



Dynamic Pathfinding for Non-Player Character Follower on Game

Paulus Harsadi^{1,*}, Siti Asmiatun², Astrid Novita Putri³

¹ Informatics Engineering, STMIK Sinar Nusantara, Jl. K.H Samanhudi No.84-86, Purwosari, Jawa Tengah 57149, Indonesia

^{2,3} Informatics Engineering, Universitas Semarang, Jl. Soekarno Hatta, RT.7/RW.7, Jawa Tengah 59160, Indonesia

Email: ¹paulusharsadi@sinus.ac.id*; ²siti.asmiatun@usm.ac.id; ³astrid@usm.ac.id

* corresponding author

ARTICLE INFO

ABSTRACT

Article history:

Received: Apr 26, 2021;
Revised: May 03, 2021;
Accepted: Jun 02, 2021;
Available online: Sep 30, 2021.

Keywords:

Artificial Intelligence;
Dynamic pathfinding;
Artificial Potential Field;
Flocking behavior;
Game;
Unity 3D.

Artificial Intelligences in video game are important things that can challenge game player. One of them is creating character or NPC Follower (Non-player character Follower) inside the video game, such as real human/animal attitude. Artificial Intelligences have some techniques in which pathfinding is one of Artificial Intelligence techniques that is more popular in research than other techniques. The ability to do dynamic pathfinding is Dynamic Particle Chain (DPC) algorithm. This algorithm has the ability of flocking behavior called boid to explore the environment. But, the algorithm method moves from one boid's point to another according to the nearest radius, then it will be able to increase computation time or needed time toward the target. To finish higher computation problem in dynamic pathfinding, the researcher suggests an algorithm that is able to handle dynamic pathfinding process through attractive potential field function of Artificial Potential Field to start pathfinding toward the target and flocking behavior technique to avoid the obstacle. Based on the test result by simulation of moving environment and complex, the computation time of algorithm is faster than comparison algorithms, DPC and Astar. It concludes that the suggested method can be used to decrease computation level in dynamic pathfinding.

© 2021 JTI C.I.T. All rights reserved.

1. Introduction

Artificial Intelligences (AI) in video game are important things that can challenge player. One of them is creating characters of video game, such as real human/animal attitude [1]. The problems raise in AI application on game covering learning, planning, natural language processing, and pathfinding. Pathfinding is the most interesting technique to research than other techniques because pathfinding is a basic technique that is often required and applied in a video game [2]. The application of pathfinding includes map analysis to find the best score in the movement from one point to other points. The best track is decided from the number of score and some criterias, such as the shortest track, the fewest track, and the most safe track [3].

Many researchs on finding method, especially pathfinding in video game, have been done [2]. Algorithm A* [4] or pathfinding A* and development variation of pathfinding A* have already attached in AI game. Moreover, those are mostly used in video game [5]. Surely, pathfinding A* and development variation are capable enough to solve the pathfinding problem but its environment still stays static. If it compares with dynamic environment, there may be moving obstacles while the algorithm is working so it will be getting stalled. One of solutions offered by video game developer is using real time pathfinding



method. It assumes dynamic environment being static everytime when the environment is moving so that it needs high computation time.

The approach of Real time pathfinding includes Time-bounded A* (TBA*) [6], Real-time Iterative-depending Best-first Search (RIBS) [7], and K Nearest Neighbor LRTA* (kNN LRTA*) [8]. For information, many researchers develop algorithm using dynamic pathfinding method which doesn't consider dynamic environment as static environment when the environment is changing. For examples, its researches are Real-time Pathfinding with Genetic Algorithm (RTA-GA) [9] combining between A* and Genetic algorithm and a research using heuristic method or algorithm based on evolutionary approaches, such as ACO (Ant Colony Optimization [10][11][12][13] and genetic algorithm (GA) [9][14][15][16]. Combine Dynamic Pathfinding (DPA) and A* Algorithm [17], Fuzzy logic and A* Algorithm [18] and a research using heuristic method as dynamic particle chain (DPC)[19]. Application of heuristic method ensures the discovery of best track, and until now A* is still being implemented[20].

