

Performance Analysis of Specification Computer and Mobile with Implementation Tawaf Virtual Reality using A* Algorithm and RVO System

Moh. Zikky¹⁾, M. Jainal Arifin²⁾, Kholid Fathoni³⁾, Agus Zainal Arifin⁴⁾

Game Technology Study Program, Multimedia Creative Department
Politeknik Elektronika Negeri Surabaya, Jl. Raya ITS Sukolilo Surabaya ^{1,2,3)}
Faculty of Information and Communication Technology, ITS Surabaya ⁴⁾
Email: zikky@pens.ac.id¹⁾,
jainal.arief31@gmail.com²⁾, kholid@pens.ac.id³⁾, agusza@cs.its.ac.id⁴⁾

Abstract

High-Performance Computer (HPC) is computer systems that are built to be able to solve computational loads. HPC can provide a high-performance technology and short the computing processes timing. This technology was often used in large-scale industries and several activities that require high-level computing, such as rendering virtual reality technology. In this research, we provide Tawaf's Virtual Reality with 1000 of Pilgrims and realistic surroundings of Masjidil-Haram as the interactive and immersive simulation technology by imitating them with 3D models. Thus, the main purpose of this study is to calculate and to understand the processing time of its Virtual Reality with the implementation of tawaf activities using various platforms; such as computer and Android smartphone. The results showed that the outer-line or outer rotation of Kaa'bah mostly consumes minimum time although he must pass the longer distance than the closer one. It happened because the agent with the closer area to Kaabah is facing the crowded peoples. It means an obstacle has the more impact than the distances in this case.

Keywords: Performance device, VR Tawaf simulation, HPC for crowded simulation,

1. INTRODUCTION

Hajj is one of the Five Pillars of Islam which obligated for every Muslims who have capabilities to perform the hajj pilgrimage at least once in their lifetime. Hajj activity must be performed in Masjidil Haram Mecca which is Kaabah as the centre of it. Sometimes, prospective pilgrims have not been

able to master the right hajj pilgrimage process likes its main activity namely Thawaf. Thawaf is the activity around The Holy Kaaba 7 times, beginning and ending in the direction parallel to the Hajar-Aswad. Surely, this place is always crowded pilgrims and never been quiet. Every pilgrim should choose the right path to perform its pillars-requirement to get an effective time and work. Everyone who does a mistake tawaf such as miss-facing body because of shifted by crowded people, he/she must repeat until its activity is perfectly done.

To prevent its mistake activity, every pilgrim's candidate should take training and predict how to take a right path of tawaf to minimize the time-consumed because of crowded people-obstacle and fulfil the pillar effectively before doing the real one. One of an immersive technology device which mostly used for training media is Virtual Reality (VR). VR is the effective training media with the immersive visual surrounding which is reduce the risk of activity and the cost (definitely) if compared to make a miniature place of Masjidil-Haram replica. This VR technology constructed with Masjidil-Haram surrounding with virtual guide-book. Thus, every trainer can choose a path around the Kaabah with random people-obstacle behaviour. Then, the VR-user will be moved based on A* algorithm behaviour to pass its path. In this section, this simulation provides a user the first person shooter (FPS) perspective to fell an immersive Masjidil-Haram surrounding. At the end of tawaf, a user will see a time-listing of every rotation. So, from this training, a candidate could understand how to do an effective tawaf with the effective times. The crowded situation a of Masjil Haram (around Kaa'bah) as shown at Figure 1.

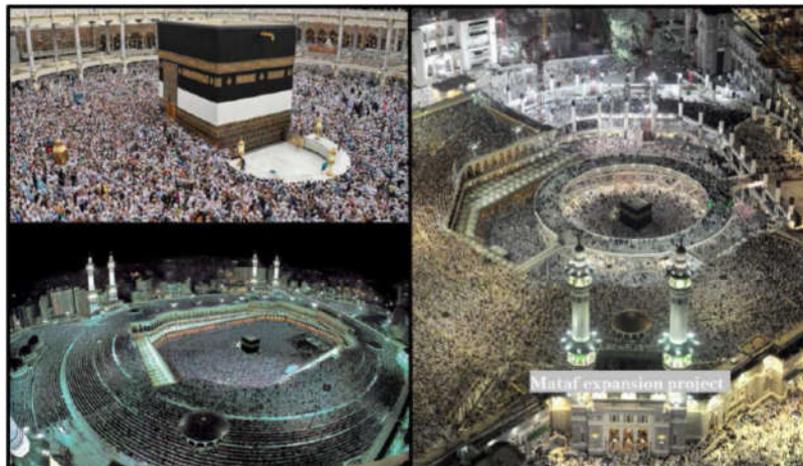


Figure 1. Three views of tawaf taking place within the Mataf region before (left) and after (right) the expansion project (Saudi Press Agency)[1]

2. RELATED WORKS

This part will examine how to make a search algorithm for the effective path on the activities of tawaf when around Kaaba by using the A* algorithm and RVO (Reciprocal Velocity Obstacle). Previous research [1] describes how

the behaviour problems of any dynamic agents to any crowd very much and assembly, for example in the mall, the market and the activities of tawaf into Al-Haram Mosque. Next [2] explains the simulation of the crowded on tawaf activities by using FSM algorithm and RVO. Thus, it also describes the crowd on the tawaf and Sa'i activities; so, pilgrim-candidate has to understand in avoiding another pilgrim's dynamic-collision [3].

2.1 Dynamic Agents

The agent made has its character in planning the movement and shortening the bands of the agents. By adding the scale of the actual speed for the agent. The agents are moving toward the determined targets, and every movement may get the stack condition with other agents. So, this research technique will optimised by combining designs of free movement in the crowd of those agents. Rahul Narain, et al [1] used the simulation as a local planning module in conjunction with a global planner, such as a roadmap-based algorithm or the continuum-based optimization on a coarse grid, for the simulation of large dense crowds. They used the main simulation loop is structured as Figure 2; At every time step, after each agent was determined its preferred velocities using a global planner, the discrete set of agents is converted to a continuous representation, yielding density ρ and velocity. The UIC is solved in this continuous setting to give the corrected velocity field (v), and the end, agents determine their actual motion based on the velocities sampled from its velocity field.

