Virtual Camera Control

by Ahmad Zainul Fanani

Submission date: 06-Apr-2020 10:55AM (UTC+0700)

Submission ID: 1290673373

File name: 2_9474-Journal_IRECOS.pdf (1.08M)

Word count: 5226

Character count: 26427

International Review on Computers and Software (I.RE.CO.S.), Vol. 11, N. 6
ISSN 1828-6003 June 2016

Virtual Camera Control Based on Cinematographic Rules of Camera Shot and Camera Angle

Ahmad Zainul Fanani[1, 2], Supeno Mardi Susiki Nugroho[1], Mochamad Hariadi[1]

Abstract – *This research is focused on developing automatic virtual camera control in 3D video game to support a filmmaking based on machinima. Rule-based virtual camera control defines the camera position and the angle. A virtual camera is developed by representing camera shot and camera angle since it can control its position and rotation automatically. Knowledge acquisition for camera shot and camera angle is conducted by simulating virtual camera to the character based on rules of camera shot which are close-up, medium-shot, and long-shot, and rules of camera angle which are eye-level, low angle and high angle. The constrained areas are represented by defining paths. The paths have the function of track references to define position and rotation for the virtual camera, so the virtual camera can capture character vertically, horizontally, and diagonally. Data of constrained areas are representation of camera shot and camera angle rules. The data are used as knowledge base and rules in the term of perspective packages. The virtual camera is generated using Genetic algorithm by randomizing a number of position and rotation value as initial population, then the system calculates the value to find ideal virtual camera which can fulfil the rules in perspective packages. An implementation of the research is conducted by developing a game in which the automatic virtual camera is generated to produce a short animation with duration of 120 seconds. The rules of transition are implemented using different cut as type of transition in every 4 seconds, which means that there are 30 cut to produce. The result shows that the method used in this research reflects the expectation. All the 30 cuts produced in the game are captured using automatic virtual camera. With frame rate at 40-60 fps, the virtual camera is generated in 2 to 16 generations, with duration from 16 to 37 milliseconds.*

Keywords*: Machinima, Virtual Camera Control, Rule-Based, Knowledge-Based Systems*

Nomenclature

ECU/XCU	Extreme close-up
BCU	Big close-up
CU	Close-up
MCU	Medium close-up
MS	Medium shot
MLS	Medium long shot
LS/WS	Long shot
VLS/WLS	Very long shot
ELS/XLS	Extreme long shot
EL	Eye level
HA	High angle
LA	Low angle
BE	Bird's eye
FOV	Field of view
$p_n(x)$	Interpolation polynomial
$p_1(x)$	Interpolation polynomial of one degree
$A(x,y,z)$	Camera position for Euclidean 3D system
$A'(x',y',z')$	Transformation value of coordinate x, y and z axis
R_x, R_y, R_z	The matrix Euclidean transformation for x, y and z axis

I. Introduction

This research is focused on developing automatic virtual camera control in 3D video game to support a filmmaking based on machinima. Machinima, which comes from the words of *machine* and *cinema*, is a real-time filmmaking which uses virtual 3D environment that in general is 3D video game [1]-[3]. Virtual camera in machinima needs controlling technique as the filmmaking in real world which uses camera and related techniques to capture a scene in a good perspective.

Automatic camera control is an approach to control the virtual camera in machinima. Automatic camera control uses photography and cinematography techniques to define a perspective in capturing a good picture. Defining formula of camera control technique and camera path based on computation is an on-going research [4], [5]. An automatic virtual camera control, designed precisely, will result in a picture with precise, effectively and efficient perspective. Automatic virtual camera control is aimed to define a framework to control dynamically the virtual camera movement and the virtual environment which are unpredictable when searching a set of visual property desired [6].

 DOI: 10.15866/irecos.v11i6.9474

There are several methods used for automatic virtual camera control, such as optimization-based, rule-based, and constraint-based. Rule-based is the method used in this research.

II. Related Works

Bares et al [7] have used the optimization-based method by proposing a complete searching based on discretization searching area. *CONSTRAINTCAM* is a system constraining partial satisfaction to camera control by identifying the inconsistency through a construction of constraint pair graphics which is not compatible.

A similar principle is used for defining visual composition problems, such as defining the static camera position, as pure optimization process using genetic algorithm. A set of huge properties is used, including spatial explicit relation, partial and full occlusion, and size [8]. Constraint-based method is used by making camera path defined by declaring a set of object's property which is a target of shot, including vantage angle, frame and the size of object [9].