This research tries to find the best solutions to do the pathfindings in dynamic environment with low computation. One of these solutions is DPC algorithm. DPC algorithm is created from flocking behavior called boid [21] where the movement of agent toward its target is used as point for the track. The point result of boid movement will be used by the agent to move with the rule of point election selecting boid in nearest radius with its target and unimpaired track. In addition, this method enables to open many tracks than using grid method to represent the environment. However, even though the process of finding tracks is fewer, only the area passed by boid, but DPC algorithm only chooses boid in radius area and then recomputation is held to choose the next boid or track node. Therefore, the computation time to reach the target is not maximal. It is because the rule process of DPC algorithm is similar enough with A* pathfinding which is breadth-first search algorithm [22]. This research assumes that dynamic pathfinding algorithm applies flocking behavior of boid as a method to avoid the obstacles. Meanwhile, to move toward the process, player uses attractive potential field function of Artificial Potential Field.

2. The Proposed Method

2.1. Flocking Behavior

One of the applied methods in this research is boid from flocking behavior to avoid the obstacles so the boid will be used by NPC Follower as a leap in order not to make collision with those obstacles. Flocking behavior is a flock in which its members having the ability to move and interact each other. Boid is one of flock attitude concepts proposed by Reynolds [20]. Flocking behavior has three main rules and two additional rules, that is [23] : Cohesion (Flocking); tendency of boid to stay close in flock local center from others. Separation (collision avoidance); tendency of boid to stay away from the other and avoid the probability of collision. Alignment (velocity matching); tendency of boid to compare its line and speed with others. Obstacle Repulsion or obstacle avoidance; rejection of obstacle where each boid will detect the position of an obstacle. Goal Attraction; boid's interest to the target in which its rule also works for the leader.

2.2. Artificial Potential Field

In this case, APF is applied by NPC follower to move toward the target by dividing it into some areas or radius. Artificial Potential Field (APF) is firstly applied in path planning robotics in 1986 by Khatib. His basic idea creates virtual style formulated in robotic area [24]. APF consists of two functions, Attractive potential field function and Repulsive potential field [25]. This Fig. 1 below is the model of APF proposed by khatib [26].

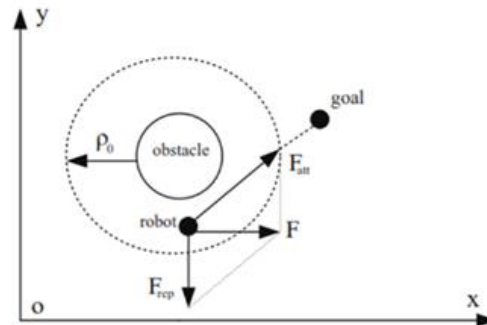


Fig. 1. Artificial Potential Field (APF) Model by Khatib [26]

APF Model by Khatib on Fig. 2 has the farthest effect than repulsive force in obstacle avoidance. F_{att} is target tensile strength on robot, F_{rep} is repulsive force on robot. F is Vector composition or resultant between the tensile strength and repulsive force.

2.3. Trigonometry

The detail process of NPC Follower in obstacle avoidance applies cosine-law in selecting process of boid as a leap in APF radius. Cosinus law is one of laws applied in triangle or trigonometry, it can be used to decide the right score of angle or the side length of a triangle [27].

2.4. Research Method

This section includes research process starting from design making either in environment or character and the proposed formula of dynamic pathfinding algorithm. Dynamic pathfinding simulation covering dynamic environment will be represent in a map in the video game. Environment of simulation consists of two obstacle scenarios having static obstacle and dynamic obstacle. Design of NPC follower and player will be using 3d soldier character holding a gun in an avatar form shown in Fig. 2.



Fig. 2. Soldier Avatar for NPC Follower and player.

