

Inert and mobile agents navigation interaction using reciprocal velocity obstacles for collisions avoidance

Susi Juniastuti¹, Moch Fachri¹, Supeno Mardi Susiki Nugroho², Mochamad Hariadi²

¹Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

²Department of Computer Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Article Info

Article history:

Received Nov 19, 2021

Revised Feb 26, 2022

Accepted Mar 15, 2022

Keywords:

Autonomous agents

Collision avoidance

Inert agents

Mobile agents

Reciprocal velocity obstacles

ABSTRACT

Reciprocal velocity obstacles (RVO) is a method used for multiagents navigation that enables collision and oscillation-free avoidance against other mobile agents. Despite its ability in collision avoidance between agents, RVO has a hard time dealing with static obstacle avoidance. This problem has led to a tendency to use RVO only for agents avoidance and use other methods to handle static obstacles avoidance. In this paper, we present our new approach for interaction between mobile agents against static obstacles in the RVO based collision avoidance. We propose a concept called inert agents that interact as static obstacles. This inert agent is stand firm as static obstacles should be, while the inert agent also able to satisfy reactive collision avoidance nature of RVO to produce better avoidance result. We conduct an experiment to compare the performance of avoidance in a certain scenario. Our method shows better results when compared with generic static obstacles.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Susi Juniastuti

Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember

Raya ITS Street, Surabaya, East Java, Indonesia

Email: susi@te.its.ac.id

1. INTRODUCTION

Since the original velocity obstacles (VO) was invented [1], countless variants of this method have emerged to meet the various needs of multi-agent navigation. The velocity obstacles method was originally designed to deal with obstacles avoidance in dynamic motion planning. Existing method for motion planning are categorized into either global or local planning [2]-[10]. Local planning is computationally efficient since it takes information through agent local data while navigating [11]-[18]. Whereas global planning requires precalculation of existing navigation plane in some form of pathfinding that is computationally expensive for the dynamic environment, especially where many obstacles and agent's position constantly changing [2]-[4], [19]-[27]. Global planning is possible as long as the information regarding navigation space is available. The example is shown in a study about crowd simulation where evacuation path is set [28], and locally plan avoidance motion using agent interact force based on social force model. Whereas in unknown environment cases, the local planning role is escalated even more since information regarding the navigation space can only be obtained locally [29]. VO is a form of local motion planning for autonomous agents that is able to direct the agent to navigate safely. VO itself works by predicting the agent's velocity that will result in a collision in the future through information gathered by sensors or other mean [1].

Through many years of development, the extension of velocity obstacles method make several improvements in many areas. Some example to mention is finite-time velocity obstacle (FVO) introduce time

constraint in avoidance process [30]. It also optimizes the process of choosing new velocity to minimize the avoidance velocity with the best velocity for agents to reach their objective. Another extension such as goal velocity obstacle (GVO) introduces goal region into velocity obstacles formulation [31]. GVO incorporates goal region which is the region of preferred velocity that will direct the agents toward its navigation objective. Inverse velocity obstacles (IVO) introduce ego-centric based framework to VO [32]. IVO assumes that each agent is the center of avoidance process at ego-frame, thus assuming that the robot is stationary at the origin point. At those condition, relative velocity outside the collision cone of VO is chosen based on ego-centric observation of the obstacle by the agent at two consecutive time instance.

To dealing with uncertainty of obstacles motion, several development were introduced such as (non linear velocity obstacles (NLVO) [33], probabilistic velocity obstacles (PVO) [34], IVO [35], and probabilistic inverse velocity obstacle (PIVO) [36]. NLVO takes into consideration the nonlinear motion of the obstacles, it also introduces the notion of risk that assigns a value that represents the risk factor to each velocity of the agents. The velocity that would produce immediate collision may have higher values of risk than some other velocity inside VO but not immediately produce a collision. PVO further extends the VO formula for uncertain estimation that may happen in sensors data. PVO assume that the data from mobile agents in real-life scenario (e.g mobile robot) are taken from sensors with a certain limit in their capability to fully grasp the necessary perception of agents against it's surrounding. This is due to the fact that noise in estimation may exist. This creates a probabilistic framework to bridge the agent's perception and navigation in reactive avoidance cases. IVO is designed with motion uncertainty of the obstacles in mind, by introducing an optimization objective function to improve avoidance decisions. It's split the avoidance process into two major processes, obstacles information processing for risk assessment and avoidance decision process. The optimization objective function of IVO consists of the risk of speed, the target speed deviation, and the collision time. PIVO is based on IVO combined with PVO that notices uncertainty in estimation due to noise occurred.

