Intelligent Computer Moviemaker with the Capabilities of Cinematic Rule-based Reasoning

Abstract This paper describes a software system designed to automate the production of digital movies with various visual effects like three-dimension animation, real image, and their composition. The production system can understand user’s input screenplay through a parser then automatically interprets it into a relevant motion picture under the direction of a virtual director in place of a human one. The virtual director achieves user’s intentions through knowledge-based approach by setting a scene, determining the corresponding shot types and shot sequence, and planning virtual camerawork dependent on the cinematic expertise stored in a domain knowledge base. We model the filmmaking knowledge and rule-based reasoning strategies in expert system language CLIPS. Video data is encoded in XML and tracked by the MPEG-7 standard.

Keywords: Digital Movie Producer, Rule-based System, 3D Animation, Knowledge Representation, Virtual Director

1. Introduction

One of the most important things on Web communication lies in giving such an environment that enables any people to make his presentation and deliver it uncomplicatedly. Towards this technical orientation, we are developing a technique DMP (Digital Movie Producer) by which anyone can easily make and deliver his own movie 1)–4).

Current computer technology has reached a level that allows users to create a virtual world they can imagine. For example, software animators (Alias Maya, Autodesk 3D StudioMax, Avid SoftImage) for computer animation creation or software filmmakers (Adobe Premiere, Apple iMove) for video edition have been widely used to create digital film professionals and nonprofessionals. But these software tools largely rely on human’s intervention for the design of the resulting pictures. For those common users who are non-artists, it is also not easy to create high qualitative motion pictures.

Another difficulty of using the advanced software tools mentioned above is the fact that the process of 3D animation generating or film editing is quite troublesome. Point-click 3D animation packages only enable the automatic generation of action between key frames. Filmmaking packages also only provide partial automation at the post-production stage of filmmaking. When specifying a computer-generated animation sequence or picture, user manually inputs not only the underlying models which describe the geometry and dynamics of all the objects to be depicted, but also a large set of rendering options that need to be specified to affect aspects such as viewing perspective, lighting conditions etc, so that the user must undertake a lengthy off-line programming session. By Artificial Intelligent (AI) approach, the digital filmmaking procedure can be automated dependent on cinematic knowledge base. For those nonprogrammers, the cinematic knowledge-based environment instead of programming can save time and labor greatly.

The desktop movie making system DMP we are implementing can interpret a verbal screenplay into a relevant motion picture automatically with various visual effects like real image, three-dimensional (3D) animation, or their composition. Works on applying film theory for computer graphics generation have been put forward. Christianson et al. adopted the notion of film idioms from film theory and formalized them into a sequence of shots 5). He et al. encoded the film idioms into hierarchically organized finite state machine applied in real-time system 6). Amerson & Kime proposed a system FILM (Film Idiom Language and Model) for real-time camera control in interactive narratives 7). New methodologies employ knowledge-based approach to address the tasks of graphics and time-consuming. Point-click 3D animation packages only enable the automatic generation of action between key frames. Filmmaking packages also only provide partial automation at the post-production stage of filmmaking. When specifying a computer-generated animation sequence or picture, user manually inputs not only the underlying models which describe the geometry and dynamics of all the objects to be depicted, but also a large set of rendering options that need to be specified to affect aspects such as viewing perspective, lighting conditions etc, so that the user must undertake a lengthy off-line programming session. By Artificial Intelligent (AI) approach, the digital filmmaking procedure can be automated dependent on cinematic knowledge base. For those nonprogrammers, the cinematic knowledge-based environment instead of programming can save time and labor greatly.

The desktop movie making system DMP we are implementing can interpret a verbal screenplay into a relevant motion picture automatically with various visual effects like real image, three-dimensional (3D) animation, or their composition. Works on applying film theory for computer graphics generation have been put forward. Christianson et al. adopted the notion of film idioms from film theory and formalized them into a sequence of shots 5). He et al. encoded the film idioms into hierarchically organized finite state machine applied in real-time system 6). Amerson & Kime proposed a system FILM (Film Idiom Language and Model) for real-time camera control in interactive narratives 7). New methodologies employ knowledge-based approach to address the tasks of graphics and time-consuming. Point-click 3D animation packages only enable the automatic generation of action between key frames. Filmmaking packages also only provide partial automation at the post-production stage of filmmaking. When specifying a computer-generated animation sequence or picture, user manually inputs not only the underlying models which describe the geometry and dynamics of all the objects to be depicted, but also a large set of rendering options that need to be specified to affect aspects such as viewing perspective, lighting conditions etc, so that the user must undertake a lengthy off-line programming session. By Artificial Intelligent (AI) approach, the digital filmmaking procedure can be automated dependent on cinematic knowledge base. For those nonprogrammers, the cinematic knowledge-based environment instead of programming can save time and labor greatly.