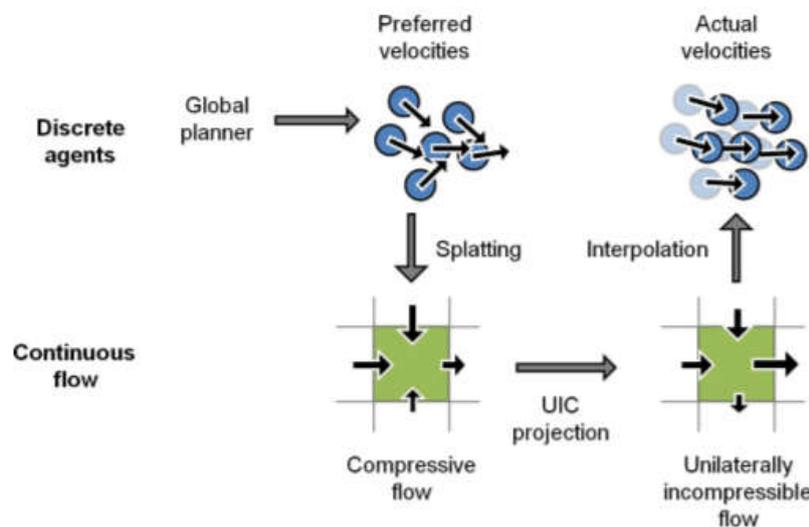


Figure 2. Behaviour Agents for Global Planner [1]

2.2 FSM and RVO

The FSM was used as parameter and optional evaluation of the state of the agents to determine its path movement, while RVO was used for distance parameter of agents who could be hit by another agent. When one of the

agents rammed the other agents that are in the surrounding area, then this agent will move or reverse the movement of the crash's agent.

Sean Curtin, et al [2] made a model the behaviours of agents by coupling together a high-level finite-state machine (FSM) with a low-level, local-collision avoidance (LCA) mechanism for performing local motion planning. The FSM evaluates the agent's physical state and property set to define the parameters for the LCA algorithm, which, in turn, updates the agent's physical state. Furthermore, RVO performs local collision avoidance by computing space of collision-free velocities for an agent with respect to a number of neighbouring agents. Both approaches compute trajectories with respect to a preferred velocity for each agent. To model the changes in interactions between agents, both approaches use a set of parameters which can be altered to change the influences agents have on each other. This interaction as showed in Figure 3.

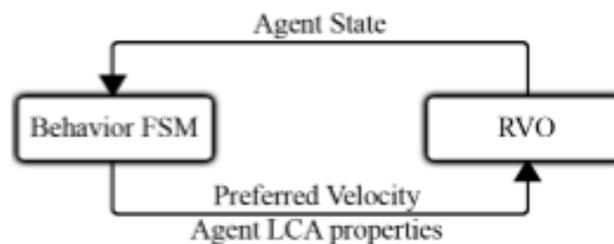


Figure 3. Agent State Using Combination Behavior FSM and RVO [2]

The arrangement of Al-Haram Mosque area above is the order which is used for the simulation of the movement of the pilgrims in the research, moving seven times with counterclockwise. The layout of the Tawaf area in Masjidil haram and its direction as shown in Figure 4.

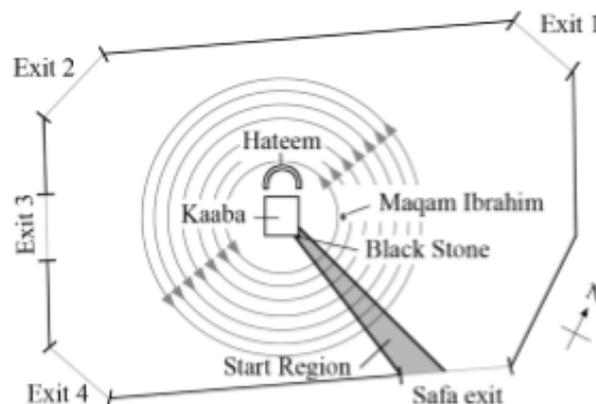


Figure 4. Simulation of Movement Pilgrims for the Tawaf [2]

2.3 Tawaf Activities

Pilgrims presented as a unit or the existing queue agents inside Masjidil-Haram area. Several factors that influence the effectiveness of tawaf activities

are the distance, movement, and its path's choice of pilgrim agents. Figure 5 is the flowchart, which represents how the agents are created and move find the best way to avoiding obstacle with effective AI.