One classical problem with VO is oscillatory motion in agents avoidance. Reciprocal velocity obstacles (RVO) is designed to deal with this oscillation that occurred during the avoidance process between agents [37]. These were designed for dealing with local reactive collision avoidance, further implementation of this method for computer games shows that RVO had hard times dealing with static obstacles [38]. In this paper, we proposed a new approach in dealing with static obstacles in local motion planning. We utilize RVO for multi-agent navigation. Our approach is to exploit the reactive nature in RVO avoidance by introducing a new concept called the inert agent that will be used to substitute the static obstacles. This concept enables multi-agent navigation with all objects in navigation space are treated as agents.

2. METHOD

In RVO, each agent is expected to navigates independently [37]. RVO takes into account that the avoidance process between agents is a reactive process, the other agents are expected to make the same reasoning in the collision avoidance process. The creator of this method designed RVO as the extension of the velocity obstacles concept, to prevent oscillatory motion caused by reactive avoidance. Avoidance velocity in RVO is calculated by using the average value of velocity that lies outside the avoided agent's velocity obstacle with agent current velocity. RVO is defined as in (1), which shows the RVO of agent B to agent A. It contains every velocity of agent A inside the velocity obstacles in $VO_B^A(\mathbf{v}_B)$ averaged with the current velocity \mathbf{v}_A . Which the $VO_B^A(\mathbf{v}_B)$ itself is velocity obstacle of agents B to agent A. It contains every possible velocity that if taken, will cause agent A to collide with B at a certain point in the future.

$$RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) = \{\mathbf{v}'_A \mid 2\mathbf{v}'_A - \mathbf{v}_A \in VO_B^A(\mathbf{v}_B)\} \quad (1)$$

RVO_B^A = reciprocal velocity obstacles of agent B to agent A;

VO_B^A = velocity obstacles of agent B to agent A;

\mathbf{v}_A = current velocity of agent A;

\mathbf{v}_B = current velocity of agent B;

\mathbf{u} = velocity addition for next velocity.

RVO enabled agent's navigation that is free from collision and oscillation when handling avoidance between agents. However, in an investigation, RVO has hard times dealing with static obstacles [38]. As stated

by J snape in an article regarding RVO integration in games: "maybe it's a little counter-intuitive, but dealing with a static line obstacle is harder than dealing with a dynamic obstacle!" [38]. The solution he provided was to avoid using RVO for static obstacles avoidance.

If we refer to the origin of RVO [37], it was proposed as a concept for local reactive collision avoidance between agents, so it's intuitively the good solution. Nevertheless, we have different perspectives into dealing with these problems, whereas the agents are the one that produces motion in order to navigate through world space, it's crystally clear that the problems lies on the static obstacles. In (2) shows oscillation-free theorem of RVO.

$$\begin{aligned} \mathbf{v}_A &\in RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) \\ &\Leftrightarrow \\ \mathbf{v}_A &\in RVO_B^A(\mathbf{v}_B - \mathbf{u}, \mathbf{v}_A + \mathbf{u}) \end{aligned} \quad (2)$$

In the original explanation [37], the author explains that the newly chosen velocity will invalidate the old velocity. Based on the translation invariance shown in (3) and RVO of agent B to agent A in (1), the new velocity chosen for agent A which is $\mathbf{v}_A + \mathbf{u}$, is outside of RVO_B^A . However, the $\mathbf{v}_A + \mathbf{u}$ are the best velocity to take since it's the closest one to the reference velocities. This take account some assumption that agent B make similar calculation in avoidance process, hence the reaction of the opposite agent is expected($\mathbf{v}_B - \mathbf{u}$) thus in (2) is fullfilled. Since static obstacles bear no awareness in velocity space this cause problem in reactive collision avoidance case.

$$\mathbf{v}_A \in VO_B^A(\mathbf{v}_B) \Leftrightarrow \mathbf{v}_A + \mathbf{u} \in VO_B^A(\mathbf{v}_B + \mathbf{u}) \quad (3)$$

As we previously mention, the reaction of all agents involved in the avoidance process is expected. Generic static obstacles involved have no awareness at all since that's the natural way of obstacles, a hindrance in the navigation plane. We propose an approach to treat all objects in the navigation plane as agents. We define autonomous agents in our approach into two major categories, mobile agents and inert agents. Mobile agents are a group of agents that has the ability to move and navigate freely in a navigation plane. Whereas inert agents are a group of agents with no ability to move and navigate freely in a navigation plane. This inert agents act like static obstacles which is the only ability it has is none other than obstructing the way of mobile agents. Inert agents can only choose zero as possible speed thus can only react with zero velocity in the avoidance process.