CAMDROID controls virtual camera using constrained-optimization method by defining behaviour of virtual camera in the term of function and constraint to camera parameters [10]. Path finding technique is used with A* method to control camera based on non-player character movement, so the camera can capture the character movement avoiding a collision with objects in its surrounding [11].

Rule-based is also used as a method to build automatic virtual camera control by defining a set of rules for optimizing the virtual camera control. Rule-based is used to represent a knowledge domain by defining a set of production rules which refers to "situation-action rules" or "if-then rules" [12].

A model for filmmaking production systems called DMP (Digital Movie Producer) using knowledge and rule-based reasoning is developed to create digital motion picture to decide the temporal order of video clip through the rule-based approach [13]. DMP is a production system based on inputted screenplay which is interpreted automatically into motion picture.

The architecture of DMP contains virtual director, knowledge base, inference engine, and movie player.

The knowledge base contains knowledge of objects, color, lighting, scene, shot, and spatial-temporal knowledge. The inference procedure consists of premises divided into two talk, silence, actor A or B talks, while the actions are break into a set of basic shot, such as two-shot and full shot, profile-shot and close up, over-the-shoulder shot, point of view shot and close-up angular shot and close up. Drucker et al [14] have developed a system for procedural camera movements called CINEMA system. The systems is divided into two majors parts: the first one is the database containing information about objects, positions over time, and events over times, while the second part is a parser that accepts and interprets user commands.

The procedural interface is developed based on a set of primitive function, which are a set of commands for moving the camera, a set of command for inquiring about the current camera state, and a set of mathematical routines for manipulating the values returned from the other functions. In this research, automatic virtual camera control is developed using rule-based method by constraining cinematographic rules of camera shot and camera angle to control camera position and rotation.

Camera shots and camera angle rules are implemented in order to capture a perspective which fit with standard in film industry. Knowledge-based system is used to develop an automatic virtual camera control system.

The rules of camera shot and camera angle are used as knowledge in program by defining perspective packages containing data of ideal camera's position and rotation.

The perspective packages are selected by random generation and implemented to adjust the position and rotation of the virtual camera which follows the actor as the game is played. The inference process of ideal camera is generated by randomizing a number of camera's position and rotation based on constrained-areas in order to search a camera which is fit with defined rules. In this case, Genetic algorithm is used to randomize and searching the ideal camera.

III. Chinematography

Camera shot records an action from a perspective in a moment which displays a unit of photography with certain characteristics including person, place, or moving pictures from certain distance and angle [15] [16].

Camera shot consists by various point of views, which are:

1) Extreme close-up/ECU/XCU is displaying a picture in a detail way, the frame focuses on an aspect of subject, such as eyes, mouth, ears, or hands;
2) Big close-up/BCU that displays a picture of human face that occupies as much of the frame as possible, and shows the key features of eyes, nose, and mouth;
3) Close-up/CU that displays a picture of human face that may cut off the top of the subject's hair, while the bottom of the fame can begin anywhere just below the chin;
4) *Medium Close-up*/MCU that displays a picture of human starting from the chest up to the top of the head;
5) *Medium shot*/MS displays the human figure at or below the waist;
6) *Medium long shot*/MLS displays ¾ of subject's body in one frame;
7) *Long shot*/LS/WS displays the full body of subject in one frame;
8) *Very long shot*/VLS/WLS displays the full body of subject with the environment. The environment within the film space fills much of the screen.
9) *Extreme long shot*/ELS/XLS displays the environment within the film space by encompassing a wide and deep field of view.

Fig. 1 shows the various points of view based on camera shot.

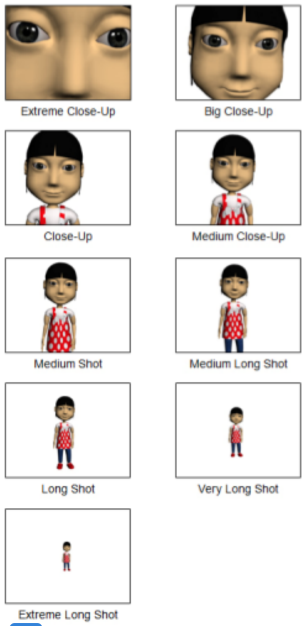

Fig. 1. Different points of view based on camera shot

Camera angle is related to the camera adjustment of the angle to the subject. The types of camera angle are [17]:

1) *Eye-level*/EL that positions the camera near the eye level of the subject;
2) *High angle*/HA positioning the camera above the subject and pointing down.
3) *Low angle*/LA positioning the camera below the subject and angled upward;
4) *Bird's eye*/BE positioning the scene to be seen from above.