The desktop movie making system DMP we are implementing can interpret a verbal screenplay into a relevant motion picture automatically with various visual effects like real image, three-dimensional (3D) animation, or their composition. Works on applying film theory for computer graphics generation have been put forward. Christianson et al. adopted the notion of film idioms from film theory and formalized them into a sequence of shots 5). He et al. encoded the film idioms into hierarchically organized finite state machine applied in real-time system 6). Amerson & Kime proposed a system FILM (Film Idiom Language and Model) for real-time camera control in interactive narratives 7). New methodologies employ knowledge-based approach to address the tasks of graphics and time-consuming. Point-click 3D animation packages only enable the automatic generation of action between key frames. Filmmaking packages also only provide partial automation at the post-production stage of filmmaking. When specifying a computer-generated animation sequence or picture, user manually inputs not only the underlying models which describe the geometry and dynamics of all the objects to be depicted, but also a large set of rendering options that need to be specified to affect aspects such as viewing perspective, lighting conditions etc, so that the user must undertake a lengthy off-line programming session. By Artificial Intelligent (AI) approach, the digital filmmaking procedure can be automated dependent on cinematic knowledge base. For those nonprogrammers, the cinematic knowledge-based environment instead of programming can save time and labor greatly.

The desktop movie making system DMP we are implementing can interpret a verbal screenplay into a relevant motion picture automatically with various visual effects like real image, three-dimensional (3D) animation, or their composition. Works on applying film theory for computer graphics generation have been put forward. Christianson et al. adopted the notion of film idioms from film theory and formalized them into a sequence of shots 5). He et al. encoded the film idioms into hierarchically organized finite state machine applied in real-time system 6). Amerson & Kime proposed a system FILM (Film Idiom Language and Model) for real-time camera control in interactive narratives 7). New methodologies employ knowledge-based approach to address the tasks of graphics and time-consuming. Point-click 3D animation packages only enable the automatic generation of action between key frames. Filmmaking packages also only provide partial automation at the post-production stage of filmmaking. When specifying a computer-generated animation sequence or picture, user manually inputs not only the underlying models which describe the geometry and dynamics of all the objects to be depicted, but also a large set of rendering options that need to be specified to affect aspects such as viewing perspective, lighting conditions etc, so that the user must undertake a lengthy off-line programming session. By Artificial Intelligent (AI) approach, the digital filmmaking procedure can be automated dependent on cinematic knowledge base. For those nonprogrammers, the cinematic knowledge-based environment instead of programming can save time and labor greatly.

The desktop movie making system DMP we are implementing can interpret a verbal screenplay into a relevant motion picture automatically with various visual effects like real image, three-dimensional (3D) animation, or their composition. Works on applying film theory for computer graphics generation have been put forward. Christianson et al. adopted the notion of film idioms from film theory and formalized them into a sequence of shots 5). He et al. encoded the film idioms into hierarchically organized finite state machine applied in real-time system 6). Amerson & Kime proposed a system FILM (Film Idiom Language and Model) for real-time camera control in interactive narratives 7). New methodologies employ knowledge-based approach to address the tasks of graphics and time-consuming. Point-click 3D animation packages only enable the automatic generation of action between key frames. Filmmaking packages also only provide partial automation at the post-production stage of filmmaking. When specifying a computer-generated animation sequence or picture, user manually inputs not only the underlying models which describe the geometry and dynamics of all the objects to be depicted, but also a large set of rendering options that need to be specified to affect aspects such as viewing perspective, lighting conditions etc, so that the user must undertake a lengthy off-line programming session. By Artificial Intelligent (AI) approach, the digital filmmaking procedure can be automated dependent on cinematic knowledge base. For those nonprogrammers, the cinematic knowledge-based environment instead of programming can save time and labor greatly.
generation. In 8-9), domain knowledge base was applied in automatically generating animation focusing on camera shot design while in 10) animation creation focused on human gesture. Cognitive modeling for intelligent agent was employed by Szarowicz et al to solve the same cinematic problem II). However few efforts were made in encoding the cinematic knowledge base and use it to automate the procedure of digital movie making from screenplay. Our desktop KB system DMP aims to create digital motion picture and decide the temporal order of video clips where the rule-based module is embedded as a subsystem in the integrated system environment to realize the automation.

