A Human Inspired Collision Avoidance Strategy for Moving Agents

Pejman Kamkarian
Electrical and Computer Engineering Department,
Southern Illinois University, Carbondale, IL 62901, USA
Email: pejman@siu.edu

Henry Hexmoor
Computer Science Department, Southern Illinois
University, Carbondale, IL 62901, USA
Email: hexmoor@cs.siu.edu

Abstract- This paper presents an approach for controlling collision avoidance among a group of moving multi-agents such that they are not able to communicate with each other and hence, cannot share information. The basics and key features of our collision control algorithm are discussed to include practical examinations. Our approach is based on multi-agent systems and help moving agents to pursue their goals using collision free routes. In terms of validating our solution, we plan to apply into a configuration set of agents located in our experimental space. We also explain our solution algorithm that we have developed, along with the examination that we subjected it to, as well as sketching some of the most important challenges that remain to be addresses in our future researches.

Keywords- Multi-agents; Moving agents; Agents collision control; Intelligent agents; Agents decision making system.

I. INTRODUCTION

In one class of mobile robotics research heralded by [1], sets of rules are used for collision avoidance. Rules correspond to sensing abilities of the mobile robots. Assuming that each robot has abilities to detect collisions and can measure the distance to a colliding robot and its velocity in the forward direction of the avoiding robot, the following two rules are used. In these rules, the robot can know the location of the colliding robot by using sensors for detecting collisions and can judge whether it is approaching or leaving by measuring its velocity. At first, the robot will attempt to apply the rules without communications. If the rules do not apply because of an environmental restriction, the robot stops and starts communicating to determine a way out of the deadlock.

(1) If the colliding robot is located in front and near and it is approaching, then avoid it from the left.

(2) If the colliding robot is located in front and near and it is departing, then stop locomotion for predetermined time duration.

When rules are not applicable, in another class of mobile robotics research, collision avoidance is a negotiated activity between pairs of robots that have potential collisions. During the message exchange of warning and its reply, priorities of each robot are reported to each other. Mobile Robot 1 detects the collision, and after it takes into account the priorities, it determines if it is reasonable for robot 1 to avoid collision due to high priority. If robot 2 detects the collision, reverse processes are executed. If the priority of robot 2 is found to be higher than robot 1 as a result of negotiation, robot 1 sends to robot 2 a declaration to proceed instead of a command to wait for robot 1. Then, robot 2 moves to avoid collision and sends a command to restart to robot1. Alternatively, the coordination algorithm is useful when groups of robots move in opposite directions and while navigating or when a specific region is a target for many robots, in particular [9]. The main goal in the coordination algorithm is that it forces robots to wait while the other robots continue to move to their target, and then allows the remaining robots to move. Consequently, with this coordination, the congestion problem will be decreased, and at the same time, the percentage of reaching robots to their target will be increased. We used a version of this priority scheme in an earlier paper [6].
There have been several attempts to use human behavior as inspiration. Human inspired methodology is summarized in the following three steps [8].

(1) Keep your direction and velocity of motion if there is minor possibility of a collision.

(2) Else, if a major possibility of a collision exists, and there are no ways to manoeuver around it; then, stop to let the other person to continue to move in their direction and velocity,

(3) Else, if a possibility of a collision exists, and there is a way to manoeuver around it; then Change your direction of movement with slightly changing speed to around the other person, and joining back to your original path of motion.

The strategy proposed in this paper is also largely human inspired and it can be applied to robots in crowded environments. We begin by an outline of conditions and premises in section 2. We then examine a magnified scenario of encounters between a pair of individuals in collision sites in section 3. Spiral orbits as a strategy for collision aversion is discussed in section 4. Experimental results are discussed in section 5 followed by conclusions in section 6.

II. CONDITIONS AND PREMISES

There are several terms and conditions that need to be considered in regards to developing a suitable strategy that is able to avert collisions among multiple moving agents as they move toward their goals located in a site. There are two entities available on our two dimensional space; agents and goals.

Below, we outline the most important features that are directly and indirectly pertinent for elaborating our solution.

