Camera Placement Based On Director's Style Using Fuzzy Logic

Hartarto Junaedi[123], Mochamad Hariadi[12], and I Ketut Eddy Purnama[12],

[1]Department of Electrical Engineering, Institut Sepuluh Nopember Surabaya (ITS), Surabaya , Indonesia
[2]Department of Computer Engineering, Institut Sepuluh Nopember Surabaya (ITS), Surabaya, Indonesia
[3]Department of Information System, Sekolah Tinggi Teknik Surabaya, Surabaya, Indonesia

Summary
Manual camera placement for a particular scene requires repeated processing and calculations, which will take a lot amount of time and cost. Therefore, automatic camera placement in a virtual environment is suggested. The camera placement is not only done automatically, but it also incorporates a directors style. Incorporating a directors style in a cinematic product will create an impressive ambiance. In this research, an approach based on fuzzy logic is used. This research uses a game simulation with three different scenes and some events – such as running, jumping, squatting, and fighting – as well as two different styles. The camera placement in this research is not merely based on fuzzy output, but it also utilizes several additional modules so that the result corresponds more to the real world. In this game, there is an event selector for each scene to validate the event sequence. There is also a state director module which functions as a virtual camera operator in the animation so that the camera movement will correspond to cinematographic principle. Camera operator holds a role in the calculation of camera position based on x, y, and z-axes, as well as in the smoothing of camera placement coordinates. The output of this simulation is validated using histogram profiling, and from the resulting histogram, it will be concluded whether the camera placement is suit according to a directors style.

Key words:
Fuzzy Logic, Machinima, Camera Positioning, State Director, Event Selector

1. Introduction

In recent years, the use of computers has been improved in all industries. This includes the use of computers in creative industries, such as the development of a game, animation, and cinematography. Nowadays, the use of 3D technology is expected to be at a higher level. Game and animation are getting even closer to the real condition. The virtual world is expected to correspond to the real world. As a result, a directors involvement is demanded more in the development of games and animations. A director, giving life to a cinematic product, occasionally must create a storyboard to visualize the desired idea [1].

One of the technology is machinima. Machima [2] is a technique to use 3D technology to produce a cinematic work in the virtual world. Usually, in the creation of animated films or game, camera placement and movement are done by an animator or a photography director. However, manual placement of a virtual camera in a virtual environment requires numerous modeling and calculations that need to be repeated for each scene. This definitely demands quite a substantial cost and time [3].

Paper [4] discussed that the movie Avatar, directed by James Cameron could be said as a milestone in the birth of film production based on a virtual environment. To produce the movie Avatar, a virtual camera technology was developed to record what the director wanted in the virtual environment. This virtual camera has the functions of a normal camera, but it could be used in the virtual environment.

Cinematography is the arrangement of lighting and camera in recording photographic images for a movie [5]. In order to produce a good movie, cinematographic factors need to be considered; because proper camera arrangements could make a movie more appealing and better correspond to the storyline made. Good cinematography helps the audience to understand the storyline of a movie, whether it is animated or real. Some points to note in cinematography are Camera Angles, Continuity, Cutting, and Composition [6]

There are some main points in cinematography and one of Them is camera placement or usually known as camera angle. Camera Angle [7] is the placement of a camera to take a scene in the movie. Every scene in a movie could be taken from a few angles simultaneously to obtain different perspectives. Some take categories among others are based on taking distance (close up, medium shot, long shot) and based on height (low angle, eye level, high level). It is also possible to take a scene based on a combination of categories, for example, medium shot and a high angle at the same time. Figure 1 shows some kinds of camera angle and their combinations. Close up shot could be categorized into different types of shot, such as Medium Shot, Head and Shoulder Close Up, Extreme Close Up, and Over the Shoulder Close Up (Figure 1c)

Every movie director has his unique style to direct and Take scenes. This style distinguishes a director from another.

Manuscript received August 5, 2018
Manuscript revised August 20, 2018

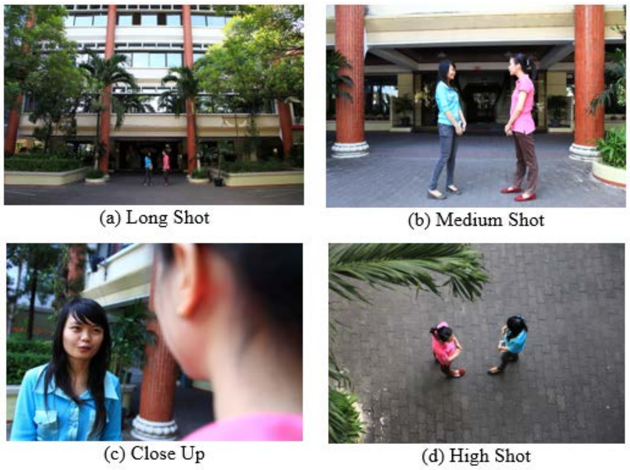


Fig. 1 Camera Angle.

There are some rules that must be obeyed in order to maintain the continuity in object movement and object position [8]. As each director has his style, it is not uncommon for a director to be directly involved in taking a scene to ensure that it is done according to his style. One famous director is Quentin Tarantino [9] with some successful Box Office movies, including Kill Bill, Pulp Fiction, From Dusk Till Dawn, et cetera. Quentin Tarantino is a brilliant student in filmmaking and an expert in using cinematic language in his works to express his thrilling story visually. Every movie Fans would know and said that such is his style. His trademark style is thriller action, darkness added with sadism element and various effects like blood splash. These elements has been a trademark style of Quentin Tarantino.

This research will attempt to apply a directors style to an animation or a game. By applying a directors style, an animation or a game will be more appealing. For instance, a famous game, such as Mario Bros has been made into various versions and researched by many researchers [10] [11].It was a simple side-scroll game with fixed camera placement. If an engine for camera placement is incorporated into this game, this game will have a totally different ambiance. We could imagine if Mario Bros game engines is coupled with RPG camera engine such as that in Zelda or even Real-Time or Action Adventure Camera Engine such as that in Assassin Creed [12] [13].

Figure 2 shows a few different camera placement styles from War Game. The same scene in a game, with a different camera placement style, will give a different impression to the player. Figure 2d is a scene of the entire formation of troops, whereas in Figure 2b and Figure 2c is only a part of it.

This journal will be organized as follows. The second part will discuss previously related researches. The third part will discuss the research method used. The fourth part will discuss the implementation and testing. Finally, the fifth part will discuss the conclusion of this research.