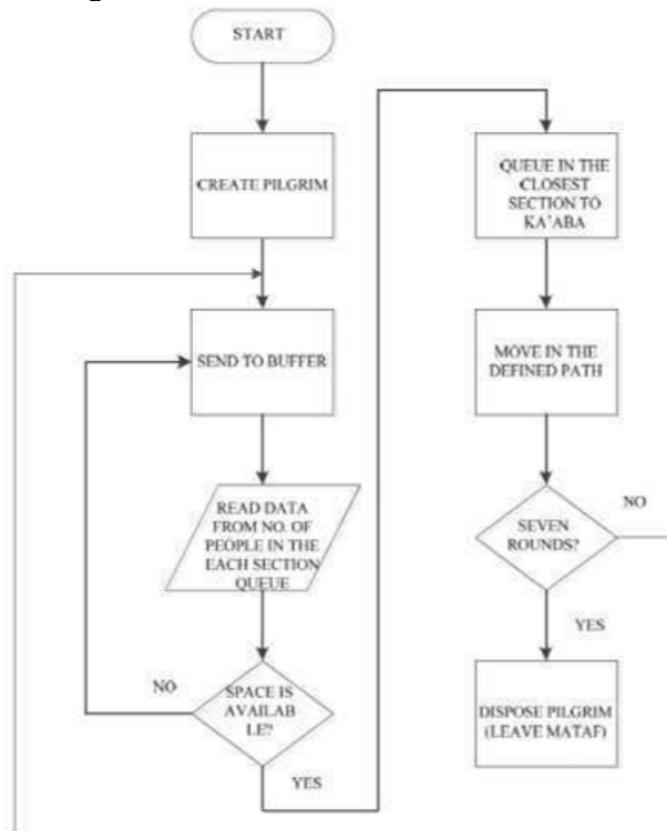


Figure 5. Flowchart Tawaf Activities

3. ORIGINALITY

This study measures the devices performances in the implementation of pilgrim tawaf activity VR simulation based on smartphone and PC. Application of this learning media contains not only graphical, sound and textual information, but also an interactive between NPC and user. This application uses VR technology and interaction in gaming technology to produce immersive effects. For object visualisation, this application is rendering the several 3D objects replica likes Kaabah, pilgrim's people, Masjidil Haram, and another object with similar scale comparison. Its application will also suggest user show to take the effective path during tawaf and go around a Kaabah.

4. SYSTEM DESIGN

System design of this research based on the movement of NPC (Non-Player Character) Pilgrims as the simulation in tawaf activity. Then, we define three categories of the NPC and flocking movement.

A. NPC Rotation

NPC Rotation is the NPC has circled Kaaba, with the centre of Kaaba as a wandering shaft. The rotation or sequence is a change in the position of the object with how to roll through the determined centre and a certain angle. Rotation of the geometry transformation of α crowd rounded with the opposite direction of clockwise.

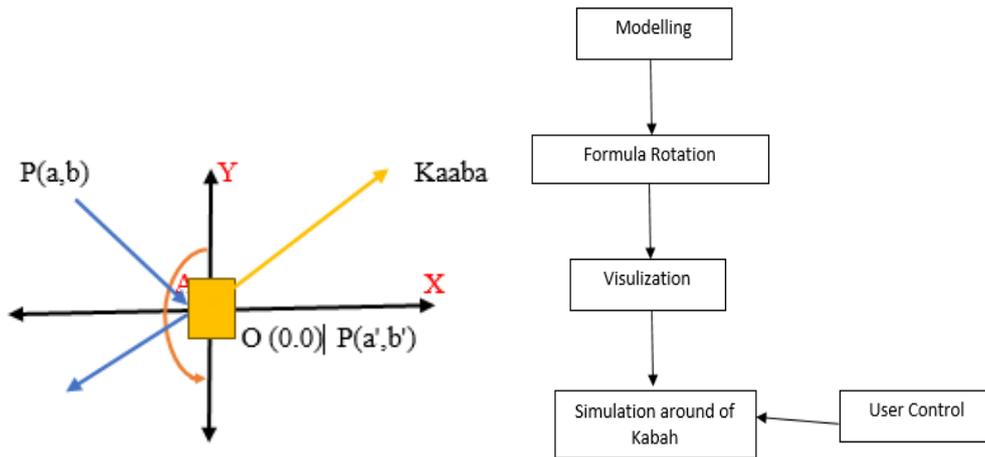


Figure 6. (a) Formula of Rotation (b) Flowchart system of NPC Rotation

Two images in Figure 6 showed the formula and flowchart design for NPC rotation. Figure 6(a) is the scheme formula of rotation with $O(0,0)$ as the pivot as much as α degree value, and Figure 6 (b) showed how the system design of its rotation created.

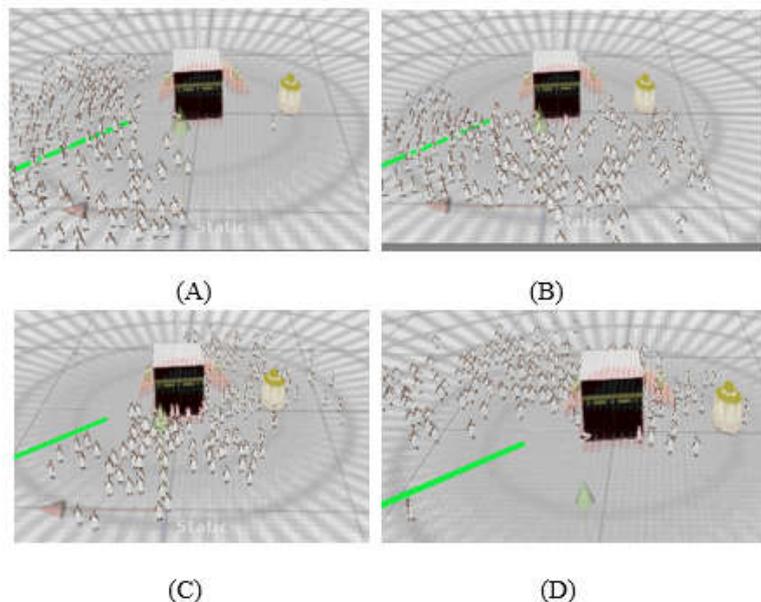


Figure 7. (A) NPC agent located on the axis position $-X$, (B) NPC agent located on the axis position $-Y$, (C) NPC agent located on the position of $+X$, (D) NPC agent

located on the position of +Y. The image above showed the implementation of rotation formula and its system design; the NPC object is rotating with Kaabahpivot.