(1) Agents can be located at arbitrary position on our feasible two dimensional spaces. There are limitations on their numbers; however, the total number of agents at any cycle of experiment will not exceed the total number of goals. Our experimental space, has the following default criteria: $EC = \{EC_1, EC_2, \dots, EC_i\}$, $EC_i = \{A_i, G_j\}, A_i = \{a_1, a_2, \dots, a_i\}, G_j = \{g_1, g_2, \dots, g_j\}, \{\forall(i,j) \in \mathbb{N} \mid (i \in A, j \in G) : i \geq j\}$, where EC, A_i and G_j present experimental cycles, agents and goal respectively.

(2) There is a one to one correspondence between agent groups and the goal collection, which means that each goal will be assigned to a unique, single agent. As a matter of course, each agent maintains its goal during cycles of experimentation. This is captures in the mapping$f: X \to Y,$ $where\ set\ X\ indicates\ the\ agent\ set\ and\ Y\ represents\ the\ goal\ set; f(A_i) = G_i.$

(3) Each agent starts moving at any time during the experiment. $T = \{\Delta t_{a_1}, \Delta t_{a_2}, \dots, \Delta t_{a_i}\}$; $\forall(\Delta t_{a_{i-1}}, \Delta t_{a_i}) \in T, \Delta t_{a_{i-1}} \neq \Delta t_{a_i}$, where Δt_{a_i} , is the start time for a_i. In other words, there is no common time for each agent to start moving toward its goal within experiment cycles.

(4) All agents are assumed to have the same uniform size and shape as pointed out in many other sources such as [2], [5], [11].

(5) Each agent has its own distinctive communication protocol; however they are all use the same collision control method to handle collision sites, if occurred. As a matter of fact, as opposed to those who employ a method of communications among multiple moving agents for their verifications, such as [2], [5], we assumed that agents do not have any connection to each other and hence are not able to disseminate information among themselves during a period of experiment.

(6) Moving toward goals is not necessarily bounded on a straight path, which means, each agent is able to select a route with any speed based on the situation.

(7) Goals can be located at any arbitrary location in two-dimensional space; however, they cannot be relocated and changed their positions during each cycle of an experiment. Agents know their exact location as well as their goal locations; $a_i = \{c_{(x,y)_i}, g_{(x',y')_i}\}$ where $c_{(x,y)_i}$is the coordinate of the agent and $g_{(x',y')_i}$ is the coordinate of the goal location. Each agent is capable of calculating its location and also its corresponding goal location at any needed time during each experimental cycle. An effectual way to achieve this goal is to equip agents with a vision sensor device which is already demonstrated in former publications such as optical motion planning mapping in [7], [10], or motion sensor themselves that are presented in [4], [3] and [12].

III. COLLISION SITE

Each agent is able to recognize a limit site (i.e., a region) located in front of its vision sensor, which is computed by $vs = \int_0^{\theta}\int_0^{r} dS = \int_0^{\theta}\int_0^{r} \tilde{r} d\tilde{r} d\tilde{\theta} = \int_0^{\theta} \frac{1}{2} r^2 \theta$, and based on that, it decides and determines the best possible path toward its goal at any time. In this formula, r and θ denote maximum vision depth and vision range angle for each agent vision sensor, respectively. We define a collision site form, if an agent detects another agent in its vision site range. In such situations, orientations of those agents participating to form collision sites are not taken into account. Figure 1 shows a prototypical collision site.

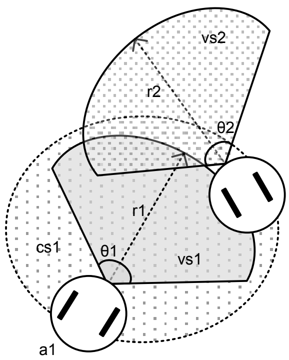

Fig 1. a_1 detects a_2 on its vision site (vs_1), and hence collision site (cs_1) is formed

All agents that are forming such collision sites will use our solution as a potential collision control strategy to enter and exit those sites. The process of handling a collision site consists of two phases. The first phase is forming a distance set $cs_i = \{D_j a_{j(x,y)_j}, D_k a_{k(x,y)_k}, \dots\}$, where $D_j a_{j(x,y)_j}$, indicates distance between a_j and a_i. The second phase is to form the smallest circular area that contains the agent that has the nearest distance from a_i where both agents located on the perimeter of it. For instance, a_i, forms $(x-l)^2 + (y-k)^2 = r^2$, where x, y, l, k, and r, are coordinates for a_i and a_j respectively, if $\forall(a_j, a_k) \in cs_i$, $D_j a_{j(x,y)_j} < D_k a_{k(x,y)_k}$. a_i, then starts moving toward a temporary expansion spiral route computed by $r = pe^{q\theta}$, where e indicates the base of natural logarithms, and p and q are parametric positive real constant values respectively, as polar coordinates, (r, θ), shown as $\overrightarrow{VR_1}$ route, in Figure 2, around a virtual collision site circle formed by a_i and a_j. We present our general path finder solution along with relative formulas in next section.

