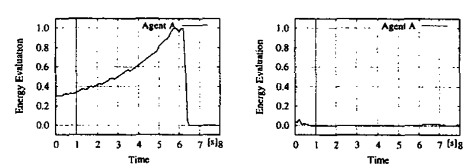
Two agents are assumed to have the same specifications of the radius 150 [mm], the maximum velocity 400 [mm/s], and the maximum acceleration 800 [mm/s]. The sampling time is 100 [ms] for fig. 12, 11, and 1 [s] for fig. 13. All graph plots show results of the period from 0 [s] to 8 [s] though the total time of navigation is about 13 [s], as collision avoidance is performed during the period.

Fig. 10-b shows the total velocity index used for simulations in the following, where the type of agent is the omni-directional.

Fig. 11 shows the resulting trajectories and evaluations of energy consumption and fig. 12 shows the heading angle of Agent A. As sampling time is small enough, a difference between paths is not noticeable. However, a difference between evaluations of energy consumption is prominent. The CCA method needs less energy than the VO method and uses energy effectively for collision avoidance. On the other hand, the VO method wastes a lot of energy according to the flapping actions shown in fig. 12-a.

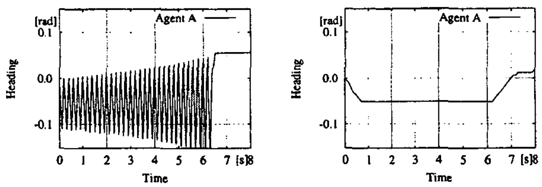



11-a:The VO method 11-b:The CCA method



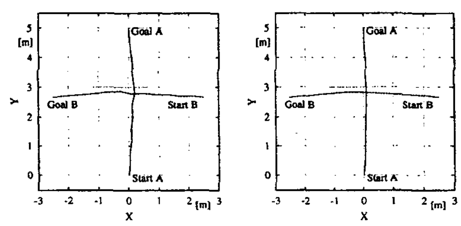

11-c:The VO method 11-d:The CCA method

Fig. 11. Results of trajectory and energy evaluation

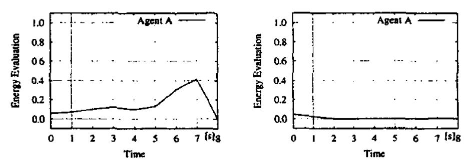



12-a:The VO method 12-b:The CCA method
Fig. 12. Headings of agents

In the case of sampling time 1 [s], graph plots corresponding to fig. 11 are shown in fig. 13. Whereas resulting trajectories are similar to those of sampling time 0.1 [s], there exits a clear difference between fig. 13-a and fig. 13-b. The CCA method still achieves smooth navigation, while the VO method leads to meandering trajectories

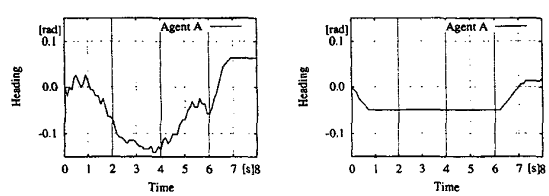



13-a:VO Method 13-b:CCA Method




13-c:VO Method 13-d:CCA Method
Fig. 13. Simulation results for a longer sampling time

Simulation results under existence of measurement noise are shown in fig. 14. Measurement noise is given as normal distribution with a deviation of 5% length of distance to other agents. Flapping action is observed during first 1 second, then a collision avoidance is achieved after that. Although there is no graph plot, the larger the deviation of measurement noise is, the longer a period of flapping becomes. This motivates us to introduce a field of view proportional to measurement noise. Detailed descriptions and discussions will appear in a future paper.




14-a:W/ Measurement Noise 14-b:W/O Measurement Noise
Fig. 14. Simulation results under existence of measurement error

## VI. Conclusion

This paper proposed the CCA method by extending the VO method and applying to multiple mobile agent environment, removing assumption of other agent's constant velocities. This method uses the CVO map instead of the conventional VO map to express future collisions. Thereby, the proposed method can handle problems which the conventional VO method cannot solve. This paper also introduced the velocity index to evaluate feasible velocities, which enables to treat task performance and collision avoidance simultaneously. In other words, the combination of the CCA method and the velocity index makes a framework for task operation with collision avoidance.

Validity of this scheme is examined by both analysis and simulation on an example setting. The effect on eliminating flapping motion which may occur with the VO method confirmed in simulation. It's worth noting that the CCA method also works well for a longer sampling time. This allows agents to use slower sensors such as a CCD, which makes the CCA method more attractive for practical use.

To optimize the solution, it is important to discuss on optimization methods. As a future study issue, the solution in this paper cannot optimize the task performance globally, because the task velocity index is based on local measurements only.

## Acknowledgment

This research is supported by The Grant-in Aid for COE Research Project "Super Mechano-Systems" by The Ministry of Education, Science, Sport and Culture.

## References

[1] FIORINI, P. and SHILLER, Z., "Motion Planning in Dynamic Environments Using the Relative Velocity Paradigm", *Proc. IEEE Int. Conf. Robotics Automat.*, Vol. 1, pp. 560-565, 1993.

[2] FIORINI, P. and SHILLER, Z., "Motion Planning in Dynamic Environments Using Velocity Obstacles", *Int. J. Robotics Research*, Vol. 17, No. 7, pp. 760-772, 1998.

[3] ERDMANN, P. and LOZANO-PÉREZ, T., "On Multiple Moving Objects", *IEEE Int. Conf. Robotics Automat.*, pp.1419-1424, 1986.

[4] YUTA, S., "Cooperation of Multiple Autonomous Mobile Robot (in Japanese)", *J. Robotics Society of Japan*, Vol. 10, No. 4, pp.433-438, 1992.

[5] ARAI, T. and OHTA, J., "Planning of Multiple Mobile Robots (in Japanese)", *J. Robotics Society of Japan*, Vol. 10, No. 4, pp.444-449, 1992.

[6] PRASSLER, E. et.al., "MAid: A Robotic Wheelchair Operating in Public Environments", *Sensor Based Intelligent Robots*, LNAI 1724, pp.68-95, 1999.

[7] FUJIMURA, K. and SAMET, H., "A Hierarchical Strategy for Path Planning Among Moving Obstacles", *IEEE Trans. Robot. Automat.*, Vol. 5, No. 1, pp.61-69, 1989.

[8] HAYWARD, V, etc., "Efficient Collision Prediction Among Many Moving Objects", *Int. J. Robitics Research*, Vol. 14, No. 2, pp.129-143, 1995.

[9] FRAICHARD, T. and LAUGIER, C., "Path-Velocity Decomposition Revisited and Applied to Dynamic Trajectory Planning", *IEEE Int. Conf. Automat. and Rotot.*, Vol. 2, pp.40-45, 1993.

[10] FIORINI, P. and SHILLER, Z., "Time Optimal Trajectory Planning in Dynamic Environments", *Proc. Int. Conf. Robot. and Automat.*, pp.1553-1558, 1996.