2.5. Dynamic Environment Algorithm

This research assumes the flock with a number of boid variable N_b . On the certain process time, that is i , boid has $X_b(i)$ position with $k \in IN_b \equiv \{1, 2, \dots, N_b\}$. Non-player character follower is located on $X(i)$ position and target position, $PlayerX_g$. On the selected time, flock i is distributed by R_2 (distance to target). There is D_m distribution as dynamic obstacle and its position is $X_m(i)$ with $m \in IN_m \equiv \{1, 2, \dots, N_m\}$.

Domain D_o is a static obstacle and X_o is a position of static obstacle. Configuration room (CS) is R^2 area after it is reduced by domain D_o and D_m so boid and NPC Follower will be considered as moving point in CS. While the process continues, NPC Follower and boid will be considered as moving particles. Fig. 3 shows the movement process.

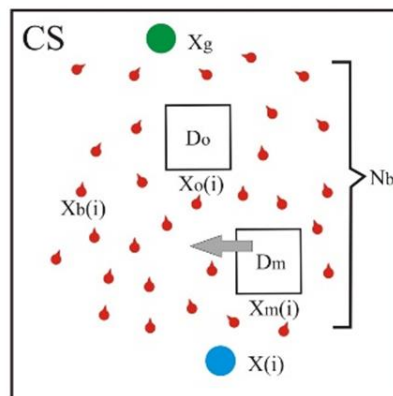


Fig. 3. Movement of Flocking Boid

Based on the three basic rules (alignment, separation and cohesion) and two additional rules (obstacle avoidance and goal attraction), point of pathfinding comes from movement of each boid to avoid the obstacles and get the target. The result of the point can be used by Artificial Potential Field to avoid obstacles in order to get the target dynamically.

2.6. Improved attractive potential field for pathfinding

APF method applied in this research is attractive potential field function. In this research, attractive potential field functions as tensile strength from the target, player to non-player character (NPC) follower. This tensile strength will be greater to pull NPC follower when the distance between NPC follower and player (target) gets further in the game shown in Fig. 4.

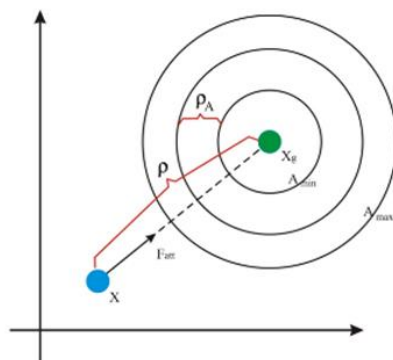


Fig. 4. Improved Attractive Potential Field Model

The function of attractive potential field adapted from the robot is defined on the explanation below [26] :

$$U_{att}(X) = \frac{1}{2} k_1 \rho^2(X, X_g) \tag{1}$$

Wherein :

- X = (x,y) is a vector position of NPC follower
- X_g = Player location

U_{att} = Score of Attractive potential field
 $\rho(X, X_g)$ = Distance between NPC follower and player
 k_1 = Constant of attractive potential field (positive score)

Attractive Force (F_{att}) is negative gradient of U_{att} . The formula is defined on the explanation below [26] :

$$F_{att} = -\nabla[U_{att}(X)] = k_1\rho(X, X_g) \quad (2)$$

Wherein :

F_{att} = Attractive Force
 $\rho(X, X_g)$ = Distance between NPC follower and player
 k_1 = Constant of attractive potential field (positive score)

2.7. Obstacle Avoidance

NPC follower doesn't apply repulsive style of Artificial potential field, but it applies position of boid flocking movement while the obstacle avoidance process continues statistically or dynamically. The notation in Fig. 3 shows that point B is position of NPC follower, point A is position of boid, and position C is player position. Cosinus rules to calculate - corner will be explained below [27] :

$$\cos \beta = \frac{a^2 + c^2 - b^2}{2ac} \quad (3)$$

Cosinus rules are applied to determine the angle β . Boid in the area will calculate the value of angle β based on position of NPC follower and player then it will be put in order based on the angle β . Fig. 5 shows the process of obstacle avoidance to the target.

