Automatic Virtual Camera Control Using A Style Imitation Technique For 3D Game Based Animation

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

[1]*Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Kampus ITS Keputih, Sukolilo, Surabaya 60111, Indonesia*
E-mail: [1,2]ahmad.fanani11@mhs.ee.its.ac.id

[2]*Departement of Computer Science, Universitas Dian Nuswantoro, Jl, Imam Bonjol 205-207 Semarang 50131, Indonesia.*

Abstract

This research aims to develop a model of automatic virtual camera control for 3D animation film making based on 3D game. The model consists of two main parts, which are rules of shot as representation of control camera knowledge, and inference engine to generate virtual camera control in capturing ideal shot. Style imitation technique commonly used to generate music composition automatically, is used to imitate film materials of reference film. Film material are defined in components of number of actors in a shot, type of action, type of control camera, shot duration, and shot sequence. Shots of reference film are analyzed based on these components to build perspective packages of camera control knowledge base. Type of control camera, perspective packages, shot sequence is used as constraint to define ideal shot. Genetic algorithm is use to generate virtual camera control, with data of shot duration of each perspective package is used as parameter of iteration. Experiment is conducted by imitating shots of 3D animation film entitled how to train your dragon 2 produced by DreamWorks Animation, and developing a 3D game in which the proposed model is implemented. Evaluation of the performance of genetic algorithm results 82.2% shots of total number of shots can be generated by the algorithm. The implementation of style imitation is measured by comparing shots produced in this research to shots of reference film based on point of view. Three experts of film and multimedia are asked to value the similarity, and Mean Opinion Score technique is used to measure the expert's opinion. The result shows that in scale 1-5, where score 1 is for poor, and 5 is for excellent, 3D animation produced in this research are similar to shots of reference film at score 3.9.

Key Words: Automatic virtual camera control, Genetic algorithm, Style imitation

1. Introduction

Camera control has an important role to capture ideal view, and to supports the success of messages delivery in every scene. In a graphic computer, virtual camera control which consists of calculation of point of view, movement planning, and editing, is a component of various applications, including data visualization, and virtual walk-through [1]. Camera shot, camera angle, and camera movement are parameters in virtual camera control. In the development of 3D animation film, virtual camera is generally controlled by setting the position, which includes camera distance and camera height, and the rotation manually. Tweening technique is used to control the camera movement, by defining the position and rotation of camera in two keyframes, and program will generate extra frames between those keyframes. Manual technique consumes times in animation production, especially in a scene in which a camera movement is altered, where extra frames generated by program many times give an unexpected result. Automatic camera control can be a solution in 3D animation filmmaking. Artificial intelligence approach can be used to generate a smart camera which can capture an ideal view automatically, and to reduce time consuming in production.

The technology in 3D animation has growth, and involved 3D game as a media to support the filmmaking. The technology is known as machinima, which uses 3D virtual environment in 3D game video for real-time filmmaking [2] [3] [4]. Camera control automation has become a research subject in machinima technology. There are some approaches used to virtual camera control, such as reactive approach which involving computation of direct respons to the chane in image properties without reordering all the searching process; optimization-based approach which controls the camera with optimation technique to the properties of images capturing as destination to be maximized; constraint-based approach which contains framework Constraint Satisfaction Problem (CSP) by proposing a declarative approach to model a higher range of the main non-linear constraint, and reliable technique to solve the problem; constrained-optimization which is a pure optimation technique that allows computation of a solution which has a good probability in which the properties are satisfied in several levels, and results a solution for declarative camera control, including constraint and fitness function, but relatively consumes high computation [5].

In this research, the constrained-optimization approach is used to develop a model of automatic virtual camera control system that supports 3D animation filmmaking based on 3D game. The challenge is to create a smart virtual camera which works real-time in 3D environment game to capture ideal shot automatically. The ideal shot is defined based on cinematography rules, and materials of reference films imitated as knowledge for the system. Imitation of materials based on given ideas is known as style imitation technique, which is commonly used to generate automatic music composition.

Style imitation technique is a music material generation based on given musical ideas, and is a main area in algorithmic composition application [6]. Some methods based on artificial intelligence used in algorithmic composition, such as Markov model, generative grammars, genetic algorithm, knowledge-based systems, and others, use style imitation technique. Style imitation technique is conducted by learning materials in an object, for example certain music material, to be developed for a new material composition generation. Grammar approach with melody-morphing method is used, in which melody A and B is the input, and the others melody between those melodies is interpolated and extrapolated [7]. Markov chain is used by [8] to generate musical transition between two different parts in a simple non-linear music application to stochastically give alternative between two markov chain. Rule-based expert system is used by [9] to develop CHORAL system, a system of melody harmonization in chorale style of J.S. Bach. Style of Bach is also developed by [10] with genetic algorithm for harmonizing the melody.

Style imitation technique is conducted by learning the material in an object, such as certain music material, and then it is developed in the generation of new material composition. In this research, the technique is adopted for automatic virtual camera control, by learning the camera shot technique in certain films as knowledge base to inference the ideal camera placement and camera movement.

2. Related Works

Constraint approach is to make a camera following an object [11] [12] [13]. The use of constraint, such as distance, height, object relatively oriented can be applied in camera. Constraint will affect the camera when the object is moving in a scene. The knowledge-based approach was used by [14] to develop autonomous real-time camera agent, a framework of real-time cinematography system for 3D environment uses a process which consists of four phases, which are selecting narrative element which allows the system to define narrative element selection which is the most relevant to run time; computing director volumes which is the phase to change the chosen narrative element to director volumes by composing viewpoints characteristic area which allows the element picture based on visibility volumes; editing over director volumes which implements the process of multiple filtering to select appropriate director volumes; computing transaction which is a phase to define the transition between director volumes.