From RVO explanation in the previous section, we explain the concept with the assumption that all agent involved has same effort in collision avoidance. this idea although true is not a limitation. We can assign priorities value among agents involved that resulted in differences in avoidance effort [37]. So we assigned those priority values in binary to represent mobile and inert agents, with the mobile agent as zero and the inert agent as one. Avoidance taken by the agent with the same priority value will take a mutual effort in avoiding each other (in our case, it is only occur between mobile agents since the inert agent has no ability to move). In the case of mobile agents avoidance against inert agents, the mobile agents take all the effort to avoid inert agents since the inert agents can only react by simply idle. Figures 1 to 3 shows avoidance between agents. Figure 1 shows us that both agents with an equal level of priority take a mutual endeavor in avoiding each other. Whereas Figure 2 shows that only an agent with a lower level of priority (agent colored in yellow) takes the avoidance manner when both mobile agents were on the opposite trajectory. Figure 3 is an example of an interaction between a mobile agent (agent colored in yellow) when faced with an inert agent (agent colored in blue). The inert agent holds still as a static obstacle should, so avoidance is only executed by mobile agents.



Figure 1. Avoidance from both agents with equal level of priority

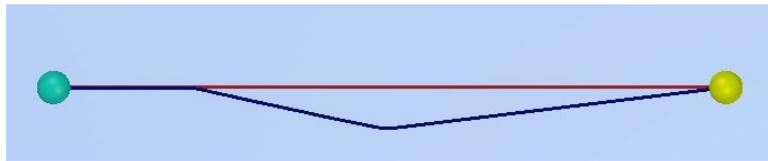


Figure 2. Avoidance from both agents with different level of priority

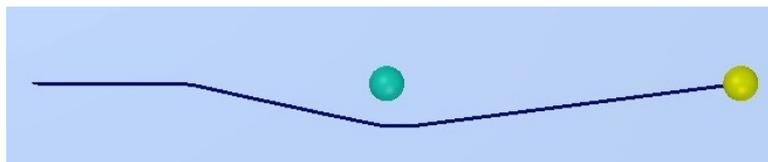


Figure 3. Avoidance between mobile agent against an inert agent

3. RESULT AND DISCUSSION

We create an experiment scenario to test our method. Figure 4 shows the environment of the navigation space. The area labeled A to D is the starting point for groups of mobile agents, whereas the static obstacles are scattered in the middle of the navigation plane. Group A will navigate toward area D, while group D will navigate toward area A. Group B will navigate toward area C, while group C will navigate toward area B. This scenario will create dense conditions in the middle of the navigation plane of simulation since all groups will intersect in the middle area.

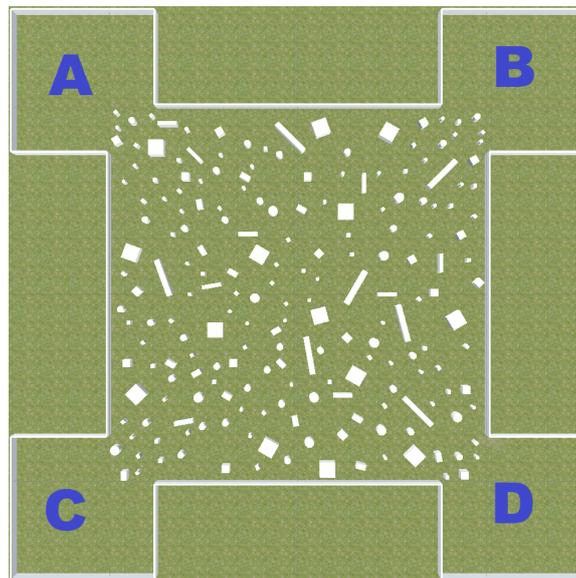


Figure 4. Navigation space environment

We conducted this scenario from total agents from 100 to 1,000 with three different methods in static obstacles avoidance. The first method is generic static obstacles avoidance with RVO. The second one is our proposed method with static obstacles as inert agents. Since the static obstacles present in simulation varying in shape, we use medial axis transform approximation uses for medial axis transform reciprocal velocity obstacle (MATRVO) [39]. The last one is the scenario with global pathfinding involved to direct the agents in navigation plane in order to reach their destination. In this scenario, the mobile agents will traverse the corridors that formed due to the presence of static obstacles using A* pathfinding, whereas avoiding other mobile agents using the RVO method. This scenario basically uses a path generated by a pathfinding algorithm to avoid the

static obstacles since the static obstacles count as impassable planes. The previous two scenarios only give the agents general direction to their destination, thus they rely on local collision avoidance to avoid the static obstacles. Figure 5 shows the sample run of our scenario with 500 mobile agents involved.