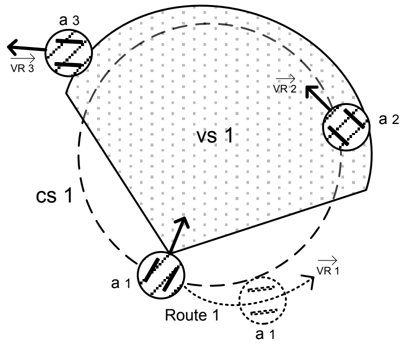

Fig 2. Collision site formed between a_1, a_2, and a_3, once a_1 detected a_2, and a_3 on its vision site. a_1 hence, starts moving on $\overrightarrow{VR_1}$ route

IV. COLLISION AVOIDENCE USING VIRTUAL EXPANSION SPIRAL ORBIT SOLUTION

In this paper, we concentrated on developing a strategy, which is capable of preventing a group of agents to strike each other while moving toward their goals. In order to achieve our objective, we developed an algorithm that each agent in our experimental space uses to determine the safest, short route at each decision making cycle. This consists of two general strategies that will be used in each presumed situation. The path finder algorithm determines a straight line connecting path as the shortest route toward agent goal during the times that agent is not causing or participating in formation of collision sites. The algorithm, however, will determine an expanding spiral path as a temporary route when the agent is still located in the collision site, in terms of preventing potential collisions among moving agents in our two dimensional space. Robots follow those temporary routes until exiting from those collision sites successfully. By analogy, this is a virtual, pivoting dance step between two agents in the collision sites. State diagram in Figure 3 demonstrates general strategies that we used in our route finder algorithm solution.

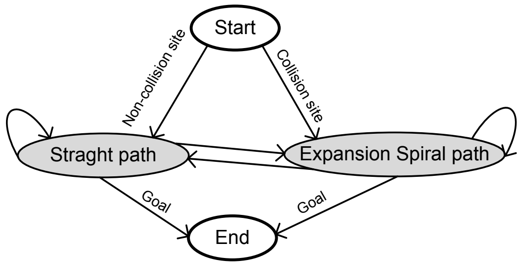

Fig 3. State graph of our path finder solution

Our path finder algorithm including collision avoidance solution inside is shown in Algorithm 1.

Algorithm 1. Path finder algorithm

---

1. Initialize your current location.
2. Initialize your assigned goal location.
3. Recognize and analyze around yourself by using your vision sensor to determine your current status.
4. If distance from your goal is 0, then end. ($d_{a_i} = 0$).
5. If collision site detected, then jump to step 9.
6. Adjust your direction toward your goal.
7. Move toward goal for one scale.
8. Jump to step 3.
9. Form a group of all agents that are in your vision site, virtually.
10. Calculate distance to each agent from your current coordinates.
11. Form a circle virtually crossing between you and the nearest agent in your vision site.
12. Move toward expansion spiral path around nearest virtual circle, formed in previous step for one scale.
13. Jump to step 3.