The remainder of this paper is structured as follows. Section two introduces the system structure of DMP after analyzing the feasibility of automatic movie creation. The next section outlines the relevant filmmaking techniques utilized in the virtual 3D world from a film theoretic point of view and gives design about knowledge representation (KR) written in CLIPS language. In the fourth section, the system implementation with an example piece of animation is showed to expound how to use cinematic 'rules of thumb' to make a scene. Finally, we will have a discussion about our work.

2. System Design

2.1 Feasibility Analysis of Automatic Moviemaking

There are three general ways to generate digital movie.

1. Originating from conventional film: Change the images of film into digital video (DV).
2. Utilizing digital video camera to shoot: DV camera works as a carrier to shoot. The post-production of DV usually processes on computer.
3. Creating Computer Graphics (CG) movie: Whole movie is completely made by computer directly.

The traditional film production by the first way is very expensive and time-consuming so that it is not a rational approach for us. Movies generated by the ways (2) and (3) mentioned above are considered “real” digital ones, but it is impossible to produce personal movie readily within short time when we come up with an idea for movie because a variety of limits exist in these approaches.

Let us see these limits and analyze which steps in movie generation could be automated. For the second way, digital video production using DV camera encompasses acquisition, storage, selection/editing, and composition of video data (Fig. 1). Except that actual shooting requires human’ involvement, the process of video choosing and sequencing can be automated based on experienced editing knowledge. That is to say the process of digital video production is at most of automatic edition and composition.

possible to combine objects and actions according to the screenplay and to choose optimal placement for the camera automatically. Fig. 2 shows the production structure of CG Movie.

The dull rectangles in figure 1 and 2 indicate the functions that can be realized automatically. Therefore automatic edition and computer animation are feasible.

- **Automatic Editing** This function corresponds to the job of human film editor.
- **Automatic Animation** These jobs belong to film director and cinematographer.

2.2 System Architecture

The architecture of DMP with the capabilities of intelligent reasoning is generalized in Fig. 3.
Virtual director

The virtual film director is responsible for the visual aspect of screenplay dependent on knowledge of plot structure in KB. He gives commands for the dramatic structure, pace, and directional flow elements of the sounds and visual images to visualize the event. Composition, the location of characters, lighting styles, depth of field and camera angle are all determinant factors in the formulation of the visual information.

Knowledge base KB contains knowledge about objects, color, lighting, scene, shot, also contains spatial-temporal knowledge. It should enable the expert director or knowledge engineer to easily update and check the cinematic knowledge base.

Inference engine (Planner)

The inference engine, or called planner, is used to reason with both the cinematic knowledge and data from screenplay and other information from the user. Its result is called a "plan."

Movie player (Render)

Player assembles the resultant plan created by inference engine into images. Virtual camera records the frames that are to be played as a still or a sequence of images. We utilize Japanese NHK’s TVML player to render our digital movie, which can show animation or live-action film. Given the data of setting, lighting, objects, camera work, and sound, the player can render the movie resulted at any making stage so that we can test each shot result conveniently.

2.3 Information Format and Access

First emerging issue of designing system rests with human-computer interface. We employ verbal screenplay as input form in order to utilize the power of script of film. In order to reuse video assets, DMP uses metadata to describe the media stored in digital video library. There are some growing searchable multimedia libraries that currently have over thousands of hours of material, including films, news broadcasts, archive footage, etc. It is a good way to employ these video libraries and add our own materials to enrich it.

To conveniently communicate among different functional modules, we save the data of screenplay and digital movie clips in XML format. XML can be employed to format information of script since XML is a language suited for describing structured information and its properties. Video data may also be encoded in XML because XML acts well as a description language for multimedia. Video data are also tracked by the MPEG-7 standard because MPEG-7 coded data are mainly intended for content identification purposes while other coding formats such as MPEG-2, 4 are mainly intended for content reproduction purposes.

We provide a video modeling and query mechanism with the hierarchical structure (Fig. 4) constructed from the perspective of filmmaker to realize video reusing.