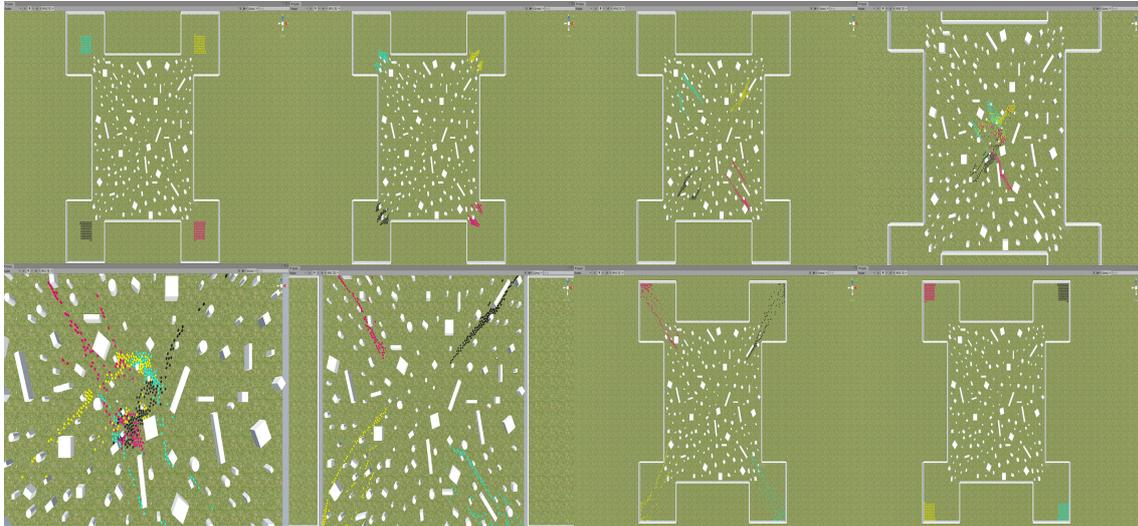


Figure 5. 500 mobile agents moving through scattered inert agents as static obstacles

Figure 6 shows us the average consumed time for all agents to arrive at their respective destinations. As the number of agents increases, so does the time required for every agent to arrive. As those graphics are shown to us, our method on average enables faster navigation compared to mobile agents with A* pathfinding by an average of 15.18%, however, it's slower compared with generic static obstacles by an average of 3.68%.

The collision that occurred from mobile agents against static obstacles shows in Figure 7. It gives us a sign about the effectiveness of our method to treat static obstacles as inert agents, as it produces less collision than generic static obstacles. However, the usual approach to not using RVO to avoid static obstacles is still the best approach, as shown in the chart in Figure 7, using A* pathfinding to make mobile agents traverse the formed corridors from static obstacles is proven to produce better results based on collision it's produced. One thing to be noted though, using a pathfinding algorithm such as A* requires time for the mobile agents to precalculate the path, whereas our method is simply giving the mobile agents general direction for each respective destination.

The number of collisions overall is dominated by mobile agents collisions against each other as shown in Figures 8 and 9. More detailed result presented in Table 1. Figure 10 shows the time required for pathfinding calculation using the A* algorithm. At 1,000 agents; 102.57 seconds is required to calculate the path for all agents, whereas our method only required 2.46 seconds to iteratively assign general direction toward their destination to all agents.