We assumed two sets in our two dimension experimental space; $A = \{a_1, a_2, \dots, a_i\}$, and $G = \{g_1, g_2, \dots, g_i\}$, where A, and G are agents and goals set, respectively. Agents know their assigned goal initially, $\forall a_i \in A, \exists g_i \in G, (a_i, g_i) \in I$, where I indicates a one to one correspondence pairs between agent set and goal set. Initially, each agent knows its exact coordinates along with its assigned goal location at the beginning of experimental cycle. $T_0 = \{a_{1(a_{1(x,y)_1}, g_{1(x',y')_1})}, a_{2(a_{2(x,y)_2}, g_{2(x',y')_2})}, \dots, a_{i(a_{i(x,y)_i}, g_{i(x',y')_i})}\}$ Agents start moving toward their goals at random times. Each agent a_i, at the beginning of the process of movement; Δt_{a_i}, a_i evaluates the environment around by analysing data obtained from its vision sensor. This strategy helps them to determine their situation and hence adjust their path accordingly. For instance, a_i is able to analyze the surface of $vs_i = \int_0^{\theta_i} \frac{1}{2} r_i^2 \theta$, as its vision site, captured from its vision sensor at any arbitrary time during movement toward its goal. $\bigcap_{p=1}^{m} vs_p = \emptyset \implies cs = 0$, indicates that there is no agent participating to form any collision site and hence a_i concludes to move toward its goal through by $y = y_1 + [(y'_1 - y_1) / (x'_1 - x_1)] \cdot (x - x_1)$, where (x_1, y_1) and (x'_1, y'_1) denote the position of a sample agent and goal on two dimension space respectively. $\exists (a_i, a_j) \in A, (vs_i \cap vs_j) \neq \emptyset \implies cs \neq \emptyset$, however, indicates there is at least one collision site for a_i, and hence it should alter its path through a temporal expansion spiral route. Those agents are located in collision site; follow these temporary spiral routes, until exiting from those sites. During the process of moving out from collision sites, agents continue analysing their environment around to detect any new agents into their current collision sites and hence, form a new temporary path based on the agent that maintains the nearest distance from them, in order to adjust their route, as it expressed in the collision site section.

V. EXPERIMENTAL RESULTS

The presented path finder algorithm, explained in the previous section, was implemented and tested with an experimental scenario. In this section, we illustrate and examine our algorithm along with the relative results and analysis. In this experiment, our system is used to plan avoiding collision among a group of 6 agents moving toward their goals, in our two dimensional experiment space. They are not able to communicate with to one another during the experiment; however, they are all using the same strategy to find and correct their path into their goals. This scenario is formed by arranging agents and goals in space, randomly, with considering assigning the farthest possible goals in terms of distances, to agents. This type of distributing agents and goals on the experimental site, leads to increasing the possibility of facing agents to more into collision sites, and hence, using our collision avoidance strategy as much as it possible. Figure 4 shows the positions of agents and goals on our experimental space.

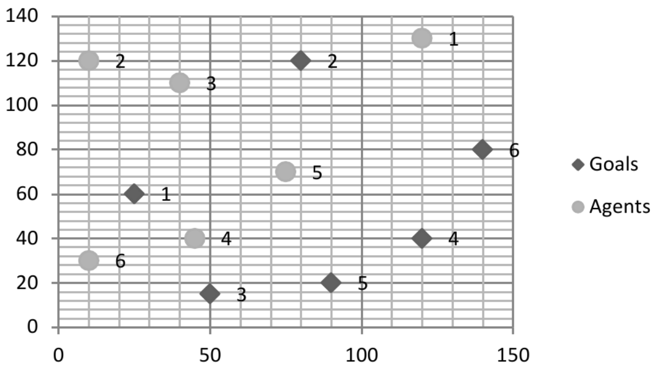

Fig 4. Agents and goals distribution on experimental site

Each agent is assigned a unique number in order to be recognized by other agents during the experiment. In addition, there is a one to one correspondence relationship between agents and goals. In other words, each agent in agent set is assigned the same number goal in goal set. During the cycle of experiment, we collected many key features such as, the times of start moving toward goals by agents, as well as the times of reaching goals, along with the total number of collision sites and virtual circles that each agent is face during the experiment, shown in the following Table 1.

TABLE 1.
RESULTS TABLE OF AGENTS MOVEMENT TOWARD THEIR GOALS

(a_i,g_i)	ΔTa_i	cs_i	vc_i	ΔTg_i
(1,1)	0	7	5	196
(2,2)	2	9	8	314
(3,3)	4	4	2	88
(4,4)	6	8	6	264
(5,5)	8	5	3	165
(6,6)	10	6	5	211

We also collected the times of entering and exiting collision sites for each agent, during the experiment, as shown in the Table 2.