2. Related Works

A number of researches on animation and machinima have been done by some researchers previously. A few references that can be used as a base for this research will be explained.

Paper [14] shows that computer game will be different from movie product because of the real-time characteristic of the game. It is different from a movie, in which a scene could be taken for a number of times, could be stopped and then continued at a later time.

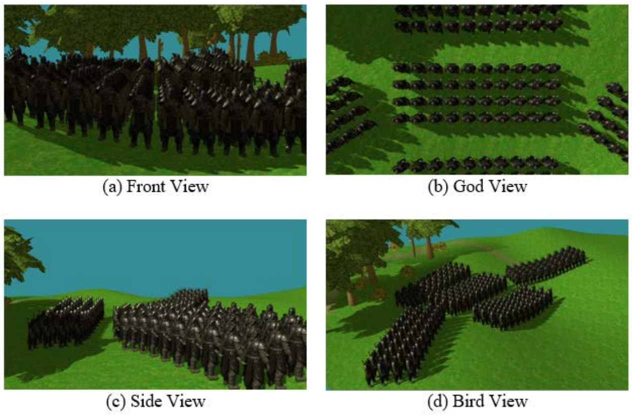


Fig. 2 War Game Point of View

Paper [15] the author uses PSO (Particle Swarm Optimisation) method to solve the problem of VCC (Visual Camera Composition). The researcher uses a hybrid strategy approach to solve the existing problem. The first step taken is to calculate the position of the camera with all of its limiting rules. After that, the position of the camera will be calculated using PSO, based on the determined area. The result of this calculation is the position of the camera, the area or the orientation of the camera, and FOV (Field of View) which will be captured by the camera.

Paper [16] suggests a PSO approach for multi-agent behavior and Paper [17] uses PSO and local regression method to smooth the camera movement. The approach based on The PSO method is also used by some other researchers, whether for calculation of camera placement or for smoothing the camera movement. Paper [18] uses multi objective PSO algorithm for virtual photography. PSO is used to apply a few rules in photography, such as rule of third, horizon line, and POI (Point of Interest).

Paper [19] and Paper [20] are researches which concentrate on second camera placement. The use of two cameras could certainly help in speeding up the process of scene taking, because the first camera will not need to be shifted too far away from its initial position. Paper [19] uses two cameras approach, but the system only calculates the position of the second camera. The first camera is in a static position (unrelated to the second camera), whereas the second camera is placed using behaviour tree method. Paper [21] also suggests research using behaviour tree to suit a single camera to cinematographic principles. The response of the camera to the environment and existing events is stored in smart events. Paper [22] suggests a new semi-automatic language to control a virtual camera in machinima environment. Paper [23] improves the placement of first-person navigation based on input parameters, such as height and weight. The camera used in this system is a fixed camera.

Paper [24] utilizes Support Vector Machine (SVM) to control camera in real time in a storytelling environment. A collection of knowledge in cinematography is trained into SVM to place the camera. The inputs for SVM are scene, virtual environment, and the actor. For this SVM training, based on the knowledge and the rules of cinematography, a film scene will be simulated. The system generated in this research includes several modules, such as the director module, cameraman module, and writer module. All these modules will interact with one another. Paper [25] creates a virtual camera which will predict camera position based on some parameters and the data will be analyzed using Machine Learning, while Paper [26] uses an approach based on Hidden Markov Model to extract cinematographic elements from the real movies.

3. Reseach Method

This research will suggest a technique to incorporate a directors style in automatic camera placement in a virtual environment such as games or animation.

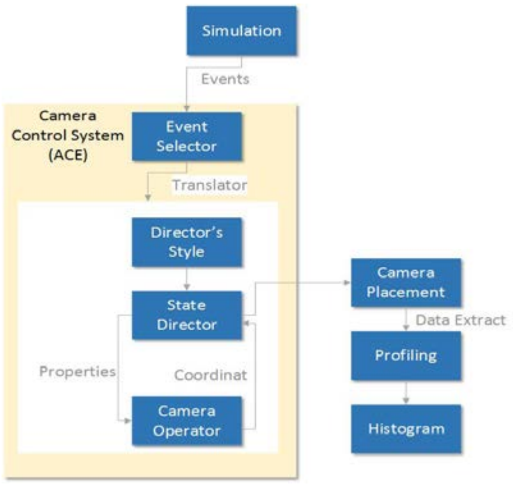


Fig. 3 General System Description

A general description of the system is as seen in Figure 3. Simulation is a game or an animation to test the developed system. Simulation is used to capture existing events and the coordinates as well as the attributes of the player, NPC, or existing obstacles. Event selector will determine valid events to be pass into automatic camera placement block. Camera placement block will determine the position to place the camera based on the style of a director, state director, and camera operator. The resulting position of camera placement will be measured with another system based on profiling to generate a histogram. From the histogram generated, it can be concluded whether the applied style is suitable or not.

Table 1: System Input

Category	Objects	Number	Input
Coordinat	Player 1	6	x,y,z,rx,ry,rz
	Player 2	6	x,y,z,rx,ry,rz
	Camera	6	x,y,z,rx,ry,rz
	Player 1 – Player 2	1	x,y,z,rx,ry,rz
	Player 1- Camera	1	x,y,z,rx,ry,rz
Atribute	Player 1	4	Speed,JP,HP,AP

The number of this system input is 24 which can be divided into two: coordinate system (Player 1, Player 2, and Camera) and players attribute. Table 1 shows the system input

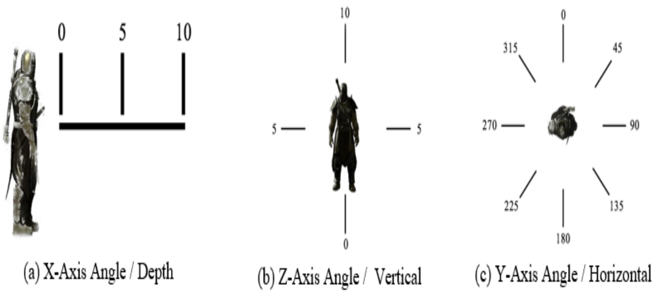
Table 2: Fuzzy Input Membership

Input Variable	Membership Function			
	Range	MF	Type	Control
Speed	0 .. 10	Slow	**	[0,0,1,3]
		Medium	*	[2,5,8]
		Fast	**	[7,9,10,10]
Jump Power	-5 .. 5	Low	**	[-5,-5,4,2]
		Normal	*	[-3,0,3]
		High	**	[2,4,5,5]
HP	0 .. 10	Empty	**	[0,0,1,3]
		Normal	*	[2,5,8]
		Full	**	[7,9,10,10]
Attack Power	0 .. 10	Low	**	[0,0,1,3]
		Normal	*	[2,5,8]
		Full	**	[7,9,10,10]

Table 2 shows the fuzzy input that is use in this experiment. Fuzzy input is not an event resulting from a button press but from a combination of game parameters. For example, the idle event is a combination of low speed, low jump power, normal HP or full HP, and low power.