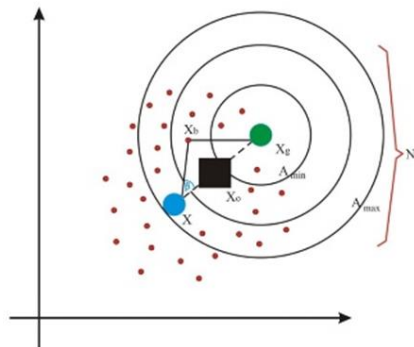


Fig. 5. The Calculation of angle β between Boid, NPC Follower, and Player

If angle β is bigger, boid position gets further with the obstacles so NPC follower can be the tentative target to avoid obstacle. Afterward, the attractive style can be applied.

3. Result and Discussion

The implementation used Unity 3D game engine. The dynamic environment was set based on the scenario using existing asset and material, such as terrain, obstacle wall, directional light, and camera. Size of environment was based on terrain resolution having 40 x 40 size and using a unit value from unity 3D game engine. Implementation of boid creates a game object resembling prefab of flat circle. The

whole rules and variables in boid will get score so the flocking behavior will be suit as expectation. Table 1 below shows the variables and rules in this research and its scores.

Table 1.
Scores of Boid Variable and Rule

Boid Rule and Variable	Score
Speed	3
Max Speed	5
Separation	6
Alignment	2
Cohesion	5
Force Obstacle Avoidance	25

Spread area of boid is restricted by Swarm radius so it will not move out from dynamic environment. Its wide and length scores are decided by swarm bounds. In this research, the swarm radius is 15 and the swarm bound is 35 x 35. Fig. 6 shows the implementation of flocking behavior by boids.

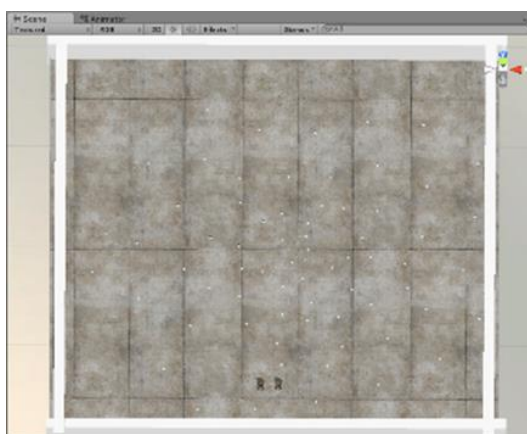


Fig. 6. Implementation of Flocking Behavior.

3.1 Implementation of Dynamic Pathfinding in NPC Follower

Dynamic pathfinding algorithm implemented in NPC follower doesn't only need script to calculate algorithm score, but it also needs other support variables. As an example, score of potential field radius is 30 and range of the area is 5, whereas the variable is implemented to the player. In this research, the implementation process of dynamic pathfinding algorithm is, (1) based on radius score and potential field range on script potential field that is implemented to the player, so boid and NPC follower will be classified by each area which is appropriate to its score range of APF force. (2) On the script Follow Behavior PF implemented in NPC follower, attractive potential field is calculated to move. If NPC follower finds obstacles while it moves to target (player), NPC follower will apply boid to the area and choose boid as track change to avoid the obstacles.

Based on the result of dynamic environment implementation, characters of proposed NPC follower and player, and dynamic pathfinding shows the simulation result of algorithm applied in NPC follower. Fig. 7 and 8 below show the implementation result of 2 .

