Real World And Virtual Object Obstacle In Augmented Reality Using Scene Perception

Aninditya Anggari Nuryono, Alfian Ma'arif, Siti Fatimah Anggrahini

Abstract : Augmented Reality is a technique for combining digital content with the real world in real-time. Intel RealSense 3D cameras are used to produce digital content in markerless based Augmented Reality. This camera reconstructs a real environment in three dimensions. Scene perception is a method for reconstructing real environments in three dimensions. Utilization of this camera in Augmented Reality in the form of an autonomous agent. An autonomous agent has a navigation function to get to the destination point by searching for paths called pathfinding. Autonomous agents have three behaviors, namely, seek, arrive, and action selection. These behaviors are used autonomous agents to get to the destination point by avoiding virtual and real obstacles that exist in the real world. The scene perception method is used to make a mesh. This mesh is a virtual grid in the real world that is used as an Augmented Reality area. The navigation results of the autonomous agent using the scene perception method in Augmented Reality can work properly. Autonomous agents can go to their destination point by avoiding virtual and real obstacles.

Index Terms : Augmented Reality, Autonomous Agent, Scene Perception, Markerless, Pathfinding, Mesh, Intel RealSense

1. INTRODUCTION

Augmented Reality (AR) by definition is the merging of digital content with the real world in real-time created by computers. Users can see two-dimensional and three-dimensional objects projected by the camera to the real world in AR. Utilization of cameras such as webcams and three-dimensional (3D) cameras such as those with depth can be used in AR technology. One of the depth technology in three-dimensional cameras is Intel RealSense [1]. This camera can reconstruct real environments in three dimensions. Utilization of a three-dimensional camera to AR in the form of an autonomous agent, which is a unit that behaves independently using artificial intelligence. Autonomous agent navigation in AR applications using the Intel RealSense R200 camera and Unity 3D game engine. Agencies that use artificial intelligence focus on movement and navigation, such as agent-non-player characters (NPCs). Transfer the location of the agent from the starting point to the destination using the pathfinding method. Pathfinding is a method for obtaining a path between two points. In these paths, there are usually obstacles that cause the agent to be blocked and cannot pass through the path. The agent can get to the destination by finding another path that is not blocked by obstacles. On a large map and has many obstacles can cause more crossing points. The multitude and breadth of cross-point points require a method to find the intersection points that are not close to each other. The method used to solve this problem is called pathfinding or planning [2].

- *Aninditya Anggari Nuryono received his M. Eng from Department of Electrical Engineering and Information Technology Universitas Gadjah Mada, Indonesia. E-mail: anindityanuryono@mail.com*
- *Alfian Ma'arif is a lecturer at Department of Electrical Engineering, Universitas Ahmad Dahlan, Indonesia. E-mail: alfianmaarif@ee.uad.ac.id*
- *Siti Fatimah Anggrahini is a student at Department of Electrical Engineering, Universitas Ahmad Dahlan.*

1. LITERATURE REVIEW

One of the in-game AI components that can be applied in AR is pathfinding. Related research to pathfinding uses mapping navigation conducted on simulations in Unity 3D [3]. This research optimizes tactical navigation and pathfinding to optimize the mission distance from the tugboat dock to the ship dock. This navigation uses only four points and is described as a Line Renderer on Unity 3D. The calculation results obtained coordinate distance heuristic straight lines at each angle. A using the marker less method is game cards. AR RPG will appear above the game card which is viewed from the screen using the webcam. [4]. AR RPG game is played in multiplayer. The next related research was carried out using the marker less method [5]. This research uses 3D transformation, which is the basis for the movement of digital objects with interaction techniques in virtual environments. Interactions that occur using human hands and virtual objects. Tracker required for AR virtual interactions. 3D interaction techniques are used in the development of virtual reality. Subsequent research was conducted by Gianibelli [6] that an obstacle avoidance algorithm of virtual forces for path planning. Obstacle avoidance system will be tested on a mobile robot. In a study conducted by Kaydin, et al. Presented an adaptive grid path planning technique, an image-based approach to generating navigation mesh (NavMesh). In static path planning, the adaptive grid method shows better performance. The adaptive grid can be applied to both static and dynamic planning [7]. Based on several studies that have been reviewed previously, this paper uses an Intel RealSense 3D camera that is used to reconstruct the real world and create a virtual grid. This virtual grid is used so that virtual objects can be in the real world and display Augmented Reality moving towards the destination point by avoiding virtual obstacles objects [8] [9][10] and real-world obstacle [11] on the real environment based on agent behavior. AR technology consists of a combination of real-world and computer graphics, the interaction with objects in real-time, the detection of objects or Images, and provides contextual data and information [12]. Augmented Reality uses a marker [13] [14] and markerless methods [4] [15] [16].