Table 3: Fuzzy Output Membership

Output Variable	Membership Function			
	Range	MF	Type	Control
Depth	0 .. 10	Close up	**	[0,0,1,3]
		Medium	*	[2,5,8]
		Long	**	[7,9,10,10]
Horizontal	-180 .. 180	Front Left	**	[-180,-180,-160,-110]
		Left	*	[-160,-90,-20]
		Behind	*	[-90,0,90]
		Right	*	[20,90,160]
		Front Right	**	[110,160,180,180]
Vertical	0 .. 10	Low Level	**	[0,0,1,3]
		Eye Level	*	[2,5,8]
		High Level	**	[7,9,10,10]



(a) X-Axis Angle / Depth (b) Z-Axis Angle / Vertical (c) Y-Axis Angle / Horizontal

Fig. 4 Shot Direction Angle

Event Selector is the guidance of the game where a transition from a state or an event will affect another event. For example, a player cannot jump from the idle state, but he can jump when he is in the walking or running state. For each scene, an event selector will be made respectively according to the events allowed for that scene.

State Director is a function to smooth camera movement when there is an instruction for new camera placement based on fuzzy output. Hence, the camera movement is not based on a linear or curve function but it is based on cinematographic rules. For instance, camera movement from an eye-level position behind the player to god-view position; the camera will not be shifted directly based on linear function, but it will be pulled towards the left side of the player and then pulled above the players head according to cinematographic rules. This state director will make simulation results more realistic.

The following are some state directors created in this research:

- CameraIdle
  CameraIdle is the initial state of the virtual camera movement. The virtual camera will get the idle event from a player. Every movement of the virtual camera will start from this state.
- CameraBehindFollow
  The virtual camera will shift towards the position behind the player with middle distance.
- CameraBehindLong
  The virtual camera will initially shift towards the position behind the player with middle distance and then it will be pulled further behind with long distance. This movement could be said as from middle shot pulled towards a long shot.
- CameraHighAngle
  The virtual camera will first rotate to players left side towards the front of players head, Afterwards it will be lifted up to take the scene in a high angle shot tending to god shot right above players head.
- CameraTrunkShoot
  The virtual camera will initially be rotated to players left side, then it will be brought above players head (to show the general view) and directly moved to close up the second player from a low-level perspective of the first player.
- CameraCloseUp
  The virtual camera will shift to the front of the second player, leaning to the left.
- CameraLeftFollow
  The virtual camera will shift to the players left side with middle distance.
- CameraLeftLong
  The virtual camera will initially shift to players left side with middle distance, then it will be pulled behind for a long shot.
- CameraToLeft
  The virtual camera will rotate to players left side.

Camera Operator is an important part of this system, where the fuzzy function for camera placement is placed. Table 3 is the fuzzy output and the membership values used. The three variables on the fuzzy values are a representation of photographic or cinematographic knowledge which will be converted in a 3D game engine into x, y, and z-axes.

Profiling is a process to calculate the position of camera placement based on fuzzy approach. This process has been done in the authors previous research. This process will use the existing shot types approach, which are Follow Shot, Close Up Shot, High Angle Shot, Low Angle Shot, and Trunk Shot. From these five shots approach, a histogram will be generated to determine whether a camera placement has been suited according to a directors style.

3.1 Camera Movement Smoothing

In this research, a fuzzy approach is used for camera placement. As a consequence, when there are a change in fuzzy input value, the camera placement coordinates (x, y, and z) will be affected as well. To ensure that the camera placement is comfortable for viewing and stable, a threshold to shift the virtual camera coordinates on an x-axis, y-axis, and z-axis based of the degree or the radius of the shift will be used. If the degree of the shift is less than the threshold, then the virtual camera position in that quadrant will not be shifted. The threshold value used in this case is 5 degrees.

4. Implementation And Experimental Result

In this research, a simple game is designed using a 3D engine to simulate the system developed. An analysis of the complexity performance of the scenes and characters made in this game will be presented. All experiment is done in a computer environment with the following specifications: Core I7-3530QM CPU @2.40GHz, 16GB RAM, and NVIDIA GeForce GT650M, RAM 2GB video memory.

4.1 Character and Scene Setup

In this experiment, there are three scenes that can be played. The scenes consist of two outdoor scenes (desert and rocky mountain) and one indoor scene (warehouse). Each scene has its differing characteristics and objectives. The complexity of each scene can be seen in Table 4, which includes some of objects, triangles, and vertices

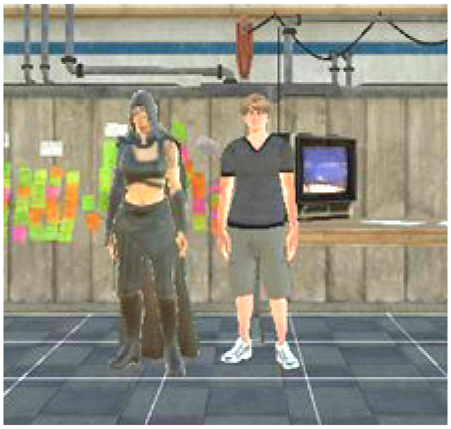


Fig. 5 Simulation Character

There are two main characters (Figure 5). The playable character on the right has 11 animations and the enemy character (NPC) on the left has three animations. The complexity of the characters based on the constituting number of objects, triangles, and vertices can be seen in Table 5.