3. Knowledge Representation in CLIPS Language

In traditional film making it takes a group of craftspeople to create a film from directing to lighting and decorating a set. Computer 3D space is the same as film in the sense of representational multimedia, i.e., offering us information with visual moving image, graphics, speech, music, and sound effects. Both traditional film and computer 3D world project the three-dimensional space onto a two-dimensional surface. But 3D space generated from software exists only in cyberspace in the digital domain of computer & computer networks. Human uses and controls camera with optical system to shoot in the real world and camera represents the viewer's eye to record the sequence of frames that are to be rendered as a movie, while in the virtual world a mathematical model or a 3D engine is needed to construct the space from scratch. Correspondingly an abstract camera is used to represent where the viewer is standing and what scene he is looking in the virtual 3D space. Camera's location and aiming direction spatially determine what image of the space is displayed.

3.1 Available Filmmaking Techniques

Because there are no practical physical and optical constraints such as velocity, toque, or lens, the virtual camera works must comply with the common rubric of cinematography otherwise viewers will be confused by the unnatural dynamic graphics. We will deal with the filmmaking techniques that could be utilized in cyberspace applications. They are the four techniques about:

(1) Mise-en-scène (what to shoot) Those aspects of film such as setting, lighting, figures, movement, appearance, and costumes within the frame created by computers are all considered as parts of the mise-en-scène.

(2) Cinematograph (how to shoot it) comprises camera angles, camera distance and camera movements.

(3) Montage (how to present the shots) Types of montage and editing-techniques are divided according to the ways in which they connect different locations, perspectives, and how they dramatize the development of a scene or a narrative, e.g., parallel editing or cross-cutting, match cutting, jump cutting (opposed to match cutting), cutaways, thematic or montage cutting, and so forth.
(4) **Sound edition** (how to present the sounds) Image-related sounds include **dialog, music, background sound, and sound effects**. Sound editing refers to the mixing of all sound tracks and the integration of the final images with the sound track.

Creating a shot involves mise-en-scène & cinematograph, and creating a shot sequence involves montage and sound edition. To shoot the scene where two people are talking, over-the-shoulder (OTS) shot (from camera 1 or 2 in Fig. 5) created by triangle principle is usually used. The triangle principle has been extensively employed in many situations, particularly suitable to quiz shows, spots program, and sit-com. The advantage lies in that each talker is framed on the same side of the scene in each shot - character A on the left side and character B on the right side. If one character in a single shot is looking to the left and the other character in a single shot is looking to the right, when these shots are edited together. We usually assume that the characters are looking at each other.

### 3.2 Knowledge Representation

Different types of KB have different roles for different goals (13). We built a filmmaking knowledge base having the representation, contents, and inference strategies. When representing human expertise, it is usual to use rules to model domain knowledge. Each rule encodes a piece of filmmaking knowledge. Besides rules, knowledge representation still implies human behavior of intelligent reasoning, i.e. determining which rule will be applied (14). Representation and reasoning are inextricably intertwined with representation. We employed forward-chaining inference mechanics, the same as the way in which human makes a plan.

We need a well-defined language that enables us to represent complex expert knowledge in a clear, precise and nature way. Inference engine also needs a quick and stable algorithm to realize intelligent reasoning. Up to date the most efficient algorithm for the implementation of rule-based systems is the Rete algorithm (15). Expert system shells using Rete algorithm are OPS5, ART, Rete++, CLIPS, Jess, and so forth. We will choose CLIPS (C Language Integrated Production System) for the following reasons:

- CLIPS is a rule-based language suited for representing heuristic knowledge. The expert system language has the properties of natural language style, uniform structure, and good extensibility, suitable to modeling human knowledge or expertise.
- COOL (CLIPS Object-Oriented Language) is the tool suited for representing knowledge that supports inheritance. In addition to rules, we still need object-oriented programming that at least has the features of abstraction and inheritance to manage the independent objects such as characters and props in the virtual 3D world.
- It also has function paradigm suited for representing procedural knowledge like walking, jumping.
- Its Rete algorithm can improve the speed of forward-chained rule-based systems. The efficiency of Rete algorithm rests with the reason that it is asymptotically independent of the number of rules.