2. PROPOSED METHOD

The proposed method of this system in Fig. 1 with an explanation as follow. Start Application is about going to the desired starting position in the user environment and start the app from the Intel RealSense menu. Walk around and scan are when the application has started, a text will appear, letting the user place the dot into an area with a certain distance. When you get that distance, there is information to place a point in an area with a less flat surface or more area structure. The start button to perform the scan will appear when the previous conditions are met.

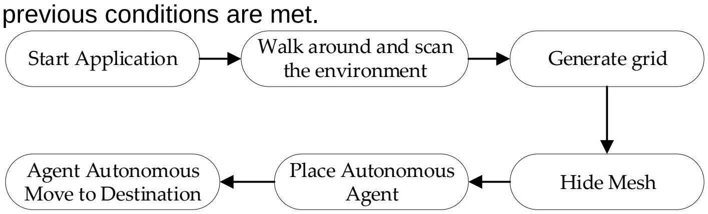

Fig. 1. *Block diagram of the Augmented Reality system*

Then, generate a grid. This stage starts with a calculation where all the existence of the floor is the area that has been scanned by using the perception scene. The farthest positions of each other are used to generate the grid between these points. This grid is a collection of nodes, with each node on set un-walkable. Set to walkable if the existing floor position is within the node area. A valid floor node will result in a value depending on how close the floor node is to the un-walkable node. This information is used as a heuristic in pathfinding. The grid will remember the last position of the object to be tracked. This will be used with a pathfinding algorithm to find the shortest path between AR starting point and AR destination point. Hide Mesh is a visible mesh that can be set on-off. When scanning the environment, with an excellent environment to see that it has been scanned or not, the condition may result in the system slightly lag when compared to not displaying the mesh. Therefore, to make the system run smoother, the visible mesh is hidden when scans and grid making is done. The last is placing a virtual object. Unity 3D is used to display virtual objects in the real world. 3D environments in the real world are used as AR terrain. Then, an autonomous agent moves to a destination point. An autonomous agent will avoid virtual object obstacles [17] and the real object obstacle. In a dynamic environment allows agents to make decisions in real-time so that the environment that has been known beforehand is included in subsequent decision making. Decisions taken and made in real time must have a predetermined navigation pattern framework. In a suitable and general navigation pattern depicted in three aspects namely, action selection, steering dan locomotion [18]. Action selection includes the state of the goal and determines how to obtain that goal. State machines can be used as representations such as predefined state patterns or when state conditions change. Steering is a calculation of movement based on the current state. The steering calculation is used using the magnitude and direction of the driving force. Then locomotion is a mechanical result of the movement of the agent. Humans, cars, and planes have different patterns of movement. Locomotion describes how the agent moves (legs, wheels) and parameters of the movement (maximum speed, maximum force, mass). These three aspects form artificial intelligence [19]. Locomotion on Newton's law-based agent, that is the force produced by mass multiplication with acceleration. Locomotion uses a simple model with a distributed mass that applies forces in various directions based on the body. The movement is limited by determining the maximum force and maximum speed, with a mass of 1 kg, a maximum speed of 1 m / s, and a maximum force of 1 N. The agent must also have a rigid body component and a collider component but do not use gravity at the rigid body. The agent moves because of the force on the rigid body. Seek's behavior tries to move to the target as fast as possible. The speed desired by the agent's behavior towards the target has a maximum speed. The direction of the force is calculated based on the difference in the speed desired by the current speed. In Fig. 2 can be seen as the behavior of seeking agents.