Cinematography knowledge which consists of camera parameter, movement, shot characteristic, sequence rules, standard shot, rules of thumbs, and others, is used to define the formalization in the domain interactive narratives. A rule-based system is used to develop an automatic camera control modeled with 7 Degrees of Freedom; 3 for camera position is used for crane shot, dolly shot, and tracking shot; 3 for orientation is to allow pan shot, rolling shot, and tilt shot; FOV can be produced based on the camera movement directly, or by zooming. A number of rules are defined to control virtual camera, such as, rules of one-shot generation, and shot sequence generation [15]. CINEMA system developed by [16] is to control camera movement in a system which contains 2 main parts. The first part is a database containing information of object, including its over time position and event, and the second part is command of user which are restricted to check the database states, and command to query or affect the camera state. A system of partial constraint satisfaction for virtual camera control, called CONSTRAINTCAM, is developed by [17]. Inconsistency is identified by constructing a graphic of a pair of constraints which are not compatible. Optimization global process is used where each configuration becomes a value which represents fitness, exhaustive generate, and evaluation process.

Genetic algorithm is used for visual composition problem of static camera position setting [18]. A set of property which consists of spatial explicit relation between objects, partial and total occlusion, and measurement, is used. Fitness function is a combination of linear-weighted from shot property compliance. Seven camera parameters are given code, and camera population is distributed randomly in a searching area. Each individual of population is evaluated based of objective function. 90% of highest populations survive until next generation, and selection is conducted using binary tournament technique. The reset of 10% are re-generated using random crossover, and/or mutation. Constraint-based approach is used with camera path built by declaring a set of property for objects defined to be shot, including vantage angle, frame and object size [19]. Camera path is modeled as a set of primitive camera movement which sequently is connected. A* method with path finding technique is used to control virtual camera based on non-player character movement. Surrounding objects are identified to set camera position in order to avoid a collision in capturing non-player character movement [20].

In our previous works [21], we have developed an automatic virtual camera control in 3D environment game, using rule-based method. Type of shots in cinematograph rules, which are camera distance, camera height, camera angle, is formalized to define perspective packages that set the searching area for camera placement. Random technique is used to generate virtual camera placement based on perspective packages. The result shows that perspective packages can capture ideal view in real-time play, but it omits the flow of storytelling in a visual way. In this research, we develop our approach to more consider to visualization of storytelling by imitating the shot in reference films.

3. System Design

This research aims to develop an automatic virtual camera control that supports 3D animation filmmaking based on 3D game. The virtual camera in the 3D game is expected to able to capture ideal view, including the visualization of storytelling. The model of automatic virtual camera control consists of two main parts, which are rules, and inference engine. Virtual camera control rule is a representation of camera control knowledge. The knowledge is acquisitioned based on type of camera shots in the cinematography rules.

The virtual camera control rules are formalized based on six parameters, which are number of actors defined based on the number of dominant actors that act in one frame; actions which controls acts of actor; environment categorized into interior and exterior; camera shot which controls perspective packages to shot; shot duration which contains data of length of time used in one shot; shot sequence which controls data of change of view in every shot. Inference engine is developed using genetic algorithm to generate the placement or movement of virtual camera using perspective packages of camera control as the constraint.

The system works by identifying the state of actor based on the number of actor, actions, and environment. This information is to select candidates of camera shot for control camera including perspective packages. The identification number of actor is conducted based on collision detection among actors in stage, while for a shot where there is no actor in the stage; user interaction is needed to inform the system by pressing button *V* in the keyboard. Button *V* is to control the visibility of actor. Pressing button *V* makes actor invisible in the stage, so the shot captures only environment, and repressing button *V* makes actor visible in the stage. The identification of action is conducted based on user interaction in controlling the act of actor. We set parameters inside building 3D model, so system can detect type of exterior or interior environment. Using identification of number of actor, action, and environment, system sets camera placement or camera movement to search the best camera control, perspective package, and shot duration to capture the best view during the game play.

3.1 Building Camera Shot Knowledge Base

Shot is the smallest unit of visual information captured at a same time (duration) by camera which shows certain actions or event. Every change of camera results a new shot, such as close-up changed to medium shot means that two shots are used which each shot has its duration. There are four parameters in shot, which are camera distance which defines the distance between camera and actor or object filmed, camera height which defines the height of camera to actor or object filmed, camera angle which defines the rotation degree of camera to actor or object filmed, and camera movement which defines the camera movement to actor or object filmed [22] [23] [24] [25].

Parameters of camera distance, camera height, and camera angle control camera placement and camera movement. In camera distance, a shot can be categorized based on size of object in a screen, which categorized to extreme long shot, very long shot, long shot, medium long shot, medium shot, medium close-up, close-up, big close-up, extreme close-up. On the other side, camera angle is used to set the camera placement which includes height and rotation. Camera angle which is commonly used is categorized into eye-level, which places the camera parallel to the eye of the subject, high angle which places the camera above the subject and directs to the below, low angle which places the camera below the subject and directs to the top, bird's eye which places the camera above the scene and directs to the below [24]. Figure 1 shows the illustration of camera distance based on size of object in a screen, and Figure 2 shows the illustration of camera angle.