B. NPC NavMesh

NPC NavMesh or Navigation Mesh is the NPC that uses the process geometry (which already provided on game engine Unity feature) with A* Algorithm [5]. Navmesh has some points that have to be done such as choosing obstacles and desired target[6], check navigation from everyagent. For example, radius from agent and target. Navmesh is very useful to find the quickest way to reach the target position already determined

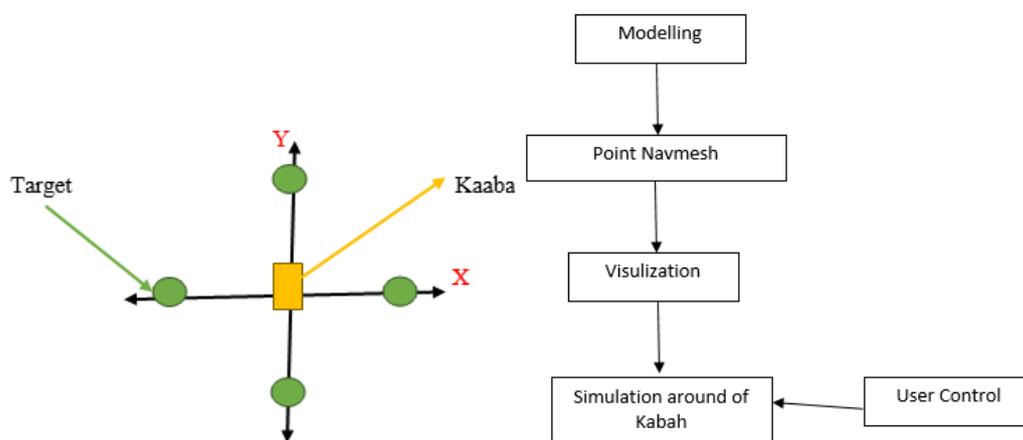


Figure 8. (a) Pointtarget of Navmesh, (b) flowchart system of NPC using navigation mesh (NavMesh) pathfinding.

Figure 8 shows the points ofNavmeshpath and flowchart system of it; in Figure 8 (a), wemakefour targets thatpivoted with Kaaba. To construct the X and Y axis, we make two targets that used as the position of the movement parameters from NPC NavMesh, and Figure 8 (b) is a flowchart system design for NPC NavMesh pathfinding.

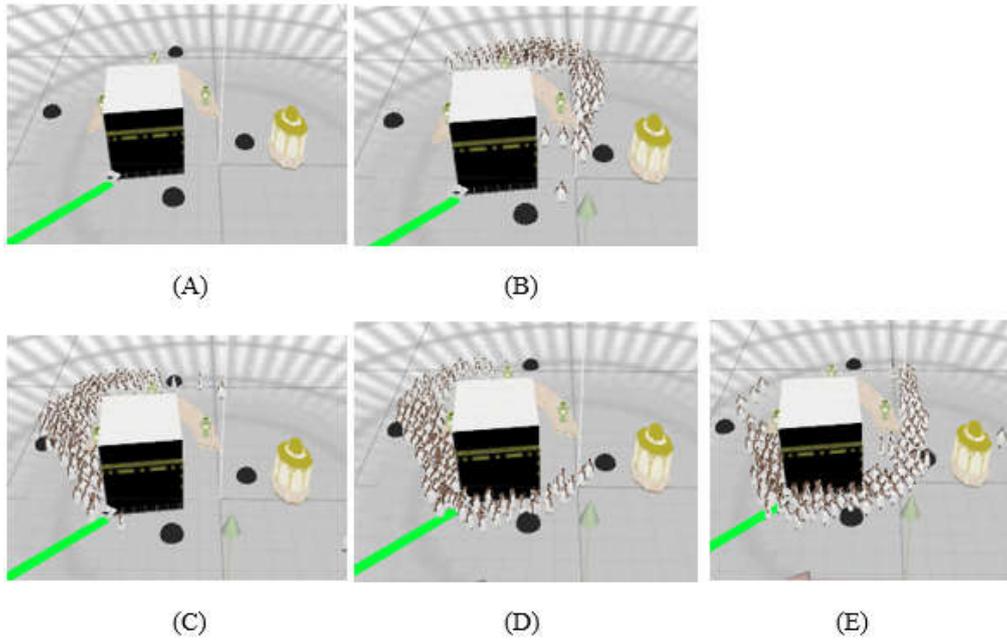


Figure 9. (A) First conditions and the placement of the target position between Kaaba and in the X and Y axes, (B) NPC agent to target position +Y, (C) NPC agent to target position -X, (D) NPC agent to target position -Y, (E) NPC agent to target position +X.

C. NPC Follow Target

NPC Follow Target is the NPC which behaviour following the target and always changed its position on movement around Kaaba. NPC follow-target has a certain speed and radius in moving [8].

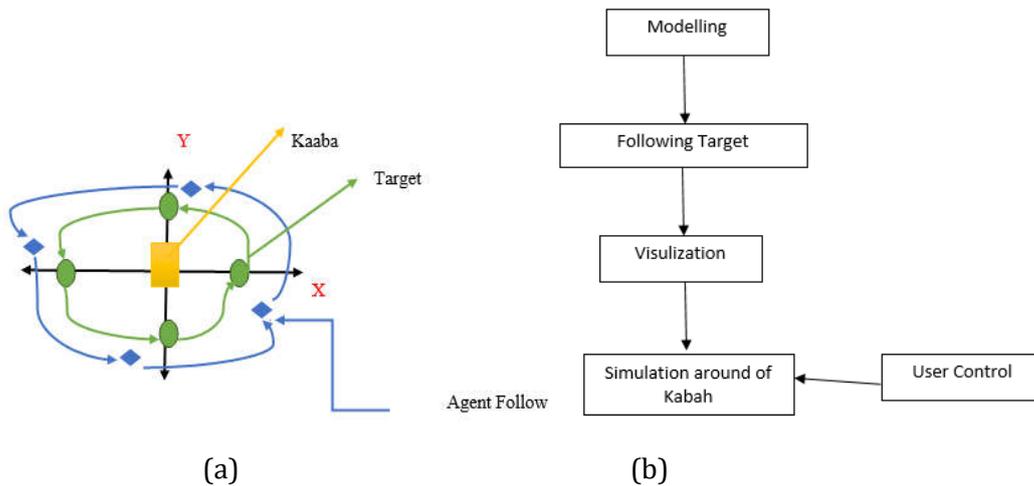


Figure 10. (a) Scheme design of Follow Target, and (b) Flowchart system of Follow Target

The two images in Figure 10 are scheme design of follow-target agent and flowchart design for NPC follow-target; figure 10 (a) showed that we use

one-object as a target and another object as a follower. Figure 10 (b) showed the system design of followtarget's flowchart.