Fig. 2 shows the illustration of the various camera angles.

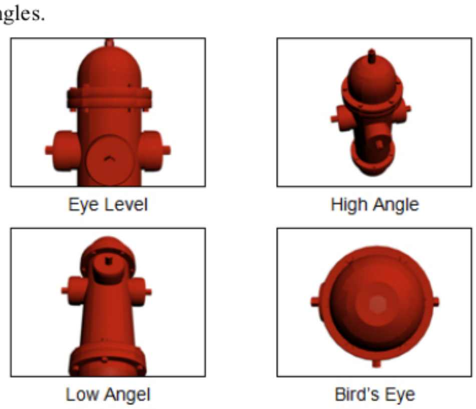

Fig. 2. Point of view based on camera angle

IV. Discussion

In this research, machinima technology is implemented by developing a game to product a short animation in which an automatic virtual camera control is created. The animation produced in this research is created by real-time rendering based on interactivity between user and game. The movement of character in the game controlled by user is the references in defining the ideal position and rotation for camera, and it is rendered automatically into video format.

Rule-based method is used to define the camera position and rotation. Camera shot and camera angle rules are used as knowledge base containing perspective packages to control the smart camera. The perspective packages are defined by the combination between camera shot consisting in CU, MS, LS, and camera angle consisting in EL, HA, LA. The combination of parameters of camera shot and camera angle results as perspective packages, such as LS-EL, LS-HA, MS-LA, and others packages.

The perspective packages then are selected by random generation which is constrained to cinematographic rules in order to arrange the sequences of the perspective packages. The camera is designed to control its position and rotation in capturing the character movement based on rules of camera shot and camera angle, including the rules of clip transition. Fig. 3 shows the diagram of automatic virtual camera control design.

The inference engine is generated by using the genetic algorithm that creates a number of position and rotation values of camera, and then selecting the ideal value based on its fitness.

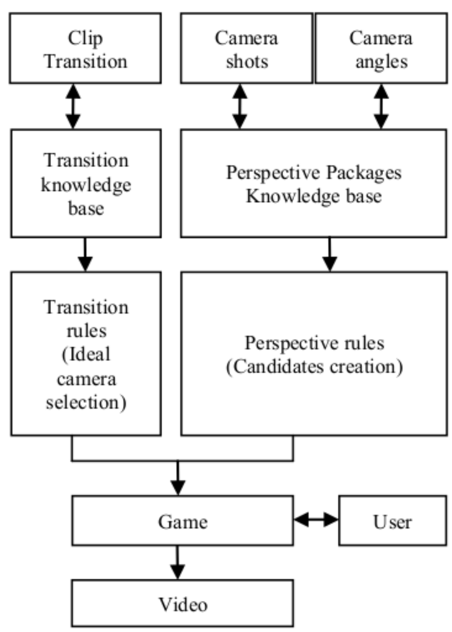

Fig. 3. Virtual camera control design

Unity is chosen as the game engine to develop the game and to implement the algorithm for the automatic virtual camera control. The virtual camera parameters for *field of view* (FOV) and *clipping planes* are defined with a fixed value.

The perspectives which refer to the rules of camera shots and camera angle are defined formerly, and then they are simulated in *Unity*. The simulation data then are used as camera control knowledge base.

IV.1. Rules of Camera Shot

In this research, the type of perspectives in camera shot is limited into CU, MS, and LS.

CU is defined as displaying character from the chest up to the top of the head; MS is defined as displaying ¾ of character's body in one frame; LS is defined as displaying full body of the character in one frame. Fig. 4 shows the illustration of CU, MS, and LS defined in this research. The knowledge base creation for camera shot is conducted by simulating the perspective based on the types of camera shot defined formerly.

The first step is adjusting camera position and the character centred-horizontally, and then adjusting the distance between camera and character based on types of camera shot. Fig. 5 shows the illustration of position and distance adjustment for camera and character.

Fig. 4. Camera shot defined in this research

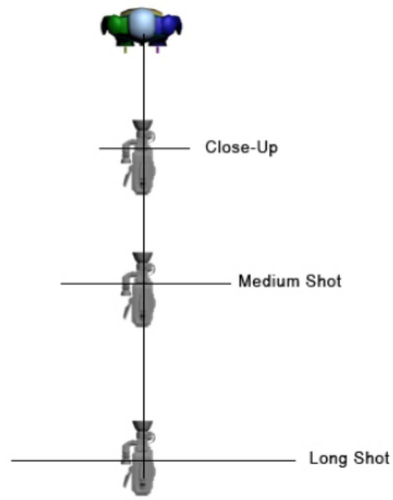

Fig. 5. Adjusting position and distance for camera and character

Maximum and minimum value of camera position and camera rotation to the character in every perspective are the main parameters in knowledge base creation for camera shot.