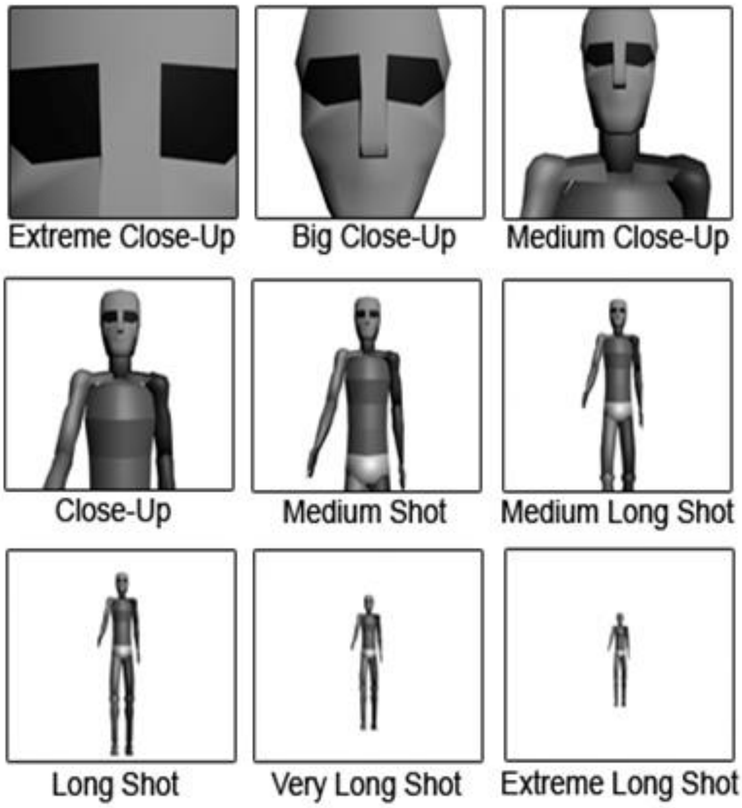

Fig.1. Illustration of type of camera distance

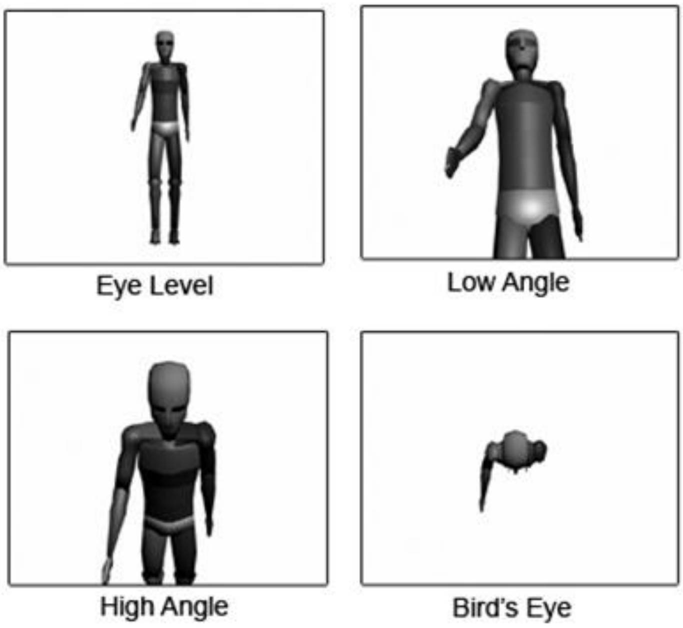

Fig. 2. Type of camera angle

Camera movement is an element of cinematography which can influence the meaning of shot and scene. Types of camera movement are pan and tilt which moves the camera horizontally and vertically, doly which moves the camera close to or away from subject, crab which moves the camera to the left or the right following the subject movement parallel [24].

Another shot setting is type of two-shot, which is a frame composition that contains two actors. Type of two-shots is profile two-shot used to help dialogue initial setting between two actors in the film, and over-the-shoulder two-shot which captures the face of an actor, where camera is placed in behind the other actor, and in a little shift from the his head. There is rule for a composition which contains two actors, called line of interest. The rule is a connection line to two main subjects in a scene. The camera placement or movement must remain in one side of the line of interest, and not to cros the line [26] [27]. Figure 3 shows illustration of type of two-shot.

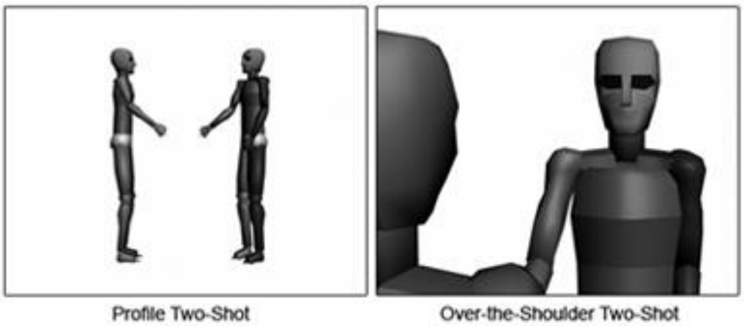

Fig. 3. Illustration of two-shot

Camera shot knowledge base contains information camera placement based on type of camera distance, and camera angle, including front, back, right, and left sides, where the camera is placed to the object. The information is recorded as indexed perspective packages. For example, there are four perspective packages defined in type of extreme close up with eye-level for each side, and also in big close-up with eye-level, and close-up with eye-level. The combination of camera distance and camera angle is added with high angle, low angle for types of medium close-up, medium long shot, medium long shot, long shot, very long shot, and extreme long shot. A combination of bird's eye type and top side is added to type of very long shot and extreme long shot. There are 86 perspective packages defined in camera shot knowledge base acquisition. Table 1 shows some of perspective packages (P) defined for type of extreme close-up, big-close-up, close-up, and medium close-up.

Table 1: Example of perspective packages

Camera Distance	Camera Angle	Sides	Persp. Pack.
Extreme Close-Up	Eye-Level	Front	P1
		Back	P2
		Right	P3
		Left	P4
Big Close-Up	Eye-Level	Front	P5
		Back	P6
		Right	P7
		Left	P8
Close-Up	Eye-Level	Front	P9
		Back	P10
		Right	P11
		Left	P12
Medium Close-Up	Eye-Level	Front	P13
		Back	P14
		Right	P15
		Left	P16
	High Angle	Front	P17
		Back	P18
		Right	P19
		Left	P20
	Low Angle	Front	P21
		Back	P22
		Right	P23
		Left	P24

Each perspective package view is simulated in 3D game engine program to identify the value of X, Y, Z of camera position and rotation to the actor. The simulation results relative value in controlling the placement of the camera. Relative value is a value range to place a camera that can capture a view as in a perspective package. Figure 4 shows the illustration of relative values for position and rotation of camera to result perspective package P25 (medium shot/eye level/Front).

Fig. 4. Result of relative values for P25 (medium shot/eye level)

Table 2 shows relative value examples of camera position, and Table 3 shows examples of camera rotation built in Unity 3D engine.