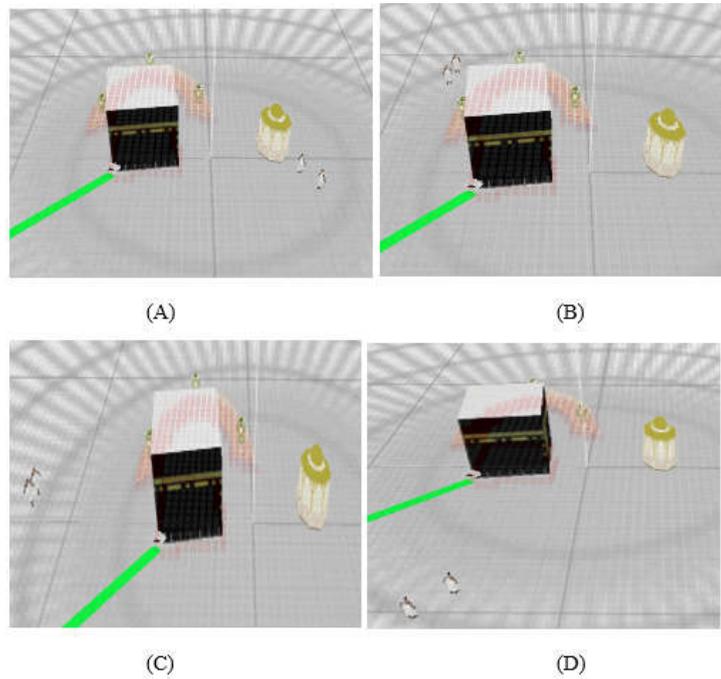


Figure 11. (A) NPC agent and target is located at position axis + X, (B) NPC agent and target is located at position axis +Y, (C) NPC agent and target is located at position axis -X, (D) NPC agent and target is located at the position axis -Y axis. In this simulation, object will always follow movement object's target.

D. NPC of Flocking

This NPC (Non-Player Character) behaviour was implemented into a group formation that embedded the similar behavior of his Chief, and they will always follow him. This Flocking [7] behavior has three basic functions; separation, cohesion, and alignment.

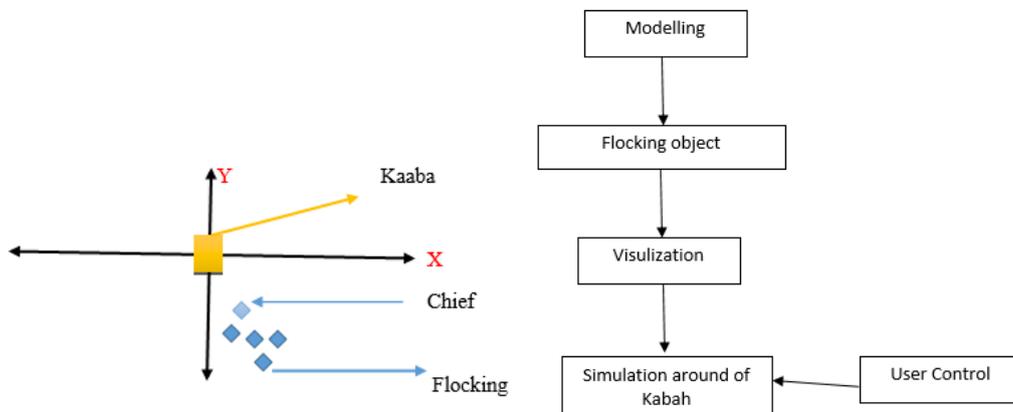


Figure 12. (a) Formula NPC Flocking (b) System Design NPC Flocking

The two images in Figure 12 are the visual formula and flowchart system of NPC flocking; Figure 12 (a) showed the sketch of NPC flocking, we use one object as a Chief and clone it to become 1000 flocking objects. Figure 12 (b) is the flowchart of how NPC's flocking works.

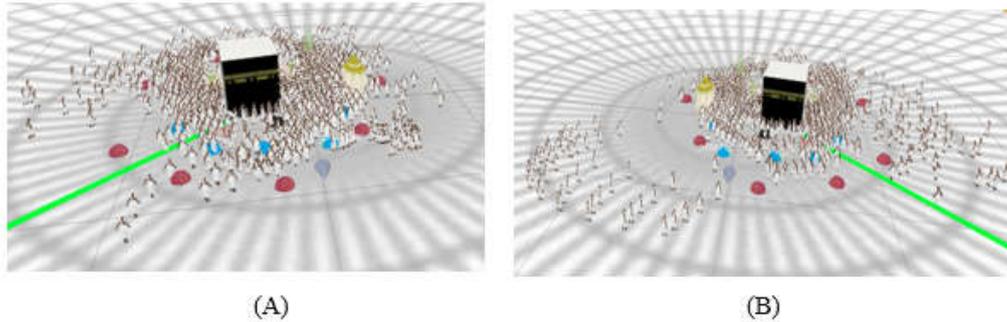


Figure 13. (A, B) Flocking is applied to the NPC agents with the rotation, NavMesh, and follow-target behaviours with around 1000s pilgrim characters.

E. User or Pilgrim with A* Algorithm

This user is playing as the character object that was embedded as a parameter to search the effectiveness path using A*-algorithm. He will walk around Kaaba gathers with the other agents of NPC in crowded Kaaba's area. A* Algorithm here is used for determining the shortest and fastest path of tawaf activity.