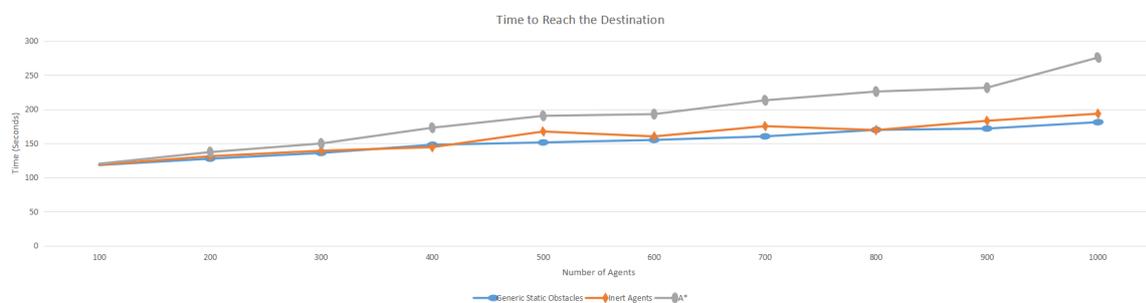


Figure 6. Graph of time to reach the destination

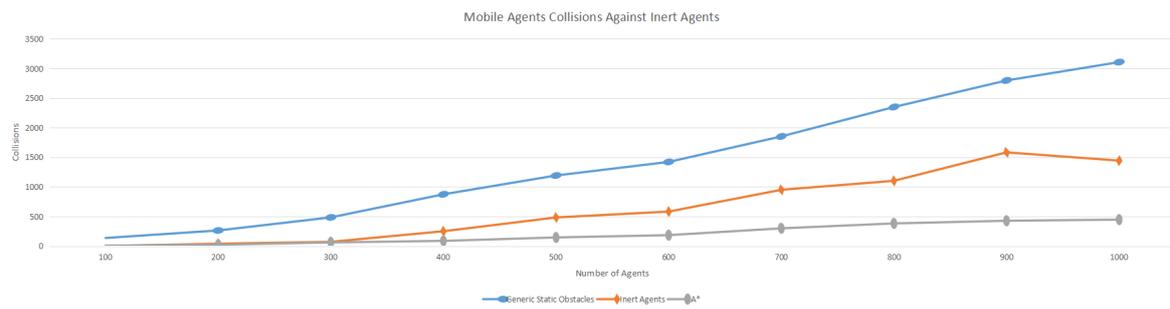


Figure 7. Graphs of mobile agents collisions against inert agents

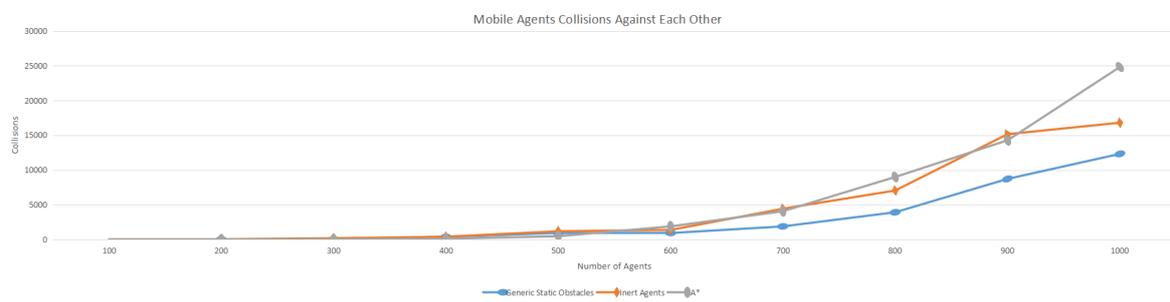


Figure 8. Graph of mobile agents collisions against each other

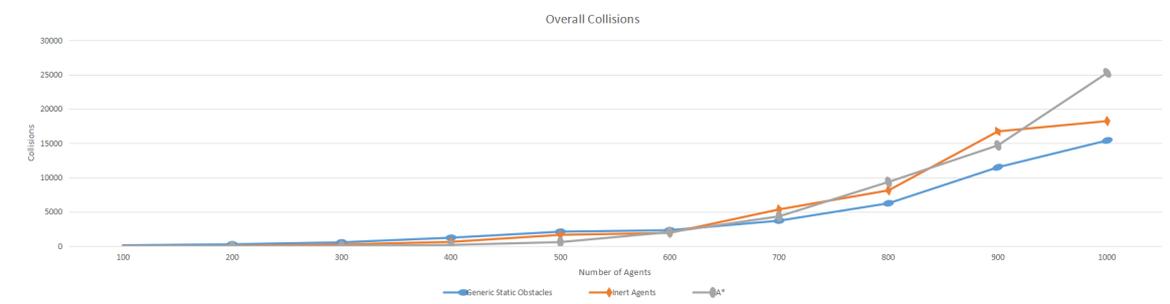


Figure 9. Graph of overall collisions of every agents

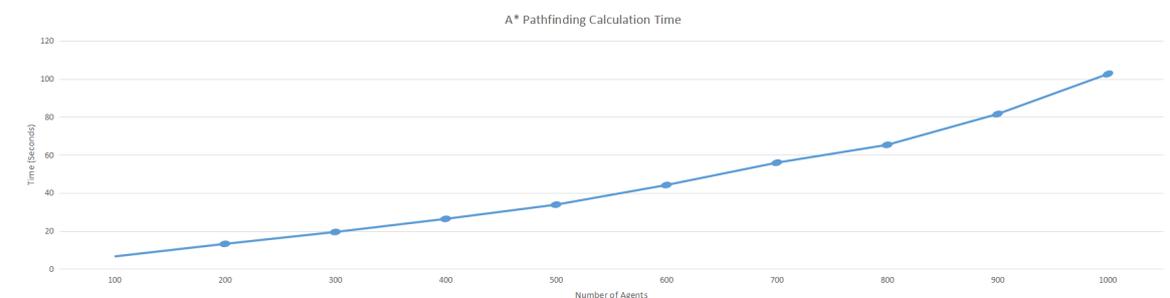


Figure 10. Graph of A* pathfinding calculation time

Table 1. The detailed result of the experiment

Number of agents	Total collision			Mobile agents collisions Against static obstacles			Mobile agents collisions against each other			Average time to reach destination (seconds)		
	Generic static obstacles	Inert agents	A*	Generic static obstacles	Inert agents	A*	Generic static obstacles	Inert agents	A*	Generic static obstacles	Inert agents	A*
100	136.00	0.00	7.00	136.00	0.00	7.00	0.00	0.00	0.00	118.19	118.61	120.15
200	276.00	68.00	22.00	263.00	40.00	22.00	13.00	28.00	0.00	127.55	131.20	137.23
300	564.00	246.00	64.00	484.00	71.00	61.00	80.00	175.00	3.00	135.95	139.25	149.81
400	1238.00	634.00	192.00	874.00	250.00	87.00	364.00	384.00	105.00	147.93	144.35	172.87
500	2119.00	1655.00	599.00	1192.00	483.00	145.00	927.00	1172.00	454.00	151.34	167.34	190.34
600	2333.00	1941.00	2052.00	1421.00	582.00	184.00	912.00	1359.00	1868.00	155.03	160.17	192.81
700	3721.00	5366.00	4349.00	1852.00	950.00	299.00	1869.00	4416.00	4050.00	160.22	175.19	213.10
800	6255.00	8136.00	9366.00	2349.00	1102.00	382.00	3906.00	7034.00	8984.00	169.80	169.47	225.91
900	11504.00	16730.00	14690.00	2798.00	1583.00	427.00	8706.00	15147.00	14263.00	171.65	183.09	231.53
1000	15413.00	18237.00	25259.00	3108.00	1443.00	446.00	12305.00	16794.00	24813.00	181.04	193.44	275.72

4. CONCLUSION

Our inert agent's concept was able to improve upon the static obstacles problems in RVO. Using the concept of inert agents to treat as static obstacles caused mobile agents collision to reduce by 65.99%. However, the idea of avoiding RVO to deal with static obstacles is still the best thing to do as it's shown by our experiment. Using the A* algorithm calculates the path for the mobile agents to take in order to traverse the formed corridors from static obstacles caused the collision to reduce collision by 87.68%. Whereas it's caused time to reach destination increase by an average of 18.23% compared to generic static obstacles. Our inert agent's method increases the time to reach their destination by an average of 3.68% when compared with generic static obstacles. Using A* pathfinding to traverse the corridor in the navigation plane is advised if the information of static obstacles can be preloaded to be calculated beforehand, thus only avoidance between mobile agents that are done using RVO. Such cases occurred in a controlled environment like a virtual environment in a game, or simulation, inside an autonomous factory, and any indoor environment with preplaced markers. Whereas our approach is better suited for the environment with no prior information regarding the presence of obstacles.