Table 2: Relative value of camera position for perspective packages

ID	X		Y		Z	
	Min	Max	Min	Max	Min	Max
P1	-0.058	0.058	1.325	1.36	0.11	0.13
P2	-0.058	0.058	1.325	1.36	-0.16	-0.18
P3	-0.11	-0.18	1.325	1.36	-0.058	0.058
P4	0.11	0.18	1.325	1.36	-0.058	0.058
P5	-0.07	0.07	1.365	1.39	0.14	0.24
P6	-0.07	0.07	1.365	1.39	-0.18	-0.24
P7	-0.18	-0.24	1.365	1.39	-0.07	0.07
P8	0.18	0.24	1.365	1.39	-0.07	0.07
P9	-0.114	0.114	1.34	1.39	0.24	0.31
..	…	…	…	…	…	…
P86	-4	4	8	13	-4	4

Table 3: Relative value of camera rotation for perspective packages

ID	X		Y		Z	
	Min	Max	Min	Max	Min	Max
P1	-4	4	170	190	-1	1
P2	-4	4	-10	10	-1	1
P3	-4	4	80	100	-1	1
P4	-4	4	260	280	-1	1
P5	-4	4	170	190	-1	1
P6	-4	4	-10	10	-1	1
P7	-4	4	80	100	-1	1
P8	-4	4	260	280	-1	1
P9	-4	4	170	190	-1	1
..	…	…	…	…	…	…
P86	80	100	0	360	0	360

Camera placement is defined based on the relative value of perspective packages. For example, perspective package P1 (extreme close-up/eye-level/front) is achieved by set the position of camera in coordinate X at -0.058 to 0.058, coordinate Y at 1.325 to 1.36, coordinate Z at 0.11 to 0.13, and set the rotation of camera in axis X at -40 to 40, axis Y at 1700 to 1900, axis Z at -10 to 10.

Relative value range to result a view which fulfils characteristic of perspective package can be formulated as:

$$CP = (minPos \leq nCP \leq maxPos)$$
$$CR = (minRot \leq nCP \leq maxRot)$$
$$PP = \{CP, CR\} \tag{1}$$

Where:
CP: camera position
CR: camera Rotation
nCP: value X, Y, Z of *CP*.
nCR: value X, Y, Z of CR.
minPos: minimum value of *CP*
maxPos: maximum value of *CP*
minRot: minimum value of *CR*
maxRot: maximum value of *CR*
PP: perspective package

A movement area is defined firstly for camera movement, and then a path for the movement is defined in the area. For instance, an area for camera movement type of dolly in is defined based on camera placement data, with camera position and rotation at the beginning is placed in area of medium long shot, or medium close-up. Camera position and rotation at the end is in area of close up, or medium close-up. There are ten types of camera movement used in this research, which are follow, pan left, pan right, pan up, pan down, tilt left, tilt right, tilt up, tilt down, dolly in, and dolly out. Figure 5 shows an illustration of area definition for camera movement, with gray area represents movement area for dolly, and camera rotation follows the shape of the gray area.

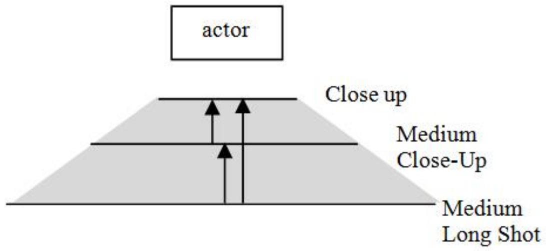

Fig. 5. Movement area definition based on relative value for dolly in

Camera movement uses at least two perspective packages for beginning and end of the movement, except for type of follow movement. Camera movement can be formulated as:

$$Camera\ movement = \{PP_{start}, PP_{end}\} \tag{2}$$

The camera shot knowledge base defined based on PP, including camera movement is then used as references to set perspective packages in style imitation process.

3.2 Defining Rules Using Style Imitation Technique

There are six parameters used for automatic virtual camera control, which are number of actors, action, environment, camera shot, shot duration, and shot sequence. Parameters value is defined by imitating shots in reference film. Number of actors (NA) is defined based on the number of dominant actors that act in one shot. Value of number of actors is set to 0 if there is no actor in a shot, 1 for one actor in a shot, 2 for two actors in a shot, and 3 for more than 2 actors in a shot. NA can be formulated as:

$$NA = \{0, 1, 2, 3\} \quad (3)$$

Type of actions (CK) is categorized based on actor's movement. Value of CK = 0 is for NA = 0. No movement condition is for stand up, sit, or other acts that need a little movement. Movement is for walking, running, or other acts that need more movement. Dialogue is for a shot where more than 2 actors are involved. CK can be formulated with value 0 for empty, 1, for no movement, 2 for movement, 3 for dialogue, as:

$$CK = \{0, 1, 2, 3\} \quad (4)$$

Environment (E) is categorized in interior and exterior. E can be formulated, with value 0 for interior, and 1 exterior, as:

$$E = \{0, 1\} \quad (5)$$

Camera shot (CC) contains type of camera placement and movement to capture a shot. Type of CC is defined with still camera/placement (0), follow (1), pan right (2), pan left (3), pan up (4), pand down (5), tilt right (6), tilt left (7), tilt up (8), tilt down (9), dolly in (10), dolly out (11). CC can be formulated as:

$$CC = \{0, 1, 2, 3, \ldots, 11\} \quad (6)$$

Perspective package (*PP*) is defined based on camera shot knowledge base in which 86 perspective shot packages (P1-P86) are built. *PP* can be formulated as:

$$PP = \{P1, P2, .., P86\} \quad (7)$$

Shot duration (*SD*) is length of time in one shot. Shot duration is defined by identifying shot duration based on *NA*, *CK*, *E*, and *CC* in film reference. Data of *SD* are sorted to find minimum and maximum value of shot duration. Minimum and maximum values of shot duration are used as duration range to capture a shot. *SD* can be formulated, with T_1 and T_2 represent the minimum and maximum value of *SD*, as:

$SD = Range\ (T_1, T_2\}$ (8)

Shot sequence (*SQ*) contains data of change of view in term of *previous shot* ⇒ *next shot*. *SQ* is defined based on following perspective package of current shot ID. Figure 6 shows pseudocode of *SQ*:

```
for (i = 0; i < (total shot – 1); i++) {
    SQ [i] = PP (ID [i+1]);
}
```

Fig. 6. Pseudocode for shot sequence

Figure 7 shows model diagram of automatic virtual camera control developed in this research.