The simple formula of A* algorithm [8] which used in this project is $F = G + H$, where:

G = Score of each step

H = Score estimation of movement

F = Amount of score in every step + estimates of movement

To start the Player (User) movement from A*, it required a grid or an area which is available for the player. The other NPC agents will be played automatically as the user's obstacles [8]. The player will search the best value or the fastest path to reach the target.

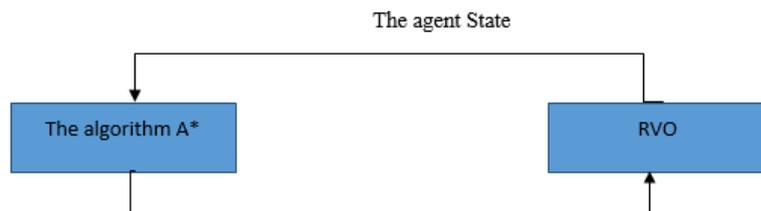


Figure 14. Combination A* and RVO

As the basic principle of A* and RVO, the simulation of tawaf is using two conditions in searching for the effective movement. The first step, a player will search the best target using A* algorithm; then, the second step

he uses RVO (Reciprocal Velocity Obstacles) as the shirking system when he faces obstacles in the area around him.

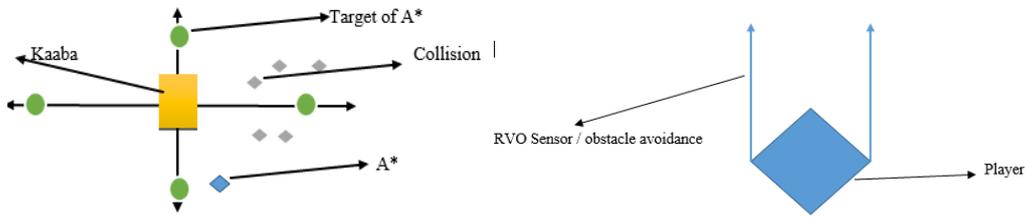


Figure 15. (a) Formula A* (b) Formula RVO

Figure 15 showed the visual scheme of A* and RVO. The algorithm A* will search for the fastest path to reach the point target which has determined, and the RVO algorithm is used for the avoid-sensor from the collision or obstacles around it. The design system of this simulation can be described as a flowchart in Figure 16, which showed several steps of user movement when avoiding an obstacle or another pilgrim and search the effective tracks in this tawaf simulation.



Figure 16. System Design Combination A* and RVO

5. RESULT AND ANALYSIS

In this study, we use three areas of a crowd pilgrims around Kaabah that drawn as line 1, 2, and 3. Each area of its line has different amounts of Pilgrims. The line which is closer to Kaabah will become more crowded with pilgrim than the outside one. Detail configuration as shown in Figure 17. For the first analysis in this research, we use 1000 pilgrims as the NPCs.

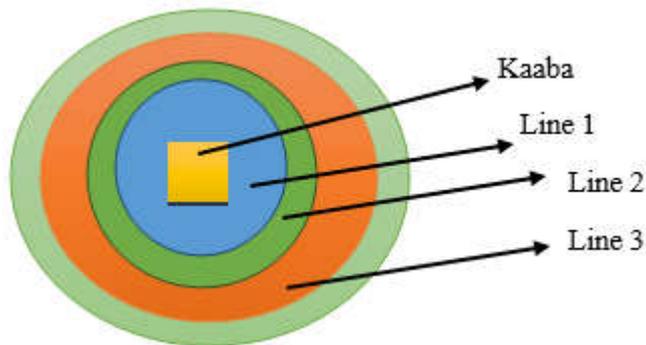


Figure 17. Area Experiment with three lines

Using Figure 17 diagram and 1000 pilgrims, we examined this application using Personal Computer (PC) and Smartphone; we noted several data and analysis as below:

1. Personal Computer (PC)

For the first experiment, we use a personal computer to analyse the time's consuming and needs of speed when we tried this tawaf simulation. We use PC with processor Intel(R) Xeon(R) CPU E3-1225 V2 @3.20 GHz 3.20 GHz and RAM 4.00 GB. With this PC, we get the result as shown in Figure 18.

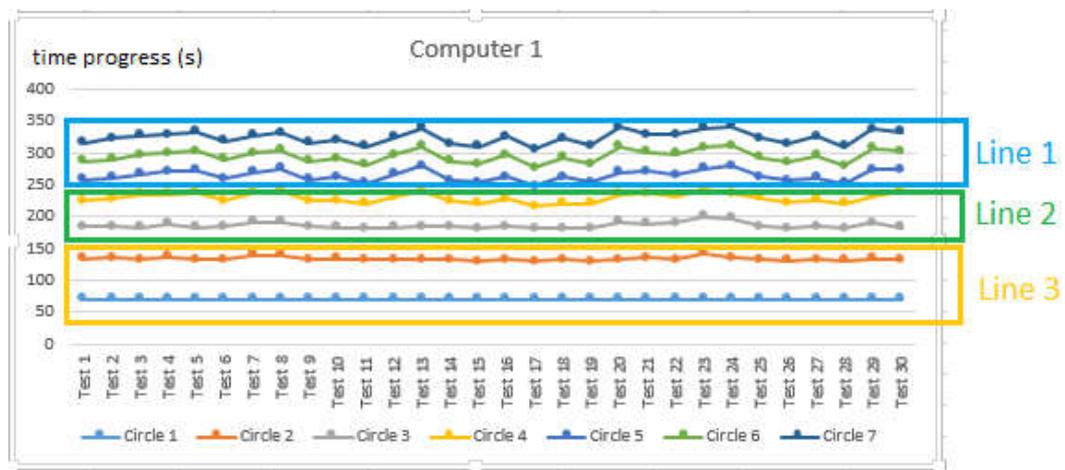


Figure 18. The result of statistics data from rotation 1 to 7

Figure 18 shows the experiment's result data with time and speed parameters of Tawaf simulation using a computer 1. We examined 30 times and noted every rotation with a different colour; the blue colour (labelled as Circle 1) is the first rotation which cycling in Line 3, and so on. From this experiment, we are resulting time-consuming average for doing virtual tawaf, the result as shown in Figure 19.