Table 4: Scene Complexity

Scene	Objects	Triangle	Vertices
Desert	120	55K	40K
Warehouse	720	629K	451K
Rocky Mountain	298	94K	85K

Table 5: Character Complexity

Scene	Triangle	Vertices
Playable Character	35K	21K
NPC	8970	6154

4.2 Scene 1: Desert

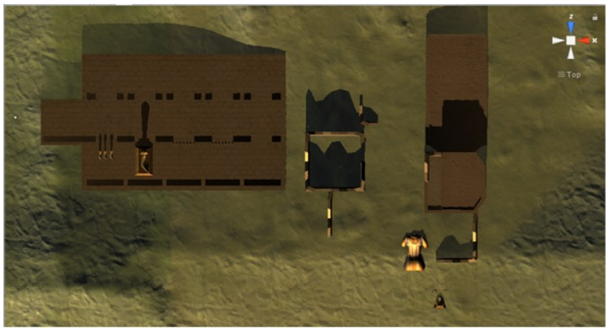


Fig. 6 Top View of Desert Scene

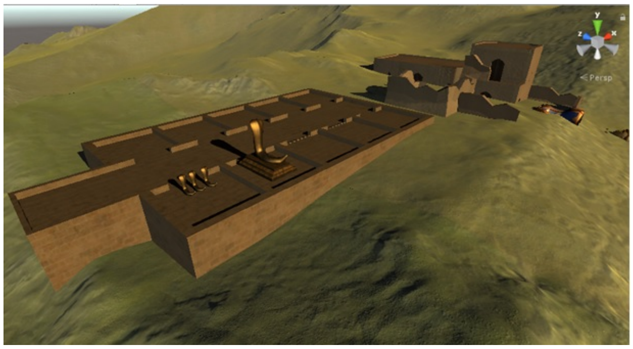


Fig. 7 Perspective View of Desert Scene

In the beginning, the main character is in a desert. The Player can walk and run, as well as jump over the obstacles found in the desert towards the ruins. When the player reaches the ruins, the player will suddenly pass out as he was attacked by a mysterious person. The objective of this game is to pass the obstacles to reach the goal that is the ruins. Figures 6 and Figure 7 is shown the desert and the ruins that is a part of this scene.

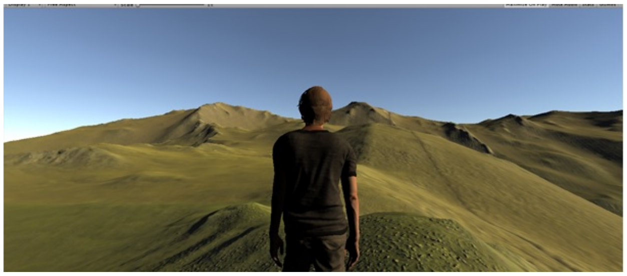


Fig. 8 The Begin of Desert Scene

Figures 8 is shown the beginning scene of the game with desert background. Player has to walk and run to find the ruins in the desert. Figure 9is shown that the player has reached the ruins and his task is to jump over some obstacles in ruins to find the entrance into it. Then in Figure 10 is a scene where the player has found the entrance into the ruins.

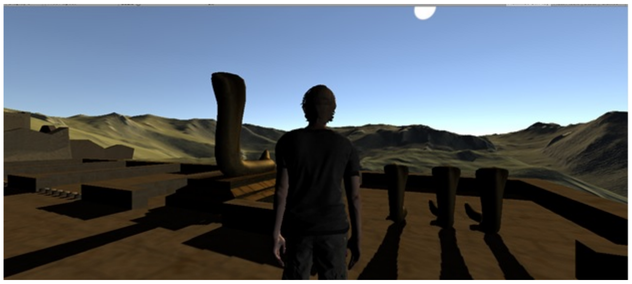


Fig. 9 Ruins of Desert Scene

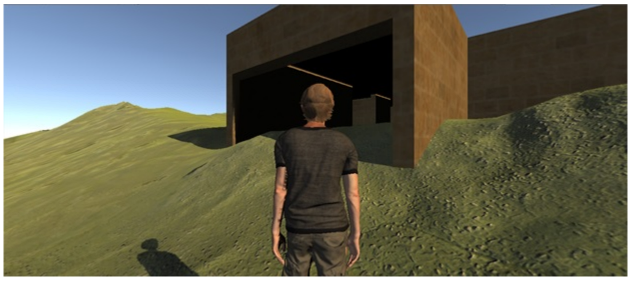


Fig. 10 The Ruins's Entrance

Some events in this scene are idle, walking, running, jumping, and passing out. Figure 15a is the scene where the player jumps and Figure 15b the scene where the player passes out. Event selector for the desert scene can be seen in Figure 11. In the scene selector we can seen that the scene started from an idle state. After an idle state there only a walking state.