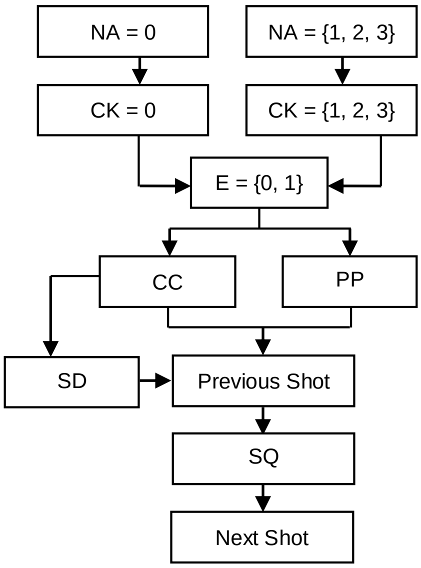

Fig. 7. Model diagram of automatic virtual camera control

Components of style imitation are then formulated to define rules of control camera, shot duration and shot sequence as follows:

IF *NA*
THEN *CK*

IF *CK*
THEN *E*

IF *NA* **AND** *CK* **AND** *E*
THEN *CC* **AND** *PP*

IF *NA* **AND** *CK* **AND** *E* **AND** *CC*
THEN *SD*

IF *NA* **AND** *CK* **AND** *E* **AND** *CC* **AND** *SD*
THEN *Previous Shot*

IF *Previous Shot*
THEN *SQ*

IF *SQ*
THEN *Next Shot*

A 3D animation film entitled *How to Train Your Dragon 2* produced by *DreamWorks Animation* is used as reference film in this experiment. We limit the imitation in first 100 shot, with total duration at 4.7 minutes. Table 4 shows some data (shot 1 – 25) resulted from style imitation process of 100 shots of the reference film.

Table 4: Data Resulted From Style Imitation

Shot ID	NA	CK	E	CC	PP		SD
					Start	End	
1	0	0	1	10	61	49	15
2	0	0	1	2	49	49	4
3	0	0	1	2	53	53	4
4	0	0	1	3	41	41	2
5	0	0	1	9	57	13	4
6	0	0	1	3	53	53	5
7	0	0	1	0	25	25	7
8	0	0	1	0	57	57	1
9	0	0	1	0	25	25	1
10	0	0	1	0	57	57	1
11	3	2	1	1	53	53	1
12	3	2	1	2	53	49	6
13	1	2	1	1	5	5	7
14	2	3	1	1	49	49	3
15	1	2	1	1	30	30	3
16	1	2	1	1	9	9	1
17	2	2	1	0	49	49	1
18	3	2	1	1	49	49	1
19	1	2	1	1	19	19	1
20	2	2	1	1	37	37	1
21	2	2	1	1	45	45	7
22	3	2	1	7	69	27	3
23	3	2	1	7	49	27	1
24	0	0	1	0	9	9	1
25	3	2	1	7	13	20	5

Data resulted from style imitation process show the technique of camera control used in the references film. For example, for a shot with no actor (*NA* = 0), type of camera control (*CC*) used to capture a shot are still (0), pan left (2), pan right (3), tilt down (9), and dolly in (10), and type of perspective packages (*PP*) used to capture the shot, including starting points for camera movement, are 0 | P5 (still | BCU/EL/F), 0 | P9 (still | CU/EL/F), 0 | P13 (still | MCU/EL/F), 0 | P25 (still | MS/EL/F), 2 | P41-P41 (pan left | MLS/HA/F - MLS/HA/F), 2 | P49-P49 (pan left | LS/EL/F - LS/EL/F), 3 | P53-P53 (pan right | LS/HA/F - LS/HA/F), 3 | P57 (pan right | LS/LA/F - LS/LA/F), 9 | P57-P13 (tilt down | LS/LA/F - MCU/EL/F), 10 | P61-P49 (dolly in | VLS/EL/F

- LS/EL/F). Duration for all shot of *NA* = 0 has range at 1-15 seconds. Table 5 shows data of imitation of *NA* = 0.

Table 5: Data of imitation of NA = 0

Shot ID	NA	CK	E	CC	PP Start	PP End	SD
1	0	0	1	10	61	49	15
2	0	0	1	2	49	49	4
3	0	0	1	2	53	53	4
4	0	0	1	3	41	41	2
5	0	0	1	9	57	13	4
6	0	0	1	3	53	53	5
7	0	0	1	0	25	25	7
8	0	0	1	0	57	57	1
9	0	0	1	0	25	25	1
10	0	0	1	0	57	57	1
24	0	0	1	0	9	9	1
26	0	0	1	2	25	25	7
40	0	0	1	2	49	49	2
46	0	0	1	0	5	5	1
48	0	0	1	0	13	13	1
55	0	0	1	0	9	9	1
56	0	0	1	0	13	13	2
83	0	0	1	0	13	13	1

Rules of control camera are defined by classifying *NA;* follow with *CK*, and then *E* to create prune of *CC* and *PP*. Table 6 shows result of prune of *CC* and *PP* for *NA* = 3.

Table 6: Rules of CC and PP for NA = 3

NA	CK	E	CC	PP Start	PP End
3	1	1	0	25	25
	2	1	0	53	53
			1	27	27
				42	42
				49	49
				53	53
				55	55
			2	53	49
			7	13	20
				49	27
				49	50
				69	27

Type of actions in shots are no movement (*CK* = 1), and movement (*CK* = 2). Background for all shots is exterior (*E* = 1). Type of camera control is 0 | P25 (still | MS/EL/F) is used for *NA* = 3, *IM* = 1, *CK* = 1, and *E* = 1. While for *NA* = 3, *IM* = 1, *CK* = 2, and *E* = 1, type of *CC* and *PP* are 0 | P53, 1 | P27 - P27, 1 | P42 - P42, 1 | P49 - P49, 1 | P53 – P53, 1 | P55 – P55, 2 | P53 – P549, 7 | P13 – P20, 7 | P49 – P27, 1 | P49 – P50, 1 | P69 – P527.