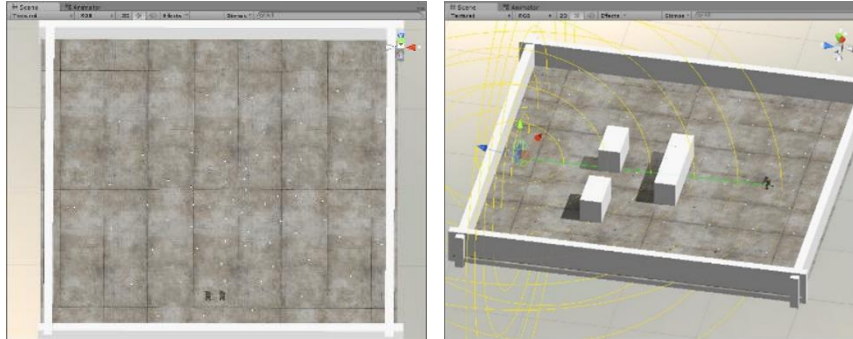


Fig. 7. Implementation of environment with moving obstacles

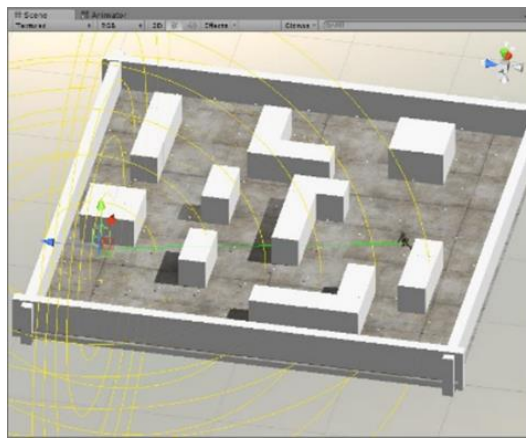


Fig. 8. Implementation of environment with complex obstacles.

3.2 Testing

Testing of proposed dynamic pathfinding (boid column PF) compares Dynamic Particle Chain algorithm (Boid column DPC) and A-star algorithm (Astar column). To get the accurate result, it needs to test both the environment with moving obstacles and the environment with complex obstacles. The testing needs to do 100 times to get mean of those three algorithms. The basic of Dynamic Particle Chain algorithm and dynamic pathfinding algorithm is flocking behavior of boid. Therefore, this stage includes in the testing if there are different numbers of boid, from 70 to 90. These tables below are the mean results of testing from 2 environments, moving environment in table 2 and complex environment in table 3 in second.

Table 2.
Result of Moving Environment Test

Astar	70 boid		90 boid	
	boid DPC	boid PF	boid DPC	boid PF
9.906526	7.63544436	6.98441418	7.5867491	6.83753357

Table 3.
Result of Complex Environment Test

Astar	70 boid		90 boid	
	boid DPC	boid PF	boid DPC	boid PF
10.25322	8.7916366	7.43820206	8.0455804	7.13606387

4. Conclusion

Based on the research of dynamic pathfinding, it concludes that : (a). The proposed method in this research can reduce computation time in problems of dynamic pathfinding. (b). The testing in simulation environment shows the faster result in dynamic pathfinding than Astar and DPC algorithms. (c). For next research, the implementation of attractive potential field function and flocking behavior of boid can be developed not only for two-dimensional environment but also three-dimensional environment. Therefore, it can be useful for not only NPC follower in walking character but also flying character as air plane in game. (d). The proposed algorithm can be developed for a flock of NPC followers so there will be not only one NPC follower doing the dynamic pathfinding.