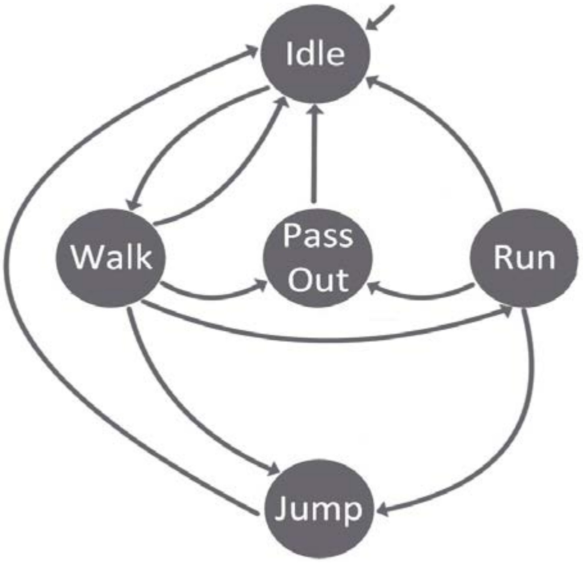


Fig. 11 Event Selector of Desert Scene

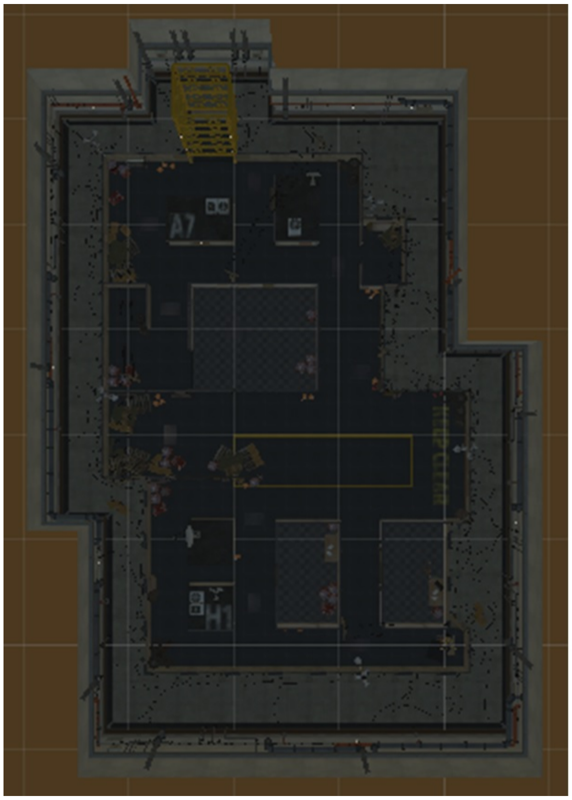


Fig. 12 Top View of Warehouse Area

4.3 Scene 2: Warehouse

When the player regains consciousness, he is already in a warehouse. The Player will search this area and meet someone to talk to. That person will be inside a room which player needs to find. After the conversation ends, player will walk to find the exit to continue the game. As the player exits the room, the player will arrive at a rocky mountain.

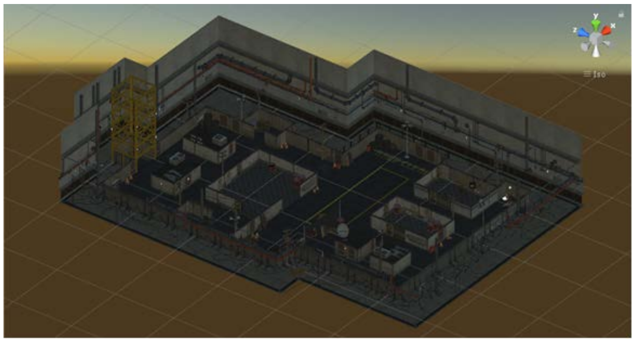


Fig. 13 Perspective View of Warehouse Area

Figure 15c is the camera placement on squatting event and Figure 15d is the resulting camera placement on running event.

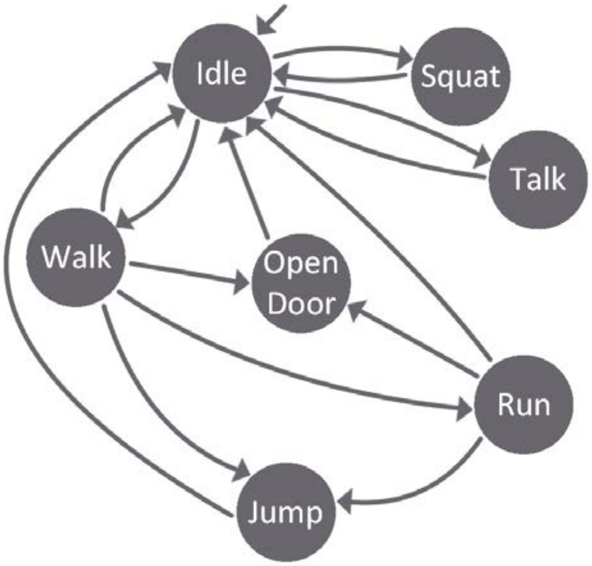


Fig. 14 Event Selector of Warehouse Scene

Shown in Figure 12 and Figure 13 is the background of the first scene. In this warehouse scene, the player will also walk around the room to find the items needed for the next level. There is a scene where there player squats to take the item found. Figure 15e is the scene where the player walk around the warehouse to find the item searched. Figure 15g is a scene where the player is in an idle state, right before he continues his next step.

Some events in this scene are idle, walking, running, jumping, squatting, opening door, and talking to the second character. Figure 14 is the event selector for the warehouse scene.

4.4 Scene 3 : Rocky Mountain

In this scene, the player will be at a rocky mountain and walking to pass the obstacles until he finds his enemy. When the player faces his enemy, they will be attacking each other until the player dies and look towards the enemy attacking him. Shown in Figure 17 and Figure 18 is the playing area for scene 3.

Figure 16 is the event selector for the rocky mountain scene. Some events in this scene are idle, walking, running, jumping, dead, and attacking or fighting.