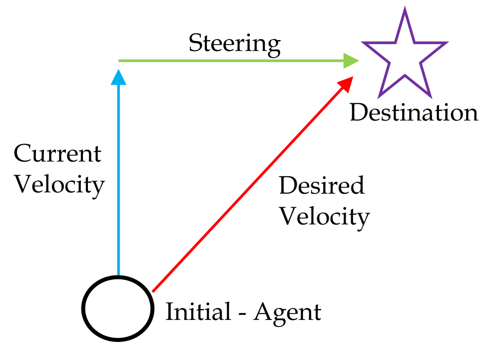

Fig. 2. *Seek Agent behavior*

The arrive behavior is almost the same as the seek behavior, but the arrive behavior stops when reaching the goal. The arrive behavior calculates the distance between the agent and the destination. Seek and arrive behavior is not appropriate to be used to overcome obstacles, so it is necessary for the obstacle behavior. Obstruction behavior of the agent to determine when there is an obstacle must be avoided. This behavior calculates the force that changes the agent's path to avoid obstacles, which can be seen in Figure 3. The combination of arriving and obstruction behavior is implemented to the agent called action selection. This behavior will be used by virtual objects in Augmented Reality. Augmented Reality will be projected using an Intel RealSense camera.

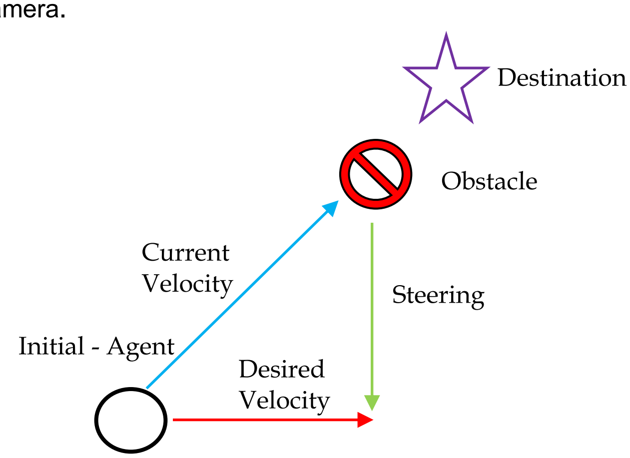

Fig. 3. *Obstacle Detection*

3. EXPERIMENTAL AND RESULT

Unity 3D and Intel RealSense cameras are used to make Augmented Reality. The stages of testing in Augmented Reality starts from the positioning of the Intel RealSense camera that faces the real environment. The camera must face the environment with a small flat area or a lot of geometry and position the camera facing the area with a maximum distance of 2.5 meters. This is because of the limitations of Intel RealSense hardware. After getting the appropriate area and distance, the "start" menu will appear on the screen. This shows that the reconstruction of the real environment will run and is used for the Augmented Reality area. Furthermore, the green virtual grid will be raised from the process of reconstruction of the real environment. In order to get a larger reconstruction area, the user must move around the user slowly. The results of the virtual grid formation process are not an error, because a fast movement that can suddenly cause loss tracking from scanning Intel RealSense cameras. After the virtual grid is formed, the virtual object will fall into the virtual grid area by removing the plane which is the terrain area of the object in the Unity 3D simulation, so that the virtual object can move above the real world. Furthermore, the target can be moved and will cause the Augmented Reality virtual object to follow the object. The seek, arrive, and action selection agents of the scene method agent are integrated using the scene perception method. The scene perception method will create a mesh. This mesh is a virtual grid in the real world that is used as an Augmented Reality area. Scene perception can be seen in Fig. 4 and the Augment Reality application menu interface on Unity 3D can be seen in Fig. 5.