Rules of shot duration are defined based on current state of *NA*, *CK*, *E*, and *CC*. For example, if *NA* = 3, , *CK* = 1, *E* = 1, with *CC* = 0, then shot duration is 1 second, while if *NA* = 3, *CK* =

1, E = 1, with CC = 7, then shot duration is in range of 1 – 5 seconds. Table 7 shows result of shot duration for NA = 2 and 3.

Table 7: Rules of SD for NA = 2 and 3

NA	CK	E	CC	SD	
				Min	Max
2	1	1	0	1	3
	2	1	0	1	4
			1	1	7
			3	6	6
			10	1	1
	3	1	0	5	5
			1	3	3
			7	7	7
3	1	1	0	1	1
	2	1	0	2	2
			1	1	7
			2	6	6
			7	1	5

Rules of shot sequence (SQ) are defined based on NA, CK, E, CC, PP, as its order in shot extraction and in term of $A \Rightarrow B$, where A is previous shot, and B is next shot. Table 8 shows examples of some shot extraction, in which value 1 for NA is used as an example of previous shot in defining rules of shot sequence.

Table 8: Data of shot extraction

Shot ID	NA	CK	E	CC	PP	
					Start	End
1	0	0	1	10	61	49
…	…	…	…	…	…	…
13	1	2	1	1	5	5
14	2	3	1	1	49	49
15	1	2	1	1	30	30
16	1	2	1	1	9	9
17	2	2	1	0	49	49
…	…	…	…	…	…	…

Table 9: Rules of shot sequence (SQ)

Shot	NA	CK	E	CC	PP	
					Start	End
Prev	1	2	1	1	5	5
Next	2	3	1	1	49	49
Prev	1	2	1	1	30	30
Next	1	2	1	1	9	9
Prev	1	2	1	1	9	9
Next	2	2	1	0	49	49

3.3 Generating Virtual Camera Control

Inference engine works by value of *NA*, *CK*, and *E*, and user interaction in controlling actor in game. Genetic algorithm is used to generate virtual camera control. Variables *CC*, *PP*, *SD*, and *SQ* are used as constraint to define the best of virtual camera. A number of virtual cameras are generated as individuals in population. Each individual has chromosome which contains six genes that represent position *PX*, *PY*, *PZ*, and rotation *RX*, *RY*, RZ. Value of position is set randomly based on minimum and maximum value recorded in knowledge base. Value of rotation is set with random 360^0.

Objective function is dfined based on the number of genes in chromosom. Each variable of *PX*, *PY*, *PZ*, *RX*, *RY*, and *RZ*, which represents genes in chromosom, has value 1 if they fit to perspective packages as defined in rules of *SQ*. The objective function can be achieved with:

$$\begin{aligned}&(Number\ of\ genes - total\ value\ of\ genes) = 0\\&6 - (PX + PY + PZ + RX + RY + RZ) = 0\\&6 - (1 + 1 + 1 + 1 + 1 + 1) = 0\end{aligned} \quad (9)$$

Figure 8 shows pseudocode used to generate population of virtual camera.

```
Number of population = NP = 5;
Number of genes = NG = 6;
Chromosome = C;
Virtual camera = VC;
For (i = 0; i < NP; i++) {
   C [i] = “VC” + [i];
}
n = 0;
while (n < NP) {
   for (i = 0; i NG; i++) {
      C [n] [i] = (PX, PY, PZ, RX, RY, RZ);
   }
   for (i = 0; i < (NP / 2); i++) {
      C [n] [i] = random (position value);
   }
   for (i = (NP / 2); i < NG; i++) {
              C [n] [i] = random (360)
   }
   n++;
}
```

Fig. 8. Pseudocode for generating population

Pseudocode above results a population which consists of five virtual cameras (VC0, VC1, VC2, VC3, VC4). Each of their chromosomes contains six genes of *PX, PY, PZ, RX, RY, RZ*, where position of each gene is set with random value with range of minimum and maximum value of position, and rotation of each gene is set with random value of 360^0.

Rank selection technique is implemented to select chromosomes. Crossover is conducted with one-point crossover, by cut genes of chromosomes into two parts of *PX, PY, PZ,* and *RX, RY, RZ*. Crossover is implemented with *PX, PY, PZ* of chromosome (n) >< RX, RY, RZ of chromosome (n+1), and *PX, PY, PZ* of last chromosome >< *RX, RY, RZ* of first chromosome. Mutation rate is set at 0.1, so the value of mutation is as follow:

$$\textit{total number of genes} \times 0.1$$
$$(6 \times 5) \times 0.1 = 30 \times 0.1 = 3. \tag{10}$$

Each gene is filled with value of random (total number of genes). Gene with random value ≤ mutation value is mutated with new value, and new generation is created. Proces of selection, crossover, and mutation are iterated until finding the best chromosome which is fit, and the process is limited with sot duration for each perspective package. The iteration is stoped when the best chromosome cannot be defined. In such condition, virtual camera control takes the previous perspective package to capture the game, and the searching process of best chromosome is restarted.

4. Implementation and Evaluation

Design of automatic virtual camera control is implemented in a game developed in this research. A short animation film with duration at 138 seconds is created based on the game. The animation consists of 2 main actors with type of actions containing stand-by, walking, dialogue, and exterior view. Pictures below show the screenshot of the animation.

Fig. 9. Screenshot of 3D animation film produced in this research

A 3D animation film produced by the game developed in this research has 45 shots. Table 10 shows some data of experiment result in producing 3D animation film.