If n is the registration point, then maximum value of camera position is $+n$, and minimum value is $-n$. Camera rotation value would adjust camera position in its path.

Fig. 6 shows the illustration of maximum and minimum value for MS of camera position and camera rotation to the character. The correlation between position coordinate and rotation value affects the perspective of the picture. In other words, the result of knowledge of camera shot is the data base containing the range of ideal maximum and minimum values to camera position based on its rotation, vice versa. For example, a perspective of close-up from front view can be achieved by adjusting the position of initial X of the camera equal to X position of the character, with 180° for value of Y rotation.

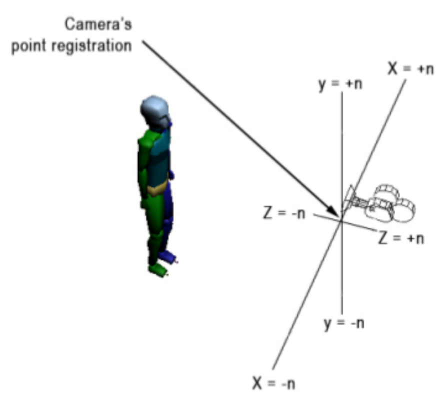

Fig. 6. Maximum and minimum value for camera position and camera rotation to the character

If maximum position of X is +3 from the position of initial X, with rotation value is +80° (or 260°), so the minimum value of X position is -3 from the initial X, with rotation value is -80° (or 100°).

Maximum and minimum distance of camera to the character in defining a certain perspective is simulated, and the result is used as knowledge base to determine ideal value for both of position and rotation. Polynomial interpolation is the approach to define the formula of ideal position and rotation for the camera, with formula:

$$p_n(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 \quad (1)$$

The first and second value of camera position and camera rotation is a straight line, hence the equation of a straight line interpolation polynomial formula is:

$$p_1(x) = a_1 x + a_0 \quad (2)$$

The coefficient of a_1 and a_0 with transcoding process and elimination would be:

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0} \quad (3)$$

and:

$$a_0 = \frac{x_1 y_0 - x_0 y_1}{x_1 - x_0} \quad (4)$$

The camera position and camera rotation can be defined from camera point among 2 camera points based on interpolation formula above, and finally, a perspective package is defined. The path allows the camera to capture a perspective in its track. Fig. 7 shows the path of camera position and rotation.

IV.2. Rules of Camera Angle

The perspectives in camera angles are divided into EL, HA, and LA for MS and LS, and EL for CU, with character body parts capture definition follows rules of camera shot as shown at Fig. 8.

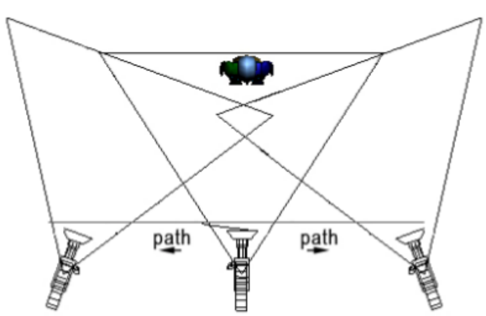

Fig. 7. Paths as a track for positioning and rotating the camera

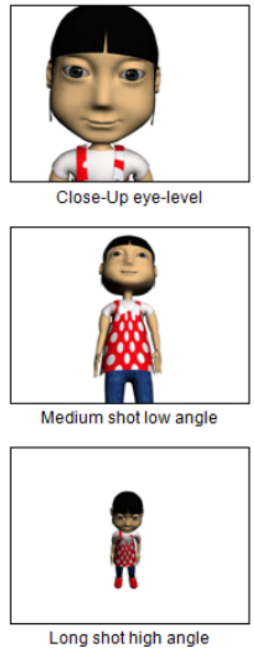

Fig. 8. Camera angle defined in the game

IV.3. Perspectives Packages

The results of path creation, including areas range for camera shot and angle, are transformed into data base of camera shot and camera angle knowledge base, and then rules of camera position and camera rotation are defined as perspective packages.