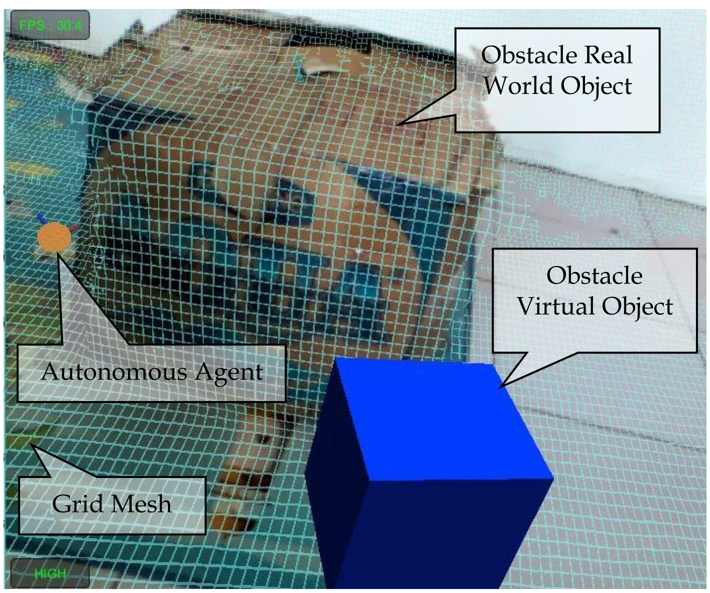

***Fig. 4**. Scene perception with mesh grid*

In Fig. 4, Scene perception builds a digital representation of the observed environment and estimates the camera pose estimation in real-time. Estimation of the camera pose is called localization or tracking and building a digital representation of the observation environment is called reconstruction. Tracking cannot work properly when the observed environment does not have enough geometric variation or does not have texture. This condition occurs when the camera's view room only contains flat surfaces such as walls, floors, or empty tables. The camera movement is aimed based on the user's movement when used smoothly and naturally, which is not sudden and fast. Camera movement that quickly and suddenly can reduce the quality of tracking and can trigger re-tracking.