Table 10: Shot produced in experiment

ID	NA	CK	E	CC	PP		SD
					Start	End	
1	0	0	1	0	P25	P25	6.7
2	0	0	1	0	P57	P57	3.2
3	1	1	1	0	P40	P40	1.6
**4*	*1*	*1*	*1*	*0*	*P40*	*P40*	*1.6*
5	1	2	1	6	P44	P42	3.0
6	1	2	1	0	P49	P49	2.5
7	1	2	1	1	P5	P5	1.6
8	1	2	1	0	P62	P62	1.2
9	1	2	1	1	P76	P76	1.2
**10*	*1*	*2*	*1*	*1*	*P76*	*P76*	*1.2*
11	1	2	1	1	P42	P42	4.5
12	1	2	1	1	P5	P5	3.9
13	1	2	1	1	P43	P43	4.9
14	1	2	1	1	P11	P11	3.2
15	1	2	1	1	P9	P9	4.2
16	1	2	1	1	P38	P38	1.9
17	1	2	1	0	P49	P49	2.2
**18*	*1*	*2*	*1*	*0*	*P49*	*P49*	*2.2*
19	1	2	1	1	P5	P5	3.7
20	1	2	1	1	P43	P43	4.1
21	1	2	1	1	P11	P11	5.2
22	1	2	1	1	P9	P9	2.9
23	2	2	1	0	P49	P49	2.0
**24*	*2*	*2*	*1*	*0*	*P49*	*P49*	*2.0*
25	1	2	1	1	P5	P5	4.9
26	1	2	1	0	P62	P62	1.4
**27*	*1*	*2*	*1*	*0*	*P62*	*P62*	*1.4*
28	1	2	1	1	P76	P76	5.1
29	1	2	1	1	P42	P42	4.2
30	1	2	1	1	P5	P5	4.8
31	1	2	1	1	P43	P43	6.3
32	1	2	1	1	P11	P11	1.0
**33*	*1*	*2*	*1*	*1*	*P11*	*P11*	*1.0*
34	1	2	1	1	P9	P9	4.8
35	1	2	1	11	P45	P69	3.0
36	2	1	1	0	P25	P25	2.6
37	2	2	1	1	P54	P54	1.1
**38*	*2*	*2*	*1*	*1*	*P54*	*P54*	*1.1*
39	2	2	1	1	P57	P57	6.4
40	2	2	1	1	P25	P25	4.2
41	2	2	1	1	P39	P39	3.8
42	2	2	1	1	P54	P54	1.7
**43*	*2*	*2*	*1*	*1*	*P54*	*P54*	*1.7*
44	2	2	1	1	P57	P57	3.6
45	2	2	1	1	P25	P25	3.7

Evaluation is conducted to measure performance of genetic algorithm in generating virtual camera control, and visual output based on imitation style. Evaluation of the performance of genetic algorithm is conducted by measuring the number of shots which can be generated by genetic algorithm to total number of shots. 37 of 45 shots can be generated by system, while 8 shots marked with asterisk in Table 10 use previous virtual camera control. Genetic algorithm is failed to generate 8 shots. The result, with *SH* represents the number of shots which can be generated by genetic algorithm, and *TSH* represents total number of shots, is described as follow:

$$\frac{SH}{TSH}\ x\ 100\%$$

$$\frac{37}{45}\ x\ 100\%$$

$$= 82.2\% \quad (11)$$

Evaluation of visual output based on imitation style is conducted by comparing shots produced in this research to shots of reference film based on point of view. This evaluation involves three experts of film and multimedia. Two experts have academic background as lecturer with subject multimedia, and one expert has practitioner background as camera man. The experts are asked to value similarity between shots produced in this research and shots from reference film in scale 1-5, with 1 for bad, 2 for poor, 3 for fair, 4 for good, 5 for excellent. Mean Opinion Score technique (MOS) is used to measure the value given by experts. Below is formula of MOS, where *R* is values given by *n* subjects.

$$MOS = \frac{\sum_{i=1}^{n} R\ n}{n} \quad (12)$$

Evaluation is conducted in three sessions. 45 shots of 3D animation produced in this research are distributed into three sessions. Shots of reference film are selected based on same value of *NA*, *CK*, *E*, *CC*, and *PP* with shots to be evaluated. Evaluation of visual output based on imitation style in session 1 results value at 3.9, session results score at 4.1, session 3 result score at 3.7. Mean score of this evaluation is 3.9, where the value represents above the score of fair, and closer to the score of good. Figure 10, 11, 12 show chart of result of similarity evaluation in 3 sessions by experts, and Figure 13 shows result of similarity evaluation with MOS.

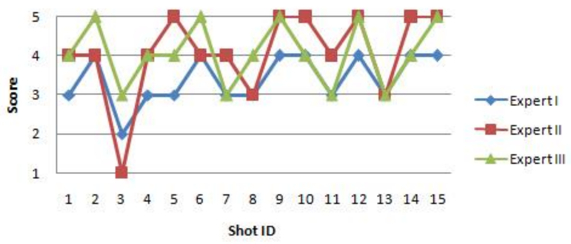

Figure 10: Similarity evaluation of session 1

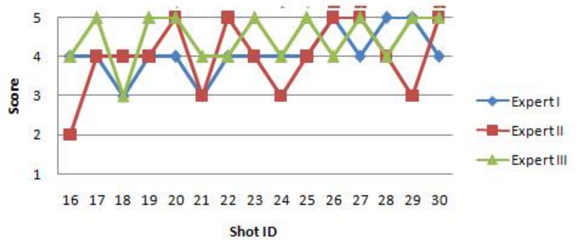

Figure 11: Similarity evaluation of session 2

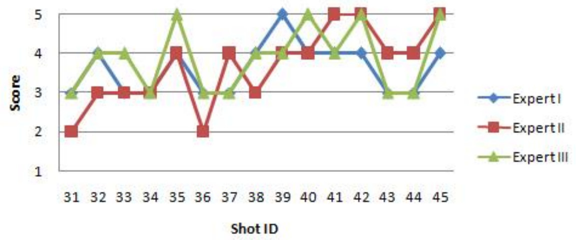

Figure 12: Similarity evaluation of session 3

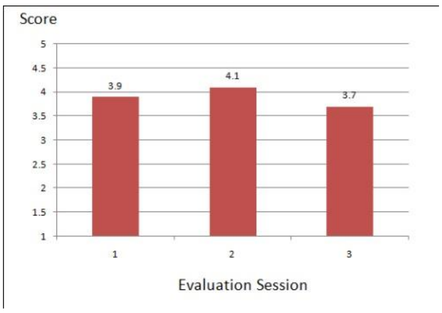

Figure 13: Result of similarity evaluation with MOS

5. Conclusion and Future Works

The genetic algorithm can generate up to 82.2% shots of total number of shots, with similarity to shots of reference film up 3.9 in scale of 5. Model of automatic virtual camera control using style imitation technique proposed in this research can be used to produce a 3D animation film based on 3D game with point of view which can be set based on reference film. Acquisition of knowledge is conducted manually, in which shots of reference film are examined by extracting the shots, and define its content manually. This process takes time, and different conclusions of view of the shots by different persons may happen. For future works, image processing technique can be used to analyze imitation components of shot of the reference film.