References

- [1] I. Millington and J. Funge, *Artificial Intelligence for Games, Second Edition*. 2009.
- [2] D. N. Adi Botea, Bruno Bouzy, Michael Buro, Christian Bauckhage, "Pathfinding in Games." pp. 21–31, 2013.
- [3] P. Yap, "Grid-Based Path-Finding," 2002, pp. 44–55.
- [4] P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968.
- [5] P. Cowling *et al.*, "Search in Real-Time Video Games," *Artif. Comput. Intell. Games*, vol. 6, pp. 1–19, 2014.
- [6] Bjornsson Yngvi, V. Bulitko, and N. Sturtevant, "TBA*: Time-Bounded A*," *IJCAI*, pp. 431–436, 2009.
- [7] N. Sturtevant, V. Bulitko, and Y. Björnsson, "On learning in agent-centered search," *Proc. 9th Int. Conf. Autonomous Agents Multiagent Syst. (AAMAS 2010)*, pp. 333–340, 2010.
- [8] V. Bulitko and Y. Björnsson, "{kNN LRTA*}: Simple Subgoaling for Real-Time Search," *AIIDE 2009 - AI Interact. Digit. Entertain. Conf.*, pp. 2–7, 2009.
- [9] A. F. D. V Machado *et al.*, "Real time pathfinding with genetic algorithm," *Brazilian Symp. Games Digit. Entertain. SBGAMES*, pp. 215–221, 2011.
- [10] M. a. P. Garcia, O. Montiel, O. Castillo, R. Sepúlveda, and P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Appl. Soft Comput.*, vol. 9, no. 3, pp. 1102–1110, 2009.
- [11] S. Liu, L. Mao, and J. Yu, "Path Planning Based on Ant Colony Algorithm and Distributed Local Navigation for Multi-Robot Systems," *2006 Int. Conf. Mechatronics Autom.*, pp. 1733–1738, 2006.
- [12] M. Brand, M. Masuda, N. Wehner, and Xiao-Hua Yu, "Ant Colony Optimization algorithm for robot path planning," in *2010 International Conference On Computer Design and Applications*, 2010, vol. 3, no. Iccda, pp. V3-436-V3-440.
- [13] J.-W. Lee, J.-J. Kim, and J.-J. Lee, "Improved Ant Colony Optimization Algorithm by Potential Field Concept for Optimal Path Planning," *Humanoid Robot.*, pp. 662–667, 2008.
- [14] H. Qu, K. Xing, and T. Alexander, "An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots," *Neurocomputing*, vol. 120, pp. 509–517, 2013.
- [15] A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Comput. Electr. Eng.*, vol. 38, no. 6, pp. 1564–1572, 2012.
- [16] Z. Y. Z. Yao and L. M. L. Ma, "A Static Environment-Based Path Planning Method by Using Genetic Algorithm," *Comput. Control Ind. Eng. (CCIE), 2010 Int. Conf.*, vol. 2, pp. 405–407, 2010.
- [17] Y. Sasaki, A. Primanita, and M. Syahroyni. "Pathfinding car racing game using dynamic pathfinding algorithm and algorithm A*" ,in Proc. -ICWT 2017 3rd Int. Conf. Wirel. Telemat. 2017, pp.164–169, July 2018.
- [18] P. Harsani, I. Mulyana, and D. Zakaria. "Fuzzy logic and A* algorithm implementation on goat foraging games", IOP Conf. Ser. Mater. Sci. Eng., Vol. 332, No. 1, 2018.
- [19] J. Espelosín, L. Acosta, and D. Alonso, "Path planning approach based on flock dynamics of moving particles," *Appl. Soft Comput. J.*, vol. 13, no. 4, pp. 2159–2170, 2013.
- [20] Kapi, Azyan Yusra, Mohd Shahrizal Sunar, and Muhamad Najib Zamri. "A review on informed search algorithms for video games pathfinding." *International Journal* 9.3, 2020.
- [21] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, 1987.
- [22] N. R. Sturtevant, "Moving Path Planning Forward," pp. 1–6, 2012.
- [23] C. W. Reynolds, "Steering behaviors for autonomous characters," *Game Dev. Conf.*, pp. 763–782, 1999.
- [24] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, vol. 5, no. 1. pp. 90–98, 1986.
- [25] L. Zhou and W. Li, "Adaptive Artificial Potential Field Approach for Obstacle Avoidance Path Planning," *2014*

- Seventh Int. Symp. Comput. Intell. Des.*, no. 1, pp. 429–432, 2014.
- [26] C. Zhang, M. Niu, H. Wu, J. He, and Y. Xiao, "A multi-motor synchronous control algorithm for artificial potential field with adjacent attractive force," *26th Chinese Control Decis. Conf. CCDC 2014*, pp. 820–825, 2014.
- [27] M. Hazewinkel, *Encyclopaedia of Mathematics (set)*. Springer Netherlands, 1994.