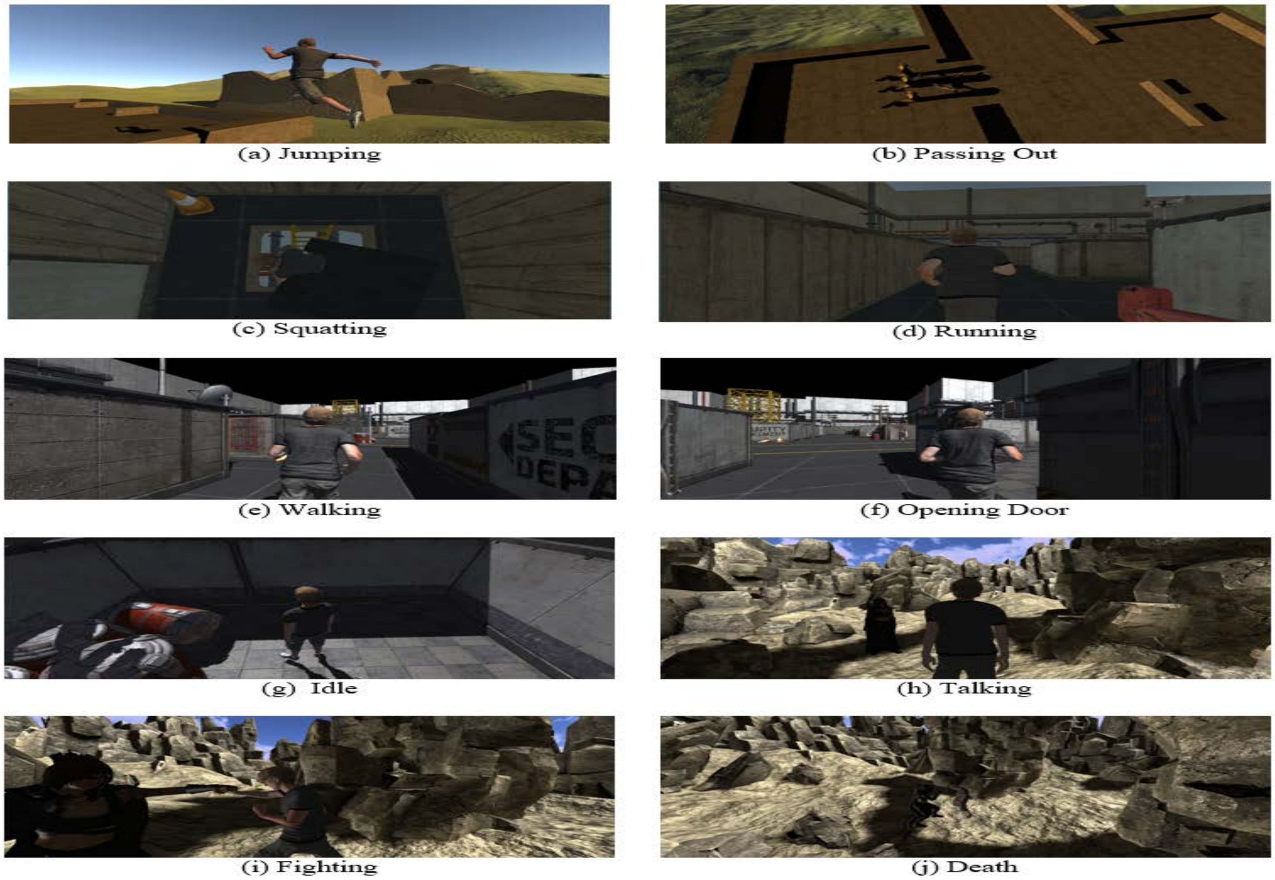

(a) Jumping
(b) Passing Out
(c) Squatting
(d) Running
(e) Walking
(f) Opening Door
(g) Idle
(h) Talking
(i) Fighting
(j) Death

Fig. 15 Event Scene

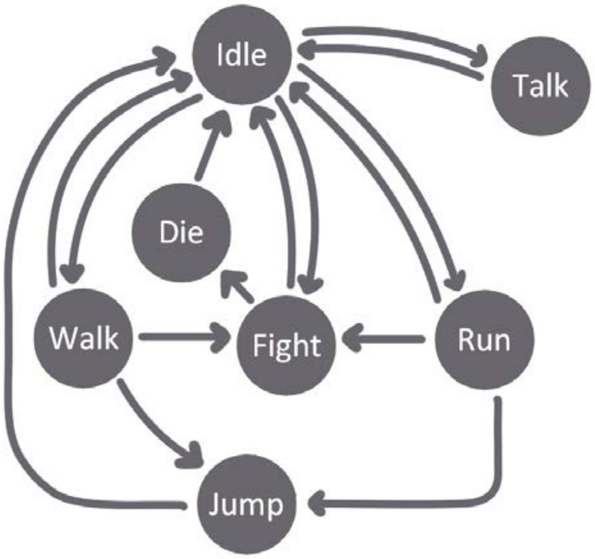


Fig. 16 Event Selector of Rocky Mountain Scene

Figure 15h is the scene where the main character meets the second character that is his enemy and talks to another before attacking. Figure 15i is the fighting scene between the two characters, whereas Figure 15j is the scene where the player loses the fight and dies.

Fig. 17 Perspective View of Rocky Mountain Scene

Fig. 18 Rocky Mountain Scene

4.5 Experiment Result

This experiment uses two different styles. One style is called dynamic style, where the camera position will keep changing according to a directors style; whereas the other style is called static style, where the camera position will not change.

Table 6: Scene Output

Scene	Dynamic Style Frame	Static Style Frame	Events
Desert	435	521	5
Warehouse	618	383	7
Rocky Mountain	465	425	7

Table 6 shows the results of the game simulation based on frame and events.

5. Experiment Result

To prove whether the position of camera placement is suited according to the intended directors style, an area plot and a histogram will be generated for each scene. Just as what has been done in writers previous research, it is stated that if the resulting mode value is on the right side of the value of one (1), then the camera placement is already suited according to the intended style.

Figure 19 is shown the area plot and histogram for Scene 1: Desert. It is stated there that the mode value is between the value of 1.2 and 1.3. Figure 21 is the resulting area plot and histogram for Scene 2: Warehouse, where it is shown that there is a greater number of scenes and the mode value of the histogram is between the value of 1.3 and 1.4. In addition, in Figure 23 is the area plot and histogram from the last scene (rocky mountain). It is shown that the mode is between the value of 1.2 and 1.3.

6. Conclusion And Future Work

This journal introduces an approach for automatic camera placement in a virtual environment based on a directors style using a fuzzy logic approach. The use of an event selector in this research significantly helps in creating the simulation because it can eliminate event sequences that do not correspond to cinematographic principles. State Director make the movement of the virtual camera not suddenly but with a transition based on cinematography principles. Camera movement smoothing based on threshold substantially helps in creating finer animation as values constantly change in real time.

Based on the result of the profiling histogram, it can be concluded that a directors style can be incorporated into automatic camera placement. However, it is expected that in the future, researches involving more events with obstacles and constantly or abruptly moving characters can be done.

Acknowledgments

The author would like to express gratitude to colleagues at Electrical Engineering Department, Electrical Technology Faculty, Institut Teknologi Sepuluh Nopember, for their assistance and advice given to finish this research paper. The author would also like to thank the students of Sekolah Tinggi Teknik Surabaya, members of Business and Multimedia Centre of Sekolah Tinggi Teknik Surabaya, as well as Focaloid (Photography and Videography Society) of Sekolah Tinggi Teknik Surabaya, for their help in preparing the data and the simulation used in this research.

References

[1] J. Hart, The Art of the Storyboard A Filmmaker's Introduction, Second Edition. 2007.

[2] J. I. Hugh Hancock, Machinima for Dummies. Willey Publishing, 2007.

[3] R. Ranon, L. Chittaro, and F. Buttussi, "Automatic camera control meets emergency simulations : An application to aviation safety," Computers And Graphics, vol. 48, pp. 23 – 34, 2015.

[4] J. Bennett and C. P. Carter, "Adopting virtual production for animated filmaking," in 7th Annual International Conference on Computer Games, Multimedia and Allied Technology (CGAT 2014) (E. Prakash, ed.), (Singapore), March 2014.

[5] J. V. Mascelli, The Fives Cinematography : Motion Picture Filming Techiniques. Silman James Press, 1965.

[6] B. Brown, Cinematography Theory and Practice : Image Making for Cinematographer ,Directors and Videographers. Focal Press, 2012.

[7] D. Arijon, Grammar of The Film Language. Hasting House Publisher, 1976

[8] K. Sullivan, G. Schumer, and K. Alexander, Ideas for the Animated Short with DVD: Finding and Building Stories. Focal Press, 2008.