References

[1] Paolo Burelli, Gergios N. Yannakakis, "Toward Adaptive Virtual Camera Control in Computer Games", *SG 2011, LNCS 6815*, Springer-Verlag Berlin Heidelberg, 2011.

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

[3] J. Ng, "Understanding Machinima: Essays on Filmmaking in Virtual Worlds", *Bloomsbury*, (2012).

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

[5] Marc Christie, Patrick Olivier, Jean-Marie Normand, "Camera Control in Computer Graphics", *Computer Graphics Forum volume 27,* July 2008.

[6] G. Nierhaus, "Algorithmic Composition: Paradigms of Automated Music Generation", *Springer*, (2009)

[7] M. Hamanaka, K. Hirata, S. Tojo, "Melody Morphing Method Based On Gttm". In *Proceedings of the International Computer Music Conference, pp. 155–158,* 2008

[8] R. wooler, A.R. Brown,"Investigating Morphing Algorithms for Generative Music". In *Proceedings of the International Conference on Generative Systems in the Electronic Arts*, 2005

[9] K. Ebcioglu, "An Expert System For Harmonizing Four-Part Chorales", In: *Schwanauer SM, Levitt DA (eds) Machine models of music*. MIT Press, Cambridge, Mass. ISBN 0-262-19319-1, 1992

[10] R.A. Mc Intyre, "Bach in a Box: The Evolution of Four Part Baroque Harmony Using the Genetic Algorithm". In: *Proceedings of the First IEEE Conference on Evolutionary Computation*, Orlando, Florida, 1994, vol 2. IEEE, Piscataway, NJ, pp 852–857, 1994

[11] Paolo Burelli, Gergios N. Yannakakis, "Toward Adaptive Virtual Camera Control in Computer Games", *SG 2011, LNCS 6815*, Springer-Verlag Berlin Heidelberg, 2011.

[12] Ali M. L., Goodwin S. D., "Applications of CSP Solving in Camera Control", *IEEE Communications Society*, IEEE CCNC 2008.

[13] Owen Bourne, Abdul Sattar, and Scott Goodwin."A constraint-based autonomous 3d camera system", *Institute for Integrated and Intelligent Systems*, Griffith University, 2007.

[14] A. Hornung, G. Lakemeyer, G. Trogemann, "An Autonomous Real-Time Camera Agent for Interactive Narratives and Games", *T. Rist et al. (Eds): IVA 2003, LNAI 2792, pp. 236-243, 2003*, Springer-Verlag Berlin, (2003).

[15] S. Jinhong, S. Miyazaki, T. Aoki, H. Yasuda, "Filmmaking Production System with Rule-based", In: *Image and Vision Computing, 366--371*. (2003).

[16] S.M. Drucker, T.A. Galyean, D. Zeltzer, CINEMA: a System for procedural camera movements, *Proceeding I3D '92 proceedings of the 1992 symposium on Interactive 3D graphics Pages 67-70,* (1992).

[17] Bares W., Thainimit S,. Mc Dermott S., Boudreaux C. , “A Model For Constraint-Based Camera Planning”, *Smart Graphics AAAI Spring Symposium, Stanford*, California, March 2000.

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

[19] Jardillier F., Langu’enou E., “Screen-space Constraints For Camera Movements: The Virtual Cameraman”, *Proceedings of the Eurographics Conference (EG 98), vol. 17,* Blackwell Publishers, 1998.

[20] A.Z. Fanani, D. A. Prima, Bela B.F.J, Edward S., M. Hariadi, I Ketut E. P., “Secondary Camera Movement in Machinema using Path Finding”, *International Conference on Technology, Informatics, Management, Engineering & Environment (TIME-IEEE)*, Bandung, Indonesia, June 23-26, 2013.

[21] A.Z. Fanani, S.M.S. Nugroho, M. Hariadi, Virtual Camera Control Based on Cinematographic Rules of Camera Shot and Camera Angle, *International Review on Computers and Software (IRECOS) Vol. 11 No. 6*, (2016).

[22] M. Pramoggiore, T. Wallis, Film: “A Critical Introduction”, *Laurence King Publishing*, (2005)

[23] R. Edgar, J. Marland, S. Rawie, “The Language of Film”, *Bloomsbury*, New York, (2015)

[24] C. J. Bowen, R. Thompson, “Grammar of the Shot”, *Focal Press*, (2013).

[25] B. Mamer, “Film Production Technique: Creating the Accomplished Image”, *Cengage Learning*, (2009)

[26] E.E.S. de Lima, C.T. Pozzer, M.C. d’Ornellas, A.E.M. Ciarlini, B.Feijó, A.L. Furtado, “Virtual Cinematography Director for Interactive Storytelling”, *Proceedings of the International Conference on Advances in Computer Entertainment Technology, Pages 263-270*, 2009

[27] L. He, Virtual Cinematography, “Theory and Practice for Automatic Real-Time Camera Control and Directing”, Microsoft Research, http://research.microsoft.com/users/lhe

*Corresponding author: Ahmad Zainul Fanani

Department of Electrical Engineering,

Institut Teknologi Sepuluh Nopember,

Kampus ITS Keputih, Sukolilo Surabaya 60111, Indonesia

E-mail: ahmad.fanani11@mhs.ee.its.ac.id