REFERENCES

- [1] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, Jul. 1998, doi: 10.1177/027836499801700706.
- [2] J. Barraquand and J.-C. Latombe, "Robot motion planning: a distributed representation approach," *The International Journal of Robotics Research*, vol. 10, no. 6, pp. 628–649, 1991, doi: 10.1177/027836499101000604.
- [3] C. Cheng, Q. Sha, B. He, and G. Li, "Path planning and obstacle avoidance for AUV: a review," *Ocean Engineering*, vol. 235, Sep. 2021, doi: 10.1016/j.oceaneng.2021.109355.
- [4] Z. Chen, Y. Zhang, Y. Zhang, Y. Nie, J. Tang, and S. Zhu, "A hybrid path planning algorithm for unmanned surface vehicles in complex environment with dynamic obstacles," *IEEE Access*, vol. 7, pp. 126439–126449, 2019, doi: 10.1109/ACCESS.2019.2936689.
- [5] N. Wang and H. Xu, "Dynamics-constrained global-local hybrid path planning of an autonomous surface vehicle," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 6928–6942, Jul. 2020, doi: 10.1109/TVT.2020.2991220.
- [6] B. Shi, Y. Su, C. Wang, L. Wan, and Y. Luo, "Study on intelligent collision avoidance and recovery path planning system for the waterjet-propelled unmanned surface vehicle," *Ocean Engineering*, vol. 182, pp. 489–498, Jun. 2019, doi: 10.1016/j.oceaneng.2019.04.076.
- [7] A. Ahmed, A. Maged, A. Soliman, H. El-Hussieny, and M. Magdy, "Space deformation based path planning for mobile robots," *ISA Transactions*, Aug. 2021, doi: 10.1016/j.isatra.2021.08.019.
- [8] D. Lyu, Z. Chen, Z. Cai, and S. Piao, "Robot path planning by leveraging the graph-encoded Floyd algorithm," *Future Generation Computer Systems*, vol. 122, pp. 204–208, Sep. 2021, doi: 10.1016/j.future.2021.03.007.
- [9] Y. Tang, Y. Miao, A. Barnawi, B. Alzahrani, R. Alotaibi, and K. Hwang, "A joint global and local path planning optimization for UAV task scheduling towards crowd air monitoring," *Computer Networks*, vol. 193, p. 107, Jul. 2021, doi: 10.1016/j.comnet.2021.107913.
- [10] Y. Hong, S. Kim, and J. Cha, "Integrated global and local path planning for quadrotor using particle swarm optimization," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15621–15615, 2020, doi: 10.1016/j.ifacol.2020.12.2497.
- [11] B. B. K. Ayawli, A. Y. Appiah, I. K. Nti, F. Kyeremeh, and E. I. Ayawli, "Path planning for mobile robots using morphological dilation voronoi diagram roadmap algorithm," *Scientific African*, vol. 12, Jul. 2021, doi: 10.1016/j.sciaf.2021.e00745.
- [12] H. Sang, Y. You, X. Sun, Y. Zhou, and F. Liu, "The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations," *Ocean Engineering*, vol. 223, Mar. 2021, doi: 10.1016/j.oceaneng.2021.108709.
- [13] Z. Wang, G. Li, and J. Ren, "Dynamic path planning for unmanned surface vehicle in complex offshore areas based on hybrid algorithm," *Computer Communications*, vol. 166, pp. 49–56, Jan. 2021, doi: 10.1016/j.comcom.2020.11.012.
- [14] Y. Shin and E. Kim, "Hybrid path planning using positioning risk and artificial potential fields," *Aerospace Science and Technology*, vol. 112, May 2021, doi: 10.1016/j.ast.2021.106640.