[9] M. Pratt, How to Analyze the Films of Quentin Tarantino. Essential Critiques Set 1 Series, ABDO Publishing Company, 2011.

[10] A. Summerville, J. R. H. Mari˜no, S. Snodgrass, S. Onta˜n´on, and L. H. S. Lelis, "Understanding mario: An evaluation of design metrics for platformers," in Proceedings of the 12th Internasional Conference on the Foundations of Digital Games, FDG '17, (New York, NY, USA), pp. 8:1–8:10, ACM, 2017.

[11] S. Karakovskiy and J. Togelius, "The mario ai benchmark and competitions," IEEE Transactions on Computational Intelligence and AI in Games, vol. 4, pp. 55–67, March 2012.

[12] M. Miller, Assassin's Creed: The Complete Visual History. Insight Edition, 2015.

[13] P. Davies, The Art of Assassin's Creed Unity. UK: Titan Books, 2014.

[14] H. Nicolas, H. Ralf, and S. Thomas, "A camera engine for computer games: Managing the tradeoff between constraint satisfaction and frame coherence," Computer Graphics Forum, vol. 20, no. 3, pp. 174–183.

[15] P. Burelli, L. Di Gaspero, A. Ermetici, and R. Ranon, "Virtual camera composition with particle swarm optimization," in Smart Graphics, pp. 130–141, Springer, 2008.

[16] H. Junaedi, M. Hariadi, and I. Purnama, "Multi agent with multi behavior based on particle swarm optimization (pso) for crowd movement in fire evacuation," in Intelligent Control and Information Processing (ICICIP), 2013 Fourth International Conference on, pp. 366–372, 2013.

[17] D. Prima, M. Hariadi, I. K. Purnama, and T. Usagawa, "Virtual camera movement with particle swarm optimization and local regression," International Review on Computers and Software (IRECOS), vol. 11, no. 9, 2016.

[18] W. Barry and B. J. Ross, "Virtual photography using multi-objective particle swarm optimization," in Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, GECCO '14, (New York, NY, USA), pp. 285–292, ACM, 2014.

[19] A. Fanani, D. Prima, B. Java, E. Suryapto, M. Hariadi, and I. Purnama, "Secondary camera movement in machinema using path finding," in Technology, Informatics, Management, Engineering, and Environment (TIME-E), 2013 International Conference on, pp. 136–139, June 2013.

[20] D. Prima, B. Ferial Java, E. Suryapto, and M. Hariadi, "Secondary camera placement in machinema using behavior trees," in QiR (Quality in Research), 2013 International Conference on, pp. 94–98, June 2013.

[21] D. Markowitz, J. T. Kider, A. Shoulson, and N. I. Badler, "Intelligent camera control using behavior trees," in Motion in Games (J. M. Allbeck and P. Faloutsos, eds.), (Berlin, Heidelberg), pp. 156–167, Springer Berlin Heidelberg, 2011.

[22] W. Hu and X. Zhang, "A semiautomatic control technique for machinima virtual camera," in Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on, vol. 1, pp. 112–115, 2012.

[23] L. Terziman, M. Marchal, F. Multon, B. Arnaldi, and A. Lecuyer, "Personified and multistate camera motions for first-person navigation in desktop virtual reality," Visualization and Computer Graphics, IEEE Transactions on, vol. 19, no. 4, pp. 652–661, 2013.

[24] E. E. S. d. Lima, C. T. Pozzer, M. C. d'Ornellas, A. E. M. Ciarlini, B. Feij, and A. L. Furtado, "Support vector machines for cinematography real-time camera control in storytelling environments," in 2009 VIII Brazilian Symposium on Games and Digital Entertainment, pp. 44–51, Oct 2009.

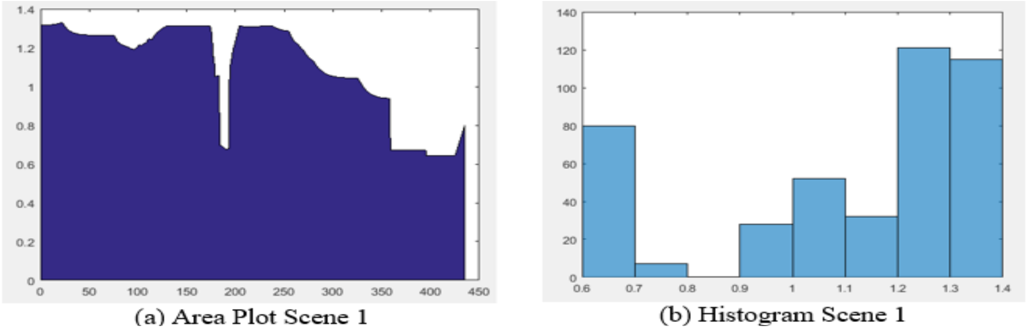
[25] P. Burelli and G. N. Yannakakis, "Adapting virtual camera behaviour through player modelling," User Modeling and User-Adapted Interaction, vol. 25, pp. 155–183, June 2015.

[26] B. Merabti, M. Christie, and K. Bouatouch, "A virtual director using hidden markov models," Computer Graphics Forum, vol. 35, no. 8, pp. 51–67.

Hartarto Junaedi earned his Bachelors degree in IT from Sekolah Tinggi Teknik Surabaya, Indonesia, in 2000, and Masters degree in IT from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 2003. He started his doctorate study at the Graduate School of Electrical Engineering, ITS, in 2012. He is a faculty member at the IS Department, Sekolah Tinggi Teknik Surabaya. His research interests are simulation, machinima, game technology, and cinematography. (e-mail: hartarto12@mhs.ee.its.ac.id, hartarto.j@gmail.com)

Mochamad Hariadi received the B.E. degree in Electrical Engineering Department from 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 a staff of Electrical Engineering Department Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. He is the project leader in joint research with PREDICT JICA project Japan.(e-mail: mochar@te.its.ac.id).