Though Rete has the drawback of high memory space requirements, the more advanced the technique of memory becomes, the small the problem turns into. In one sentence, CLIPS provides a cohesive tool for handling a wide variety of knowledge with supports for three different programming paradigms: rule-based, object-oriented and procedural so that it can fulfill the above mentioned needs for high-level tool to program the generation of movie.

#### 3.2.1 Rules about Shooting

Each shot has a purpose in the scene. A shot *sequence* is a group of shots depicting one action or which seems to belong with or depend upon each other. We make many assumptions when we watch a film. For instance, we assume that when one shot is placed next to another there is a significant relationship between them. An example sequence is showed in Fig. 6 composed of shots transformed from medium shot to two-shot through dolly shot (concerning camera movment). The definitions of these shots can be found in section 4.2.

In rule-based language, a rule is a concise description of a set of conditions and a set of actions to take if the conditions are true. Film rules about above shooting can be written in defrule construct of CLIPS as:
where two parts are separated by the ‘=>’ symbol (means ‘then’). The first part consists of the LHS left-hand side pattern (track-two-half-front) which is used to match facts in the knowledge base while the second part consists of the RHS right-hand side action (MS) (DS) (TS) that contain function calls. The rule of shot selecting will be activated when the fact (track-two-half-front) appear in the knowledge base. When the rule executes or fires, the functions (action MS), (action DS), and (action TS) are called. Annotation begins with symbol ‘$’.

3.2.2 Object’ Data
To write CLIPS rules about objects (e.g. characters or props) in 3D graphics space, it is necessary to describe all the scene information in COOL. The data structure of each object becomes a node, hierarchically organized into a tree structure standing for a scene. For example, the below class includes slots for parent node, children nodes, and character’s coordinates & direction. Each node of character contains the positional information (x, y, z) and rotational information (xdir, ydir, zdir).

```
(defclass NODE
  (slot node-index (type CHAR) (parameter INTEGER) (create-accessor read))
  (slot parent (type CHAR) (parameter FLOAT) (create-accessor read))
  (multislot children (type CHAR) (create-accessor read))
  (slot x (type CHAR) (parameter FLOAT) (create-accessor read))
  (slot y (type CHAR) (parameter FLOAT) (create-accessor read))
  (slot z (type CHAR) (parameter FLOAT) (create-accessor read))
  (slot xdir (type CHAR) (parameter FLOAT) (create-accessor read))
  (slot ydir (type CHAR) (parameter FLOAT) (create-accessor read))
  (slot zdir (type CHAR) (parameter FLOAT) (create-accessor read))
)
```

3.2.3 Objects’ Movement
CLIPS can deal with the motion of object in the way of pattern matching. For example, when character B (Tom) gets 2 feet near character A (Rose), he stops walking. Using defrule sentence to express this meaning:

```
(defvar two-talk-distance
  (character (name [Tom] (decide? walk))
  (character (name [Tom] (decide? walk))
  (position $length $length $walk $walk)
  (position $length ($length (+ $walk 2)
  & ($length (- $walk 2)
  =>
  (send [Tom] put-decide (+ $walk 0))))
```

3.2.4 Moviemaking Reasoning Procedure

See the usual steps in linear animation generation (Fig. 7). Camera works and sound can be set up at any stage.

If there are suitable video clips in video database or video web library, the required clips will be extracted from the database/library, otherwise, 3D animation will be created based on cinematic knowledge. Sets, props, and characters with action abilities have been pre-made and stored in a database as candidates since it is impossible to create accurate primitive objects automatically without modeling them in advance. If necessary, composition of animation and real image will be made.

4. Creation-from Event to Shot Sequence

A film is made up of shots arranged in sequence. The virtual director establishes a point of view on the action that helps determine the selection of shots and camerawork through rule-based planning, every shot timing out and important camera move. He first makes high-level shooting plan such as “track one’s face” for each event based on his directorial expertise, then gives commands about shot types and shot sequence, at last calculates the parameters of camera position, orientation, and movement to satisfy the these commands. When a ray from camera to target is occluded by object, the camera position and visual angle should be adjusted. If the camera is just put inside an object, the object could become transparent. If not, the position and angle of camera will be readjusted until satisfying some conditions evaluated by objective function. The main data flowing along the moviemaking pipeline from the textual input to clip output are "screenplay → elements (event, sound etc.) → shots → shot sequence → scene".