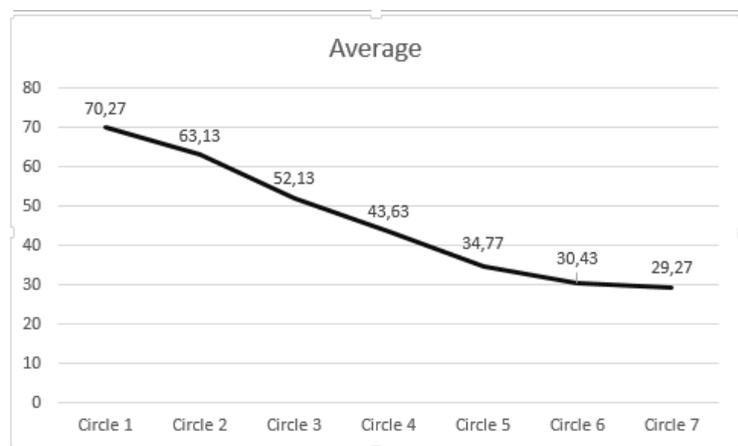


Figure 19. Results of time average in every rotation

Figure 19 shows the average result of 1st analysis using a time and speed parameters of tawaf simulation with computer onespecification. In the top results get the value of each rotation average as follows. For circle 1, the time consumes 70.27 seconds; circle2 needs 63.13 seconds, circlethreeneeds 52.13 seconds,circlefourneeds 43.63 seconds, circlefiveneeds 34.77 seconds, circlesixneeds 30.43 seconds,and for circle 7 consumes 29.27 seconds.

Then, to examine the constant results,we try to note the maximum and minimum results with 30 times the experiment's trial. From this experiment, we get the maximum and minimum values as shown in Table 1.We also compare and run with another PC specification (PC-2), it used processor Intel(R) Core CE i7-5557U CPU @3.10 GHz and RAM 8.00 GB.

From this table, we can conclude that the closer area of Kaabah mostly need much more time's consuming and running at the lowest speed, although its distance is shorterthan the outsider. However, we also get an uncommon value on Circle2(maximum); itshows greater value than a Circle one. It can happenbecause of the obstacles variety that also move around the player.

Table 1. Time's consuming experiments using PC-1 and PC-2

n-Circle	With PC-1			With PC-2		
	Min	Max	Average	Min	Max	Average
<u>Circle-1</u>	70	71	70,27	70	71	70,23
Circle-2	60	72	63,13	59	70	62,83
<u>Circle-3</u>	49	62	52,13	49	53	50,73
<u>Circle-4</u>	34	58	43,63	39	58	43,77
Circle-5	32	44	34,77	30	40	33,93
<u>Circle-6</u>	28	39	30,43	27	35	29,63
<u>Circle-7</u>	27	32	29,27	27	36	29,47

Another side of the experiment's data results as shownin Table 1; we can conclude that the higher specification of PC was not affectedsignificantly in circling process in tawaf simulation. It because the time's consuming will be longer and shorter depending on the pilgrim'samounts in that area.

B. Using a Smartphone with Android OS

The second experiment, we use a smartphone with OS android and its hardware specification:CPU Qualcomm MSM8939v2 Snapdragon 616, Octa-core (4x1.5 GHz Cortex-A53 & 4x1.2 GHz Cortex-A53), GPU Adreno 405. For the first Smartphone or Mobile 1, we get the result's data as Figure 20.

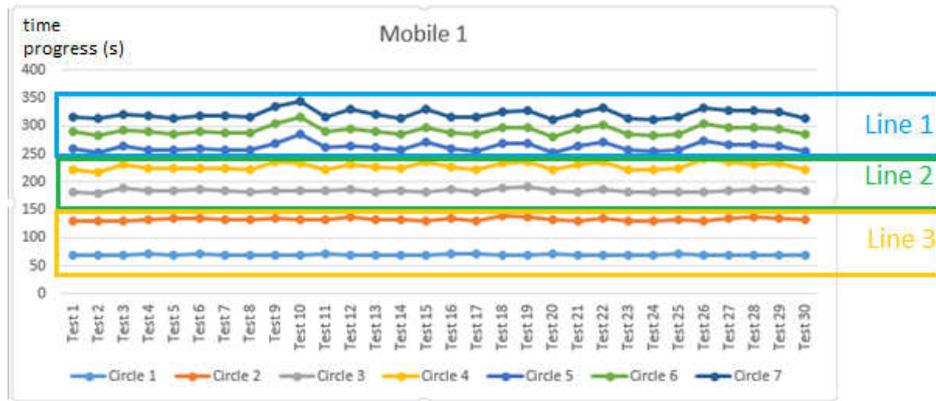


Figure 20. Results of statistical data for every rotation (1 up to 7) with smartphone

As the PC experiment, we also examined this smartphone-part with 30 times of experiments. Figure 21 is showing how data have resulted from the Tawaf simulation using a smartphone device. We also used the same colours as PC experiment for labelling circle one up to circle 7. Then, we calculated the time's consuming averages for this device, and we got data as Figure 21.