Results depicted in Table 2, show significant differences for ΔT_g, for agents that encountered a larger number of collision sites. In other words, each collision site, based on its situation, and the total number of agents that participate in forming it, can potentially cause a significant delay for agents to reach their goals. This is because, agents that are located in collision sites, change their normal behaviour to

TABLE 2.
ENTRANCE AND EXIT TIMES FOR COLLISION SITES

a_i	$\Delta T\, ent_{cs_i}$	$\Delta T\, ext_{cs_i}$
1	{5, 14, 32, 48, 89, 102, 188}	{13, 30, 45, 80, 101, 183, 192}
2	{12, 29, 54, 88, 109, 164, 195, 248, 299}	{26, 50, 87, 103, 162, 193, 245, 296, 312}
3	{22, 56, 64, 80}	{54, 63, 79, 82}
4	{11, 130, 52, 83, 113, 142, 206, 252}	{129, 50, 81, 112, 140, 202, 251, 260}
5	{15, 43, 64, 78, 99, 125, 142, 161}	{40, 62, 77, 98, 120, 139, 155, 161}
6	{22, 45, 84, 99, 128, 197}	{42, 80, 98, 125, 192, 202}

choose their paths based on following the shortest possible routes, to a temporary paths which forms based on other agents participating to a same collision site, in order to handle, and hence, exiting from them. We observed no collisions among agents meaning they were able to reach their goals successfully, during the experiment.

VI. CONCLUSIONS

In this paper, we proposed a solution to prevent collision among a group of moving agents toward their specific goals. Our demonstrated algorithm is able to analyse the information gathered by equipped vision sensors, in order to decide the best possible route, in terms of safety and collision avoidance during the time of attempting to reach to their goals. We assumed our agents are not able to communicate and hence do not share details of their environment among one another. Our solution, thus, is able to help our agents to decide and routing toward their goals independently.

Our approach is able to control collision among moving agents into their goals successfully, however, using it, causes agents to have a significant delay before reaching their goals. These delays, depend of the total number of collision sites that each agent involves during the time of pursuing goals can substantially increase the cost of time. Future works includes optimizing our solution, in terms of minimizing the cost of time needed for agents to handle and exit collision sites.

REFERENCES

[1] H. Asama K. Ozaki, H. Itakura, A. Matsumoto, Y. Ishida, I. Endo,"Collision avoidance among multiple mobile robots based on rules and communication," *Intelligent Robots and Systems '91.* 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on, vol., no., pp.1215-1220 vol.3, 3-5, 1991.

[2] C. M. Clark, S. M. Rock and J. C. Latombe, "Motion planning for mobile robots using dynamic networks", *Proc IEEE Int Conf on Robotics and Automation,* 2003.

[3] F. Expert, S. Viollet and F. Ruffier, "Outdoor field performances of insect-based visual motion sensors", *Journal of Field Robotics,* vol. 28, no. 4, pp. 529–541, 2011.

[4] J. Gaspar, N. Winters and J. Santos-Victor, "Vision-based Navigation and Environmental Representations with an Omni-directional Camera", *IEEE Transactions on Robotics and Automation,* Volume 16 Number 6, pp 890 -898, 2000.

[5] Y. Guo and L. Parker, "A distributed and optimal motion planning approach for multiple mobile robots", *Proc IEEE IntConf on Robotics and Automation,* pp. 2612–2619, 2002.

[6] P. Kamkarian and H. Hexmoor, "A Collision Control Strategy for Multiple Moving Robots", *In Advances in Intelligent Systems and Computing,* Volume 208, pp. 863-871, Springer, 2012.

[7] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning”, *International Journal of Robotics Research,* Volume 30 Issue 7, pp. 846-894, 2011.

[8] R. Kumar and A. Menon. "Collision Avoidance in a Multi-Robot System by Emulating Human Behavior." *International Conference on Information Systems Analysis and Synthesis and World Multiconference on Systemics,* Cybernetics and Informatics, 2001.

[9] L. Marcolino and L. Chaimowicz, “Traffic control for a Swarm of Robots: Avoiding Group Conflicts”, *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2009.

[10] F. Mazzini, D. Kettler, J. Guerrero and S. Dubowsky, “Tactile Robotic Mapping of Unknown Surfaces”, *With Application to Oil Wells, IEEE Transactions on Instrumentation and Measurement,* vol. 60, pp. 420-429, 2011.

[11] P. Srivastava, S. Satish and P. Mitra, “A distributed fuzzy logic based n-body collision avoidance system”, *Proc of the 4th Int Symposium on Intelligent Robotic Systems,* Bangalore, pp. 166–172, 1998.

[12] L. Zhang, T. Zhang, H. Wu, A. Borst and K. K¨uhnlenz, “Visual Flight Control of a Quad rotor Using Bio inspired Motion Detector”, *International Journal of Navigation and Observation,* Volume 2012, 9 pages, Hindawi pub, 2012.