Table I shows the knowledge base as the representation of camera shot and camera angle, which contains the data of maximum and minimum distance and the rotation between camera and character with character coordinate values for all positions and rotations equal to 0. The knowledge base contains 7 perspective packages of camera position and camera rotation related to the character. The perspective packages are CU-EL, MS-EL, MS-LA, MS-HA, LS-EL, LS-LA, and LS-HA.

The packages are controlled by constraining areas for camera's position and rotation. Fig. 9 shows the illustration of knowledge representation for camera shot and camera angle transformed into areas as perspective packages.

TABLE I
PERSPECTIVE PACKAGES

NO	PACKAGES	POSITION			ROTATION		
		X	Y	Z	X	Y	Z
1	CU-EL	-0.3 - 0.3	0.77	0.33	0	110-255^0	0
2	MS-EL	-0.9 - 0.9	0.5	1	0	105-260^0	0
3	MS-LA	-0.9 - 0.9	0	1	-25	105-260^0	0
4	MS-HA	-0.9 - 0.9	1.2	1	25	105-260^0	0
5	LS-EL	-1.8 - 1.8	0	2	0	100-260^0	0
6	LS-LA	-1.8 - 1.8	-0.8	2	-20	100-260^0	0
7	LS-HA	-1.8 - 1.8	1.2	2	30	100-260^0	0

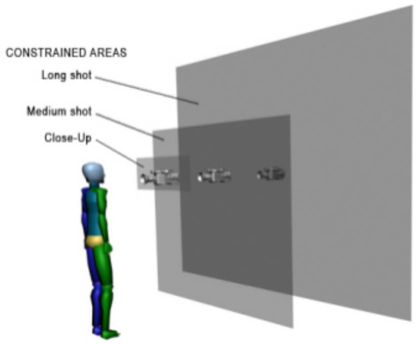

Fig. 9. Knowledge representation for camera shot and camera angle

Position and rotation of the camera are defined based on calculating its value to the character using euclidian 3D; Hence camera point A(X, Y, Z) to the character is:

$$x = r\cos\theta \quad \text{and} \quad y = r\sin\theta \tag{5}$$

The transformation value of $A'(x', y', z')$ to the Z-axis (R_z) is:

$$\begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \\ z' &= z \end{aligned} \tag{6}$$

The equation above can be modelled by matrix transformation as followed [17]:

$$R_z = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{7}$$

By using the technique above, then the matrix transformation value of $A'(x', y', z')$ to the X-axis (R_x) is:

$$R_x = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{8}$$

and the matrix transformation value of $A'(x', y', z')$ to the Y-axis (R_y) is:

$$R_y = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{9}$$

IV.4. Rules of Clip Transition

The cut, a type of clip transition that replaces instantly the picture, is used in this research.

The rule of 4 seconds for each cut is implemented to the automatic virtual camera control. The control camera engine must define a new perspective in less than 3 seconds. The cut is determined based on perspective packages. The flow of cut is determined by randomizing the perspective packages with the conditions that a same package should not be used two times in a sequence, and each package displays a perspective in 4 seconds.

IV.5. Production Rules

The production rules are defined by the term of *if* $\Rightarrow$ *then*. The rules are used to match the random generation of camera position and rotation to the parameters of perspective packages, and to control the sequence of perspective packages.

The production rules are implemented into two phases, which are the creation of candidates and the selection of a candidate as the ideal camera.

The description of the rule for creating candidates is "*If the random values of position and rotation match with one or more perspectives in perspective packages, then select the perspective (s) as candidate (s)*".

The description of rules for selecting an ideal camera is "*If a candidate is different with the previous perspective, then set camera to capture animation; If more than one candidate is different with the previous perspective, then set the first candidate as the camera to capture animation*".

IV.6. Control Camera Engine

Automatic virtual camera control is designed to define its position and rotation in capturing character in game which fits with camera shot and camera angle rules.

The generation of automatic camera is conducted by randomizing perspective packages, and then randomizing camera position and camera rotation. The fitness of camera position and camera rotation value is measured based on the parameters value in perspective packages.

The Genetic algorithm is used to generate the automatic camera.

The procedures and operator of genetic are implemented by creating a population which consists of variables with random value for X, Y, Z coordinates of position and rotation. The population is randomized based on its sequence, and then the crossover is implemented to them. If the random process results more than 2 values that are fit, then the first value is used for automatic.

The generation of the automatic camera is limited in 3 seconds. If the algorithm cannot generate the automatic camera in limited time, then the program uses the previous setting to capture character. Fig. 10 shows the work flow diagram for automatic virtual camera control.