4.1 Virtual Camera Works

Our virtual camera is modeled with seven Degrees of Freedom (DOFs) - three for Cartesian position, three for orientation, and FOV (Field of View) all the same as those a common real-world camera has, so that it could be controlled in the same way as the real one, where FOV is the angle described by a cone with the vertex at the camera’s position, determined by the camera’s focal length. Various definitions of shot types are based on camera manipulation.

4.1.1 Shot Definition
(1) Shots concerning camera movement

Three degrees for the Cartesian position enable us to make crane shot, dolly shot, and tracking shot (or trucking shot), three degrees for the orientation enable pan (or panning, pan shot), rolling shot, and tilt (or tilt shot, vertical panning), the last one the FOV can be made by moving the camera directly or by zooming (or zoom lenses) as showed in Fig. 8. Other two movements in a
shot shootings are *hand-held shot* and *aerial shot* as the camera is not propped on a tripod or against any solid mount but held on the cinematographer's shoulder or sometimes also in his lap or taken from a helicopter.

(2) Shots concerning camera size

Shot of size or length of size indicates how far away the camera is from the target. An international 8-step system for character shooting includes the following typical shots.

- **Extreme Long Shot (ELS)/General View (GV)**
- **Very Long Shot (VLS)**
- **Full Shot (FS)/Long Shot (LS)**
- **Long Medium Shot (LMS)/Medium Long Shot (MLS)**
- **Medium Shot (MS)**
- **Medium Close-Up (MCU)**
- **Close-Up (CU/CS)**
- **Extreme Close-Up (ECU)/Big Close-Up (BCU)**

Shooting other than people is usually restricted to ELS, MS and CU. Sometimes a simpler 5-step ratio of ELS, VLS, MS, CU, and ECU is used. Variations on MS include the *two-shot* (containing two figures from the waist up) and the *three-shot* (contains 3 figures). Camerawork should be realized at least the above eight shots. Other sizes such as the *wide shot* for de-emphasis and conveying a character's isolation are also used. Fig. 9 illustrates a standard of shot size for character shooting we use.

(3) Shots concerning camera angle

Shots such as *bird's eye, high, eye-level, low, and worm's eye* refer to the horizontal and vertical position of the camera in relation to the subject. Through the use of angle, the director positions the subject within the frame. For example, a low angle shot placing the camera below a character exaggerates his importance. Shot angle has a great impact on how the audience perceives both the subject and the action.

**Perspective** is the psychological position of the camera when it records the action. There are three perspectives: objective, subjective and point of view (POV). Typically the viewpoint of virtual camera may be attached to a character in a story as a *subjective shot* (first-person point of view) or be separated from any character in the story as an *objective shot* (third-person point of view). First-person-views usually appear in Virtual Reality (VR), for example when user sees the Virtual Environment (VE) through his avatar's view, while in filmmaking the common way is to view events from the perspective of a person outside the film although sometimes the subjective point of view is also used as if the scene is seen from the character's eye. We expect that if the film is depicting real life then the characters will not look at the camera. Therefore our virtual camera had better be the third-person point of view. *POV shot* is from the perspective of a specific character, but not directly through the eyes of that character. A variation of the POV shot is OTS.

4.1.2 Mathematical Model of Camera

In mathematical formula the camera operation can be expressed as:

\[
\text{Camera} (x, y, z, \alpha_x, \alpha_y, \alpha_z, \delta_x, \delta_y, \delta_z, \theta_x, \theta_y, \theta_z, t_x, t_y, t_z, \text{speed, transition})
\]

where \(x, y, z\) are coordinates, \(\alpha_x, \alpha_y, \alpha_z\) are visual angles, \(\delta_x, \delta_y, \delta_z\) are relative distances of movement, \(\theta_x, \theta_y, \theta_z\) are basic movement parameters of these shots in movement, speed is of the camera, and transition refers to the movement style (uniform or variable motion). The virtual camera works in DMP not only refer to setting up shot types but also refer to the natural transitions from shot to shot by using spatial-temporal shots continuity techniques.

4.2 One Shot Generation

In this section we will use OTS (Fig. 10) shot as an example to demonstrate how to create a shot. OTS involves the techniques of mise-en-scène and cinematography.
Spatial constraints of OTS
Rules for creating OTS fall into three categories:

1) Location of characters (involving mise-en-scène)
   - The height of character A should be approximated 1/2 the size of the frame;
   - Character A should be at about the 2/3 lines on the screen.
   - Character B should be at about the 1/3 line on the screen.