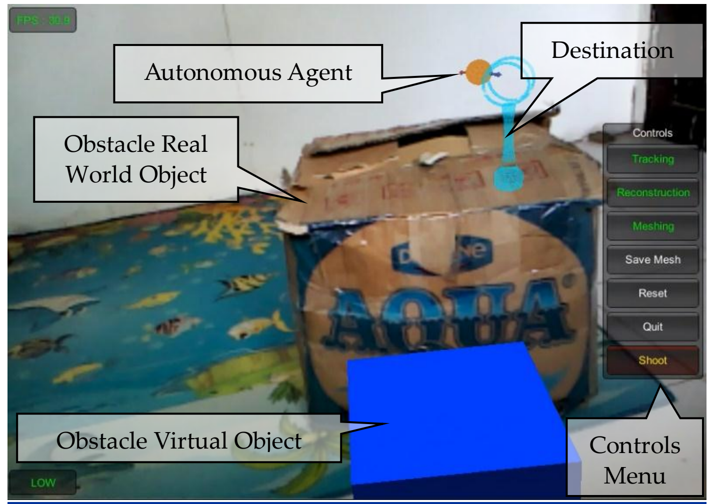

***Fig. 5**. User Interface with hidden mesh grid*

In Fig. 5, there is tracking, reconstruction, meshing, save mesh, reset, quit and shoot menu. Tracking functions are to track the surrounding environment and return the tracking results to the way they were when loss tracking occurred. Reconstruction menu functions to reconstruct the real environment so that virtual objects can be used to be in a real environment. Meshing serves to make three-dimensional real environments. Save mesh function to save the results of meshing. Reset functions to reset when scanning results are not as desired. Shoot menu can issue a virtual object in the form of a box. The virtual grid is hidden and there are agents that move towards the virtual destination point by avoiding virtual objects in the form of blue squares. The agent has to arrow directions attached, are red and blue. The direction of the blue arrow indicates the direction to the destination point, while the direction of the red arrow indicates the detection of obstacles around the agent that must be avoided. Agent path in the real world with the real obstacle in the form of "cardboard" in front of the agent. The path of movement can be seen in Fig. 6. The position of the starting point agent is on the x-axis with coordinates 0.14, on the y-axis with coordinates -0.12, and on the z-axis with coordinates 0.73 and the position of the destination point is on the x-axis with coordinates -0, 45, on the y-axis with coordinates 0.14, and on the z-axis with coordinates 1.26. The travel time needed by the agent to get to the destination point is 6.2 seconds. The agent moves and takes the direction of the lane to the left to avoid obstacles.

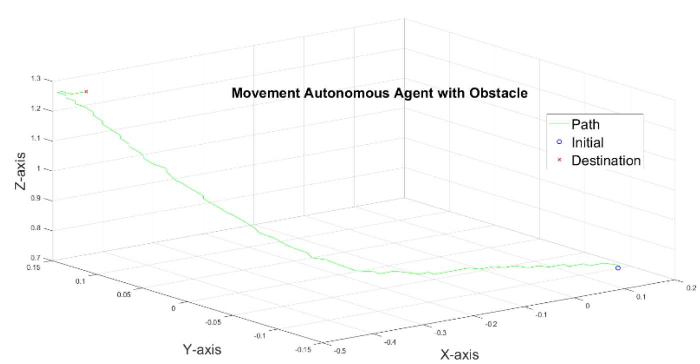

***Fig. 6**. Movement autonomous agent avoid obstacles*

The flow of the agent's movement without obstacles can be seen in Fig. 7. The position of the starting point agent is on the x-axis with coordinates 0.34, on the y-axis with coordinates -0.04, and on the z-axis with coordinates 0.63 and the position of the destination point is at x-axis with coordinates 0.02, on the y-axis with coordinates 0.19, and on the z-axis with coordinates 0.93. The travel time needed by the agent to get to the destination point is 4.5 seconds. The agent moves straight to the destination point.

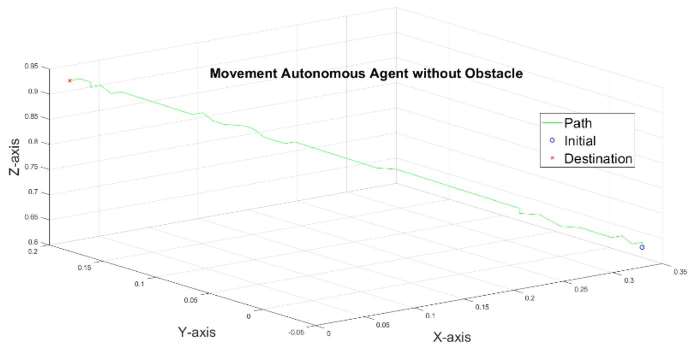

***Fig. 7**. Movement autonomous agent without obstacle*

4. CONCLUSION

The scene perception method is used to make a mesh. This mesh is a virtual grid in the real world that is used as an Augmented Reality area. The navigation results of the autonomous agent using the scene perception method in Augmented Reality can work well. Autonomous agents can go to their destination point by avoiding virtual and real obstacles. The implementation of the scene perception method using an Intel RealSense 3D camera can be applied in the Augmented Reality application. Scene perception is able to combine obstacles in the real world with virtual obstacles. Virtual and real obstacles can be detected by seeking, arrive, and action selection behavior on the autonomous agent.

ACKNOWLEDGEMENT

All authors have contributed to this work and thank reviewers for their valuable suggestions on improving the quality of this paper.

REFERENCES

[1] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, “Intel(R) RealSense(TM) Stereoscopic Depth Cameras,” 2017 IEEE Conf. Comput. Vis. Pattern Recognit. Work., pp. 1267–1276, 2017.