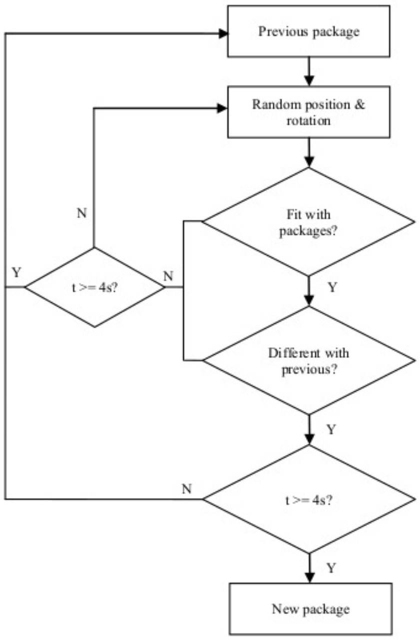

Fig. 10. Perspective package work flow diagram

V. Implementation

The game developed in this research contains a single character with empty plane as virtual environment. The animation for the character consists of standing, running, walking, squatting and jumping. User interaction in controlling the character movement is by using keyboard button and mouse. The game is used to produce short animation movie with a duration of 120 seconds. Cut transition is set in every 4 seconds, so the movie contains 30 cuts. The game is rendered in real-time into video format. The implementation is conducted by producing 2 short animations with a duration of 120 seconds for each animation. Table II and Table III show the results of automatic virtual camera control in producing short animation movie with frame rate at 40-60 fps.

The implementation of producing 2 short animations explained in tables above shows that, all the 7 perspective packages are used in both the 2 short animations, the number of distribution is shown in the Table IV and its chart in Fig. 11.

Minimum number of random for generating camera control is 2 times for both animation 1 and animation 2, with minimum times to generate at 15-16 milliseconds. Maximum number of random for generating camera control is 9 times for animation 1, and 12 for animation 2, with maximum times to generate at 37 milliseconds.

Numbers of time and random of camera control generation are seen in Table V.

The following pictures (Fig. 12) show some snapshots by the short animation which were captured by automatic virtual camera control.

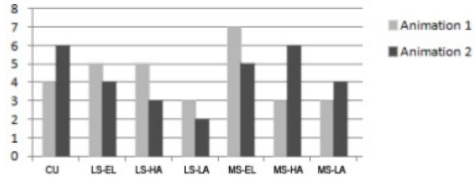

Fig. 11. Chart of the using of perspective packages distribution

Fig. 12. Snapshot of the perspectives resulted from automatic virtual camera control

TABLE II
RESULT OF SMART CAMERA

Cut	Generation		Perspective Packages
	Random	Time (msc)	
1	4	35	LS-HA
2	3	33	LS-EL
3	3	32	MS-LA
4	8	16	CU
5	2	33	MS-HA
6	6	21	CU
7	2	31	MS-HA
8	6	35	MS-EL
9	5	35	LS-EL
10	3	18	CU
11	2	34	LS-HA
12	2	33	MS-HA
13	3	34	MS-LA
14	7	37	LS-HA
15	5	34	MS-LA
16	3	33	LS-LA
17	2	34	MS-EL
18	3	33	LS-LA
19	2	32	MS-HA
20	9	18	CU
21	4	34	MS-HA
22	3	36	MS-EL
23	5	35	LS-EL
24	6	16	CU
25	2	33	MS-HA
26	2	33	LS-EL
27	3	32	MS-EL
28	3	33	MS-LA
29	3	34	MS-EL
30	4	17	CU

TABLE III
RESULT OF SMART CAMERA

Cut	Generation		Perspective Packages
	Random	Time (msc)	
1	3	35	MS-EL
2	2	35	MS-EL
3	6	33	MS-EL
4	4	33	MS-LA
5	2	34	LS-EL
6	2	33	LS-LA
7	2	34	LS-HA
8	2	36	MS-EL
9	12	16	CU
10	6	33	MS-EL
11	4	32	MS-LA
12	4	37	LS-HA
13	5	34	LS-EL
14	4	15	CU
15	3	34	MS-EL
16	3	32	LS-HA
17	3	35	MS-HA
18	2	34	MS-LA
19	3	33	LS-HA
20	2	33	MS-EL
21	2	35	LS-EL
22	3	33	MS-HA
23	4	17	CU
24	2	35	LS-LA
25	4	34	MS-EL
26	2	32	LS-HA
27	3	34	MS-HA
28	5	20	CU
29	2	32	LS-EL
30	3	33	LS-LA