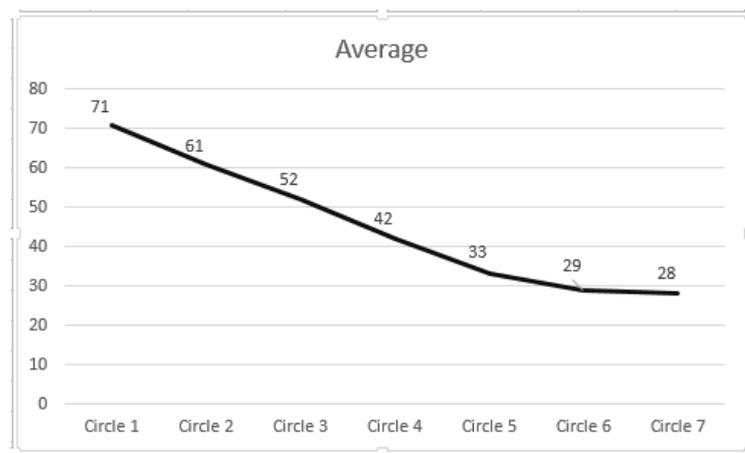


Figure 21. Time-consuming average in every rotation with a smartphone device (Mobile 1)

As the experiment of PC's device, the average data in this experiment showed a similar trend of values; it also needs a longer timing to go around "Circle 1 area" and decreasing continuously for the outsider one. Then, after we examined 30 times, we noted the maximum and minimum of time's consumed as Table 2. In this analysis, we used the 2nd smartphone (Mobile 2) device for comparing the performance of its application with the different smartphone specification; it has CPU Qualcomm MSM8929 Snapdragon 415, Octa-core (4x1.5 GHz Cortex-A53 & 4x1.2 GHz Cortex-A53), GPU Adreno 405. Then we get the data as Table 2.

Table 2. Time’s consuming experiments using Mobile-1 and Mobile-2

n-Circle	With Mobile-1			With Mobile-2		
	Min	Max	Average	Min	Max	Average
Circle-1	70	71	71	80	89	88
Circle-2	60	63	61	77	100	82
Circle-3	49	59	52	62	67	64
Circle-4	39	42	42	49	78	57
Circle-5	32	34	33	41	50	44
Circle-6	28	30	29	34	38	36
Circle-7	27	29	28	33	54	35

The values in Table 2 have different characteristics compared to the PC data result. Especially for Mobile-2 maximum data, there were fluctuated data that resulted with the Mobile-2 device, as we see how Circle 2 value is greater than Circle 1, and also Circle 4 is greater/longer than Circle 3, and so on. This data showed that smartphone device could affect the performance of this Tawaf Simulation. If its fluctuate data described in the chart, we can see Figure 22 how the line is up and down looks like randomise data.

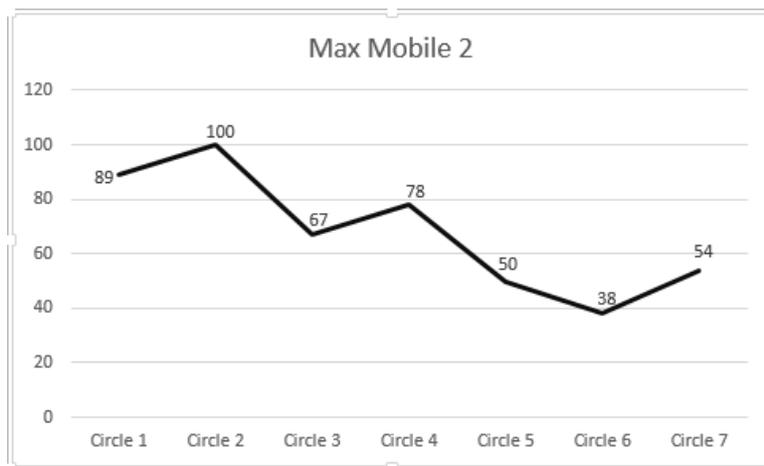


Figure 22. Results of Experiment: Maximum time’s consuming in every rotation with Smartphone devices 2 (Mobile 2).

6. CONCLUSION

In this simulation of tawaf is performed by the crowded condition. We present 1000s pilgrims in this research using an A* algorithm and RVO system to understand and analyze the effective values in every rotation in this simulation. Every result score/value of each rotation is influenced by density or obstacles with the other pilgrims, and also the distance or rotation area.

Thus, the more crowded people/pilgrims allow more slowly movement between the Pilgrims, it consumes longer times although passed the shorter distance. It happened for the closer area to the Kaabah. But if a user chooses the wider distance from Kaabah, usually he needs a shorter time’s consuming, although he must pass the longer distance, because he must walk around with the outside line’s area (*shaft*) of Masjidil Haram.

Device's specifications sometimes affect the performance of this simulation. As the comparison of two devices; using computers and smartphones with different specifications, we have resulted in data that showed higher device specifications usually would show the best performance and faster process.

REFERENCES

- [1] R. Narain, A. Golas, S. Curtis, and M. C. Lin. **Aggregate Dynamics for Dense Crowd Simulation**. University of North Carolina at Chapel Hill.
- [2] S. Curtis, S. J. Guy, B. Zafar, and D. Manocha. **Virtual Tawaf: A Case Study in Simulating the Behavior of Dense, Heterogeneous Crowds**. In <http://gamma.cs.unc.edu/LARGE>.
- [3] M. P. Stennett, D. M. Romano, and O. A. Kurdi. **Modelling and Simulation of Tawaf and Sa'yee: A Survey of Recent Work in the Field**. 29thESM' 2015, 26-28 Oct 2015, Leicester, UK.
- [4] Yenie Syukriyah, (2016). **Penerapan Algoritma A*(STAR) Untuk Mencari Rute Tercepat Dengan Hambatan**. Bandung, Indonesia
- [5] Jur Van Den Berg, et al. **Reciprocal Velocity Obstacles for Real-time Multi-Agent Navigation**. 2008 IEEE International Conference on Robotics and Automation Pasadena, CA, USA.
- [6] Reza Olfati-Saber, 2004. **Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory**. California Institute of Technology
- [7] Maxim Likhachev, **Anytime Dynamic A*: An Anytime, Replanning Algorithm**, *Carnegie Mellon University Pittsburgh, PA, USA*