[2] I. Iswanto, O. Wahyunggoro, and A. Cahyadi, “Path Planning Based on Fuzzy Decision Trees and Potential Field,” Int. J. Electr. Comput. Eng., vol. 6, p. 212, 2016.

[3] A. S. Anisyah, P. H. Rusmin, and H. Hindersah, “Route optimization movement of tugboat with A* tactical pathfinding in SPIN 3D simulation,” in 2015 4th International Conference on Interactive Digital Media (ICIDM), 2015, pp. 1–5.

[4] S. Bedoya-Rodriguez, C. Gomez-Urbano, A. Uribe-Quevedoy, and C. Quintero, “Augmented reality RPG card-based game,” in 2014 IEEE Games Media Entertainment, 2014, pp. 1–4.

[5] Y. Ariyana and A. I. Wuryandari, “Basic 3D interaction techniques in Augmented Reality,” in 2012 International Conference on System Engineering and Technology (ICSET), 2012, pp. 1–6.

[6] A. Gianibelli, I. Carlucho, M. D. Paula, and G. G. Acosta, “An obstacle avoidance system for mobile robotics based on the virtual force field method,” in 2018 IEEE Biennial Congress of Argentina (ARGENCON), 2018, pp. 1–8.

[7] A. Akaydın and U. Güdükbay, “Adaptive grids: an image-based approach to generate navigation meshes,” Opt. Eng., vol. 52, no. 2, p. 027002, 2013.

[8] I. Iswanto, A. Ma’arif, O. Wahyunggoro, and A. Imam, “Artificial Potential Field Algorithm Implementation for Quadrotor Path Planning,” Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 8, pp. 575–585, 2019.

[9] I. Iswanto, O. Wahyunggoro, and A. Cahyadi, “Path planning of decentralized multi-quadrotor based on fuzzy-cell decomposition algorithm,” in AIP Conference Proceedings, 2017, vol. 1831, p. 20060.

[10] Q. Lin, X. Wang, and Y. Wang, “Cooperative Formation and Obstacle Avoidance Algorithm for Multi-UAV System in 3D Environment,” in 2018 37th Chinese Control Conference (CCC), 2018, pp. 6943–6948.

[11] O. Wahyunggoro, T. Adji, A. Cahyadi, I. Ardiyanto, and I. Iswanto, “Local information using stereo camera in artificial potential field based path planning,” IAENG Int. J. Comput. Sci., vol. 44, pp. 316–326, 2017.

[12] M. Ramirez, E. Ramos, O. Cruz, J. Hernandez, E. Perez-Cordoba, and M. Garcia, “Design of interactive museographic exhibits using Augmented reality,” in 23rd International Conference on Electronics, Communications and Computing, CONIELECOMP 2013, 2013, pp. 1–6.

[13] B. Patrão, L. Cruz, and N. Gonçalves, “An Augmented Reality Application Using Graphic Code Markers,” in 2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), 2018, pp. 193–194.

[14] C. Lim, C. Kim, J. Park, and H. Park, “Mobile Augmented Reality Based on Invisible Marker,” in 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), 2016, pp. 78–81.

[15] R. Furlan, “The future of augmented reality: Hololens - Microsoft’s AR headset shines despite rough edges [Resources_Tools and Toys],” IEEE Spectr., vol. 53, no. 6, p. 21, 2016.

[16] T. Araújo et al., “Life Cycle of a SLAM System: Implementation, Evaluation and Port to the Project Tango Device,” in 2016 XVIII Symposium on Virtual and Augmented Reality (SVR), 2016, pp. 10–19.

[17] M. Mikhail and N. Carmack, “Navigation software system development for a mobile robot to avoid obstacles in a dynamic environment using laser sensor,” in SoutheastCon 2017, 2017, pp. 1–8.

[18] C. Reynolds, “Steering Behaviors For Autonomous Characters,” in Game Developers, 1999, pp. 763–782.

[19] P. Norvig and S. J. Russell, Artificial Intelligence: A Modern Approach. Upper Saddle River, NJ: Prentice Hall, 2010.