TABLE IV
THE USE OF PERSPECTIVE PACKAGES DISTRIBUTION

No	Perspective Packages	Animation 1	Animation 2
1	CU	4	6
2	LS-EL	5	4
3	LS-HA	5	3
4	LS-LA	3	2
5	MS-EL	7	5
6	MS-HA	3	6
7	MS-LA	3	4
	Total Cut	30	30

TABLE V
NUMBER OF TIMES AND RANDOM GENERATION

Control Camera Generation	Animation 1		Animation 2	
	Min	Max	Min	Max
Times (milliseconds)	16	37	15	37
Number of Random	2	9	2	12

VI. Evaluation

The implementation shows that the using constrained areas for creating perspective packages as knowledge for the systems obtain high success in controlling camera shot and camera angle for virtual camera.

The system controlling the virtual camera takes less than about 37 milliseconds. The perspectives of 60 cuts in the 2 short animations are shot automatically, or the accuracy reaches up to 100%. All of 7 perspective packages recorded as knowledge are used in both animation 1 and animation 2.

VII. Conclusion

The use of rule-based for defining perspective packages, and then transforming them into knowledge for controlling virtual camera is accurate to produce a film based on game.

The rules of camera shot and camera angle which are transformed into perspective packages can support the automatic virtual camera control up to 100%.

The perspective packages can reduce the process of random generation in selecting the ideal position and rotation of the camera, so the time expectation of generating the ideal camera can be achieved in less than one second. This shows that the system can produce the clip sequences effectively while the game is played.

VIII. Future Works

For the future works, this research can be developed by creating more knowledge bases, adding the number of characters, camera movement and collision detection.

The production rules should be able to set a perspective package including the arrangement the clip sequences based on the interpretation of the actions, so the visualization of the animation can be achieved based on the story built in the film.

References

[1] P. Johnson, D. Pettit, *Machinima: The art and Practice of Virtual Filmmaking*, (McFarland, 2012).

[2] J. Ng, *Understanding Machinima: Essayys on Filmmaking in Virtual Worlds*, (Bloomsbury, 2012).

[3] M. Picard, *Machinima: Video Game as an Art Form?*, Proceedings of CGSA 2006 Symposium, Canada, 2006

[4] M. Christie, P. Olivier, J.M. Normand, Camera Control in Computer Graphics, *Computer Graphics forum* Vol. 27, July 2008.

[5] S.M. Drucker, D. Zeltzer, Intelligent Camera Control in a Virtual Environment, *Proceedings of Graphics Interface* 1994, Canada, 1994.

[6] P. Burelli, G.N. Yannakakis, Towards Adaptive Virtual Camera Control In Computer Games, in L. Dickmann, G. Volkman, R. Malaka, S. Boll, A. Kruger, P. Olivier (Eds.), *Smart Graphics*, (Germany: Springer, 2011, 25-36)

[7] W.H. Bares, S. Thainimit, S. McDermott, A model for constraint-based camera planning, *Smart Graphics AAAI Spring Symposium*, Stanford, California, 2000.

[8] P. Olivier, N. Halper, J. Pickering, P. Luna, Visual Composition as Optimisation, *AISB Symposium on AI and Creativity in Entertainment and Visual Art*, 1999.

[9] F. Jardillier, E. Langu'enou, Screen-space Constraints for Camera Movements: The Virtual Cameraman, *Proceedings of the Eurographics Conference (EG 98)*, 1998.

[10] A.Z. Fanani, D.A. Prima, B.B,F. Java, E. Suryapto, M. Hariadi, I.K.E. Purnama, Secondary Camera Movement in Machinema using Path Finding, *International Conference of Technology Informatics Management Engineering and Environment (TIME-E) joined with IEEE*, Bandung, 2013.

[11] S. Drucker, D. Zeltzer, CamDroid: A System for Implementing Intelligent Camera Control, *Proceedings of the 1995 symposium on Interactive 3D graphics*, 1995

[12] V.D. Hunt, *Artificial Intelligence and Expert Systems Sourcebook*, (Springer, 2012).

[13] S. Jinhong, S. Miyazaki, T, Aoki, H. Yasuda, Filmmaking Production System with Rule-based Reasoning, *Image and Vision Computing NZ*, Palmerston North, 2003.