2) Proportions and orientation of characters (involving mise-en-scène)
   - Character A and B face each other.
   - Character A faces the view.

3) Viewpoint (involving cinematography)
   - The camera view should be as close as possible to face directly on to character A.
   - The field of view should be between 20 and 60 degrees;

Expressing constraints
In practice application the above rules are written in CLIPS correspondingly:

(deftemplate character1 "Information of character A"
  (slot name (type STRING) (default ?DERIVE))
  (slot location (type SYMBOL) (default top-left))
  (slot orientation (type Number) (default 0)))

(deftemplate character2 "Information of character B"
  (slot name (type STRING) (default ?DERIVE))
  (slot location (type SYMBOL) (default top-right))
  (slot orientation (type Number) (default 180)))

(deftemplate camera "Viewpoint"
  (slot name (type STRING) (default ?DERIVE))
  (slot degree (type Number) (default 40))
  (slot min-degree (type Number) (default 20))
  (slot max-degree (type Number) (default 60)))

(defrule OTS "definition of OTS" ?action-OTS←(action OTS)
  ;If
  (character1 (name ?name1))
  (character2 (name ?name2))
  ;Then
  (retract ?action-OTS)
  (assert character1)
  (assert character2)
  (assert camera)
)

where facts are encoded by deftemplate - a list of named fields called slots.

4.3 A Shot Sequence Generation

Heuristics about making a sequence of shots involves the techniques of montage and sound related to image, and another unit in film named event. Event is an important primitive action unit in camera planning procedure such as “a private conversation between two characters” (two-talk). In DMP shot sequence of two-talk is decided by the following planning rules (involving continuity cutting):

(a) If character A and B have a private conversation, five basic shots could be used: two-shot (default size: full shot), profile shot (default size: close-up), over-the-shoulder shot, point-of-view shot (default size: close-up), and angular shot (default size: close-up).
(b) If both character A and B are silent, use two-shot.
(c) If character A talks, select one least used shot by A from the set of basic shots.
(d) If character B talks, select one least used shot by B from the set of basic shots.
(e) If character talks, OTS should be selected first.
(f) If the selected shot is not OTS, it should be set before OTS in the shot sequence.

To stage face to face two-talk, the virtual director determines five basic shots from nine camera positions (Fig. 11), selects shots from the set and arranges them in order dependent on dialogues. Some shots in set may be used not only once (e.g. OTS shot, see Table 1 and Fig. 12) while some may be never used (e.g. Angular shot).

5. Conclusion

CG movie was made first by means of animated film as seen in drawn or puppet animation and later by means of computer-generated images and animations. In recent
years, filmic techniques have been extended to a degree possible with live actors shot in real time. In the commercial fields, current activities point to an interesting market potential for shared virtual environments where a knowledge-based approach to graphics generation is always essential. The rule-based engine in our system DMP can select the contents of presentation from video database and decide the temporal order of video clips or create motion picture of animation, where the rule-based module is embedded as a subsystem in the integrated system environment to realize the automation.

In summary, cinematic knowledge base extracted from the film craftsmen can give great aid for the process of creating high quality motion pictures, be used to automate the digital filmmaking procedure through knowledge-based approach, or help for film theory teaching with easy-to-use user interface. From its beginnings, DMP including its cinematic knowledge base has undergone continual refinement and improvement.

6. Acknowledgement

Our project DMP is supported by Japan Panasonic Corporation. This work was carried out with Panasonic Corporation and Tokyo Institute of Technology from 2002 for the project. We would like to thank Takafumi Yuki, Mitsuru Kaneko, and Koichi Emura specially.

[Reference]

2) Jinhong, Shen; Miyazaki, Seiya; Aoki, Terumasa, Yasuda, Hiroshi, "Virtual Camera Works with Planning Capabilities", The 7th World Multiconference on Systemics, Cybernetics and Informatics (SCIT03), XV, Orland, Florida, USA, pp130-135 (2003)
7) Amerson, D. and Kime, S., "Real-Time Cinematic Camera Control for Interactive Narratives"; In the Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, Stanford, CA (2001)
9) Kevin Kennedy, Robert. E. Mercer, “Planning animation cinematography and shot structure to communicate theme and mood”, Proceedings of the 2nd international symposium on Smart graphics, Hawthorne, NY, USA, pp1-8 (2002)