- [15] H. Niu, C. Ma, and P. Han, "Directional optimal reciprocal collision avoidance," *Robotics and Autonomous Systems*, vol. 136, Feb. 2021, doi: 10.1016/j.robot.2020.103705.
- [16] X. Zhang, C. Wang, L. Jiang, L. An, and R. Yang, "Collision-avoidance navigation systems for maritime autonomous surface ships: a state of the art survey," *Ocean Engineering*, vol. 235, Sep. 2021, doi: 10.1016/j.oceaneng.2021.109380.
- [17] L. Li, D. Wu, Y. Huang, and Z.-M. Yuan, "A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field," *Applied Ocean Research*, vol. 113, Aug. 2021, doi: 10.1016/j.apor.2021.102759.
- [18] H. Simas and R. Di Gregorio, "Smooth transition for collision avoidance of redundant robots: An on-line polynomial approach," *Robotics and Computer-Integrated Manufacturing*, vol. 72, Dec. 2021, doi: 10.1016/j.rcim.2020.102087.
- [19] X. Guo, M. Ji, Z. Zhao, D. Wen, and W. Zhang, "Global path planning and multi-objective path control for unmanned surface vehicle based on modified particle swarm optimization (PSO) algorithm," *Ocean Engineering*, vol. 216, Nov. 2020, doi: 10.1016/j.oceaneng.2020.107693.
- [20] P. Chen, Y. Huang, E. Papadimitriou, J. Mou, and P. van Gelder, "Global path planning for autonomous ship: a hybrid approach of fast marching square and velocity obstacles methods," *Ocean Engineering*, vol. 214, Oct. 2020, doi: 10.1016/j.oceaneng.2020.107793.
- [21] S. Han, L. Wang, Y. Wang, and H. He, "An efficient motion planning based on grid map: predicted trajectory approach with global path guiding," *Ocean Engineering*, vol. 238, Oct. 2021, doi: 10.1016/j.oceaneng.2021.109696.
- [22] C. D. B. Borges, A. M. A. Almeida, I. C. Paula J'uniior, and J. J. d. M. S'a Junior, "A strategy and evaluation method for ground global path planning based on aerial images," *Expert Systems with Applications*, vol. 137, pp. 232–252, Dec. 2019, doi: 10.1016/j.eswa.2019.06.067.
- [23] T. T. Mac, C. Copot, D. T. Tran, and R. D. Keyser, "A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization," *Applied Soft Computing*, vol. 59, pp. 68–76, Oct. 2017, doi: 10.1016/j.asoc.2017.05.012.
- [24] G. Xie, X. Du, S. Li, J. Yang, X. Hei, and T. Wen, "An efficient and global interactive optimization methodology for path planning with multiple routing constraints," *ISA Transactions*, Mar. 2021, doi: 10.1016/j.isatra.2021.03.041.
- [25] P. Chand and D. A. Carnegie, "A two-tiered global path planning strategy for limited memory mobile robots," *Robotics and Autonomous Systems*, vol. 60, no. 2, pp. 309–321, Feb. 2012, doi: 10.1016/j.robot.2011.11.009.
- [26] K. Yu *et al.*, "USV path planning method with velocity variation and global optimisation based on AIS service platform," *Ocean Engineering*, vol. 236, Sep. 2021, doi: 10.1016/j.oceaneng.2021.109560.
- [27] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Computer Communications*, vol. 149, pp. 270–299, Jan. 2020, doi: 10.1016/j.comcom.2019.10.014.
- [28] M. N. Jamaludin, S. Mohamad, M. S. Sunar, K. Isa, R. M. Hanifa, F. M. Nasir, and S. M. Shah, "OpenGL 3D crowd evacuation simulation at universiti tun hussein onn malaysia (UTHM) hostel," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 16, no. 2, pp. 1034–1041, Nov. 2019, doi: 10.11591/ijeecs.v16.i2.pp1034-1041.
- [29] S. H. Abdulredah and D. J. Kadhim, "Developing a real time navigation for the mobile robots at unknown environments," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 20, no. 1, pp. 500–509, Oct. 2020, doi: 10.11591/ijeecs.v20.i1.pp500-509.
- [30] S. Samavati, M. Zarei, and M. T. Masouleh, "An optimal motion planning and obstacle avoidance algorithm based on the finite time velocity obstacle approach," In *2017 Artificial Intelligence and Signal Processing Conference (AISP)*, Oct. 2017, pp. 250–255, doi: 10.1109/AISP.2017.8324091.
- [31] J. Snape and D. Manocha, "Goal velocity obstacles for spatial navigation of multiple virtual agents," In *Proceedings of the 2013 international conference on autonomous agents and multi-agent systems*, 2013, pp. 1191–1192, doi: 10.5555/2484920.2485137.
- [32] P. S. N. Jyotish, Y. Goel, A. V. S. S. B. Kumar, and K. M. Krishna, "Ivo: inverse velocity obstacles for real time navigation," In *Proceedings of the Advances in Robotics 2019*, 2019, doi: 10.1145/3352593.3352610.
- [33] F. Large, S. Sekhavat, Z. Shiller, and C. Laugier, "Using non-linear velocity obstacles to plan motions in a dynamic environment," In *7th International Conference on Control, Automation, Robotics and Vision*, 2002, vol. 2, pp. 734–739, doi: 10.1109/ICARCV.2002.1238513.
- [34] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid," In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Apr. 2007, pp. 1610–1616, doi: 10.1109/ROBOT.2007.363554.
- [35] W. Zhang, S. Wei, Y. Teng, J. Zhang, X. Wang, and Z. Yan, "Dynamic obstacle avoidance for unmanned underwater vehicles based on an improved velocity obstacle method," *Sensors*, vol. 17, no. 12, Nov. 2017, doi: 10.3390/s17122742.
- [36] P. S. N. Jyotish, Y. Goel, A. V. S. S. B. Kumar, and K. M. Krishna, "PIVO: probabilistic inverse velocity obstacle for navigation under uncertainty," In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, Oct. 2019, pp. 1–6, doi: 10.1109/RO-MAN46459.2019.8956406.
- [37] J. V. D. Berg, Ming Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," In *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 1928–1935, doi: 10.1109/ROBOT.2008.4543489.
- [38] A. J. Champandard, *Velocity obstacle algorithms and velocity obstacle algorithms and locomotion integration*. (2012). [Online]. Available: <https://gamma.cs.unc.edu/RVO2/publications/AIGD12.pdf>.
- [39] Y. Ma, D. Manocha, and W. Wang, "Efficient reciprocal collision avoidance between heterogeneous agents using CTMAT," Apr. 2018, arXiv: 1804.02512.