[14] S. Drucker, D. Zeltzer, T.A. Galyean, Cinema: A system for procedural camera movements, *Proceedings of the 1992 symposium on Interactive 3D graphics, ACM Press*, New York, 1992.
[15] C. J. Bowen, R. Thompson, *Grammar of the Shot*, (Focal Press, 2013).
[16] M. Pramoggiore, T. Wallis, *Film: A Critical Introduction*, (Laurence King Publishing, 2005)
[17] B. Mamer, *Film Production Technique: Creating the Accomplished Image*, (Cengage Learning, 2009)
[18] H. Richard, Z. Andrew, *Multiple View Geometry in Computer Vision* Second edition, (Cambridge University Press, 2003).

Authors' information

[1]Department of Electrical Engineering, InstitutTeknologi Sepuluh Nopember, Surabaya, Indonesia.

[2]Departement of Computer Science, Dian Nuswantoro University, Semarang, Indonesia.

Ahmad Zainul Fanani received bachelor and master degree in 1999 and 2001 respectively from Informatic Technic Department STTIBI, Jakarta, Indonesia. Since 2011, he has been studying at the Electrical Engineering of Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia as a doctoral student. He works as a lecturer of Computer Science Faculty, Dian Nuswantoro University, Semarang, Indonesia. His research interest includes computer vision, Game and Multimedia Design.

Supeno Mardi Susiki Nugroho is a lecturer in the department of Electrical Engineering Multimedia Network Department at Institut Teknologi Sepuluh Nopember(ITS), Surabaya, Indonesia. He received his Ph.D and B.Eng. from Electrical Engineering at Institut Teknologi Sepuluh Nopember in 2013 and 1994. He received his master in department of Informatic Engineering at Institut Teknologi Bandung (ITB), Indonesia in 1999. He is the leader of Lab Based Education research about game technology in ITS. His teaching and research interests include computer game simulation, modeling, and optimization.
E-mail: mardi@its.ac.id

Mochamad Hariadi graduated from Electrical Engineering Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia for his bachelor in 1995 and received MSc and PhD degree in 2003 and 2006 respectively from Graduate School of Information Science (GSIS) Tohuku University, Japan. He is currently lecturer of Electrical Engineering Departmen ITS. His research interest include multimedia processing and artificial intelligence. He is IEEE member and IEICE member.

Virtual Camera Control

ORIGINALITY REPORT

16%	10%	9%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Brawijaya Student Paper	3%
2	David Zeltzer. "CINEMA: a system for procedural camera movements", Proceedings of the 1992 symposium on Interactive 3D graphics - SI3D 92 SI3D 92, 1992 Publication	2%
3	sites.google.com Internet Source	1%
4	Submitted to EDMC Student Paper	1%
5	Submitted to Hillgrove High School Student Paper	1%
6	Jinhong Shen, Seiya Miyazaki, Terumasa Aoki, Hiroshi Yasuda. "Intelligent Computer Moviemaker with the Capabilities of Cinematic Rule-based Reasoning", The Journal of the Institute of Image Information and Television Engineers, 2004 Publication	1%

7	plaintheweb.blogspot.com Internet Source	1%
8	www.readbag.com Internet Source	1%
9	Submitted to Queen Mary and Westfield College Student Paper	1%
10	docplayer.info Internet Source	1%
11	www.ors.org Internet Source	<1%
12	Submitted to Utah Education Network Student Paper	<1%
13	Delta Ardy Prima, Mochamad Hariadi, I. Ketut Eddy Purnama, Tsuyoshi Usagawa. "Virtual Camera Movement with Particle Swarm Optimization and Local Regression", International Review on Computers and Software (IRECOS), 2016 Publication	<1%
14	Submitted to University of New South Wales Student Paper	<1%
15	www.coursehero.com Internet Source	<1%
16	www.eventlab-ub.org Internet Source	

<1%

17	Hui Chen. "Gaussian beam scattering from arbitrarily shaped objects with rough surfaces", Waves in Random Media, 7/1/2004 Publication	<1%
18	pure.itu.dk Internet Source	<1%
19	prezi.com Internet Source	<1%
20	Submitted to Emmanuel College Student Paper	<1%
21	sinta3.ristekdikti.go.id Internet Source	<1%
22	resits.its.ac.id Internet Source	<1%
23	www.ultramarine.com Internet Source	<1%
24	Submitted to Doukas School Student Paper	<1%
25	Shinsuke Saito. "Virtual space production based on composition knowledge", Systems and Computers in Japan, 11/15/1998 Publication	<1%

26	repository.ubaya.ac.id Internet Source	<1%
27	ritsumei.academia.edu Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		