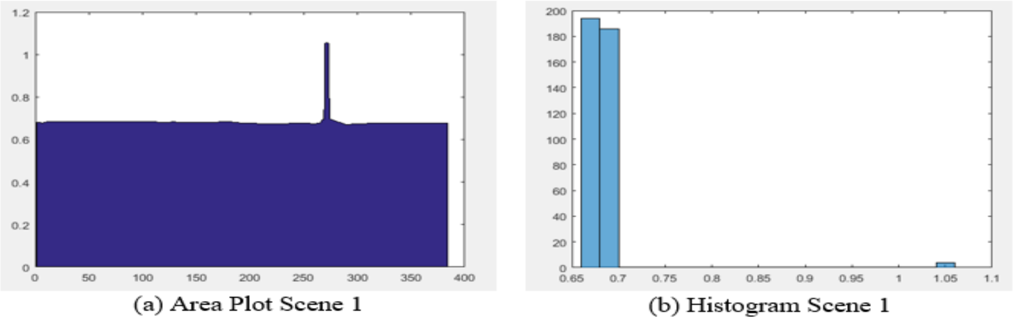
I Ketut Eddy Purnama earned his Bachelors degree from the Electrical Engineering Department at Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 1994. Then he earned his Master of Technology degree from Institut Teknologi Bandung,Bandung, Indonesia in 1999, and PhD degree from University of Groningen, the Netherlands, in 2007. He is the Head of Computer Engineering Department Institut Teknologi Sepuluh Nopember, Surabaya, Currently, he is a staff of Computer Engineering Department Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. (e-mail: ketut@te.its.ac.id).



(a) Area Plot Scene 1

(b) Histogram Scene 1

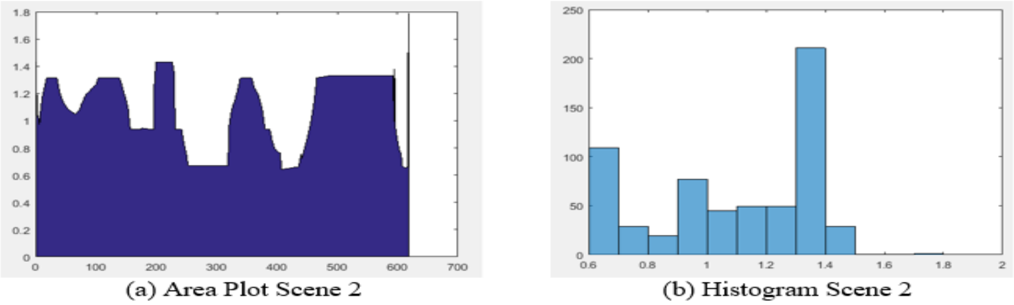
Fig. 19 Area Plot and Histogram for Scene 1 Dyamic Style



(a) Area Plot Scene 1

(b) Histogram Scene 1

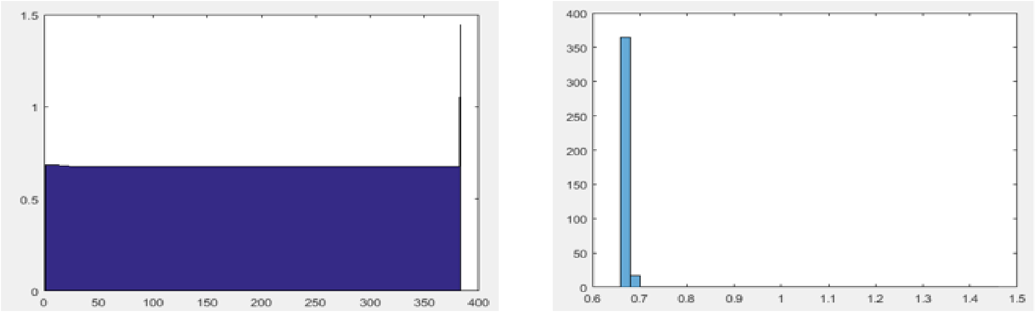
Fig. 20 Area Plot and Histogram for Scene 1 Static Style



(a) Area Plot Scene 2

(b) Histogram Scene 2

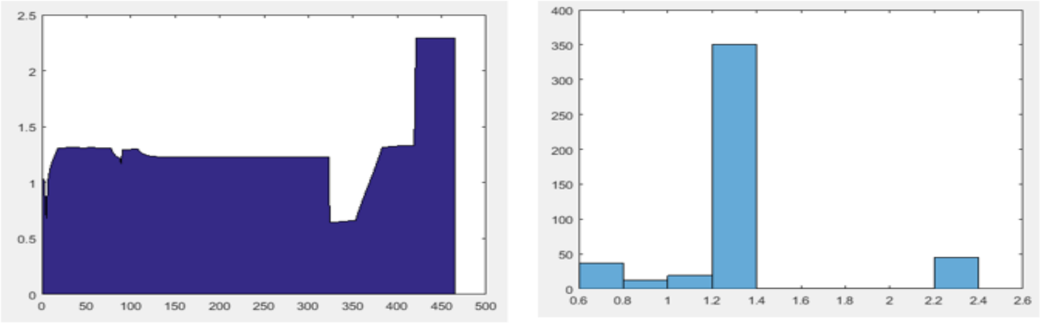
Fig. 21 Area Plot and Histogram for Scene 2 Dyamic Style



(a) Area Plot Scene 2

(b) Histogram Scene 2

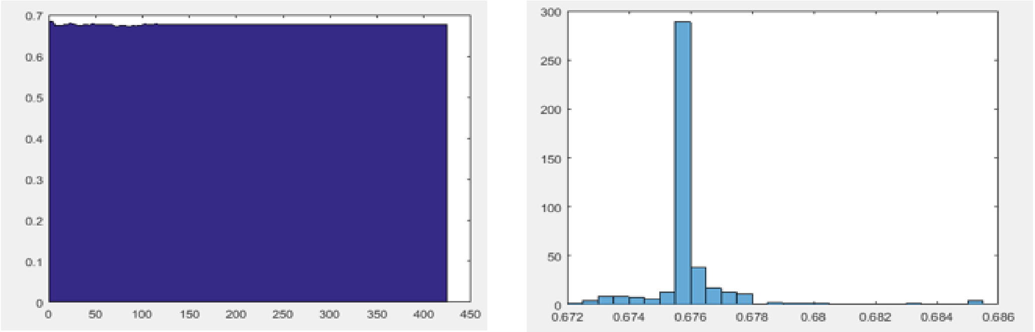
Fig. 22 Area Plot and Histogram for Scene 2 Static Style



(a) Area Plot Scene 3

(b) Histogram Scene 3

Fig. 23 Area Plot and Histogram for Scene 3 Dynamic Style



(a) Area Plot Scene 3

(b) Histogram Scene 3

Fig. 24 Area Plot and Histogram for Scene 3 Static Style