BIOGRAPHIES OF AUTHORS

Susi Juniastuti     is received a master's degree in advanced applied science from Tokyo Institute of Technology, Japan in 2005. She obtained bachelor's degree in electronics engineering from Institut Teknologi Sepuluh Nopember, Indonesia in 1994. She is currently a doctoral student of the electrical engineering department of Institut Teknologi Sepuluh Nopember, Indonesia. Her research interest is in the fields of digital systems, signal processing, and intelligent system. She is a student member of IEEE. She can be contacted at email: susi@te.its.ac.id.



Moch Fachri     received both bachelor's and master's degrees in electrical engineering department of Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, in 2016 and 2018 respectively. He is currently a doctoral student of the electrical engineering department of Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. He is also a student member of IEEE. His research interest is in game technology, mobile agents navigation, crowd modeling, and intelligent system. He can be contacted at email: moch.fachri10@mhs.ee.its.ac.id.



Supeno Mardi Susiki Nugroho     received the bachelor's degree in electrical engineering from the Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, the master's degree from the department of informatics engineering, Institut Teknologi Bandung, and the Ph. D. degree in electrical engineering from the Institut Teknologi Sepuluh Nopember. He is currently the head of the computer engineering department, Institut Teknologi Sepuluh Nopember. His research interests include image processing, serious game, data mining, and intelligent systems. He is a member of IEEE. He can be contacted at email: mardi@its.ac.id.



Mochamad Hariadi     received the B. E. degree in electrical engineering department of Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, in 1995. He received both M. Sc. and Ph. D. degrees in graduate school of information science Tohoku University Japan, in 2003 and 2006 respectively. Currently, he is the staff of electrical engineering department of Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. His research interest is in video and image processing, data mining, big data, and intelligent system. He is the project leader in joint research with PREDICT JICA project Japan and WINDS project Japan. He is a member of IEEE and a member of IEICE. He can be contacted at email: mochar@te.its.ac.id.