Filmmaking Production System with Rule-based Reasoning

Shen Jinhong, Seiya Miyazaki, Terumasa Aoki, and Hiroshi Yasuda
Faculty of Engineering,
The University of Tokyo, Tokyo, Japan
{j-shen, seiya, aoki, yasuda}@mpeg.rcast.u-tokyo.ac.jp

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

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. Usually the post-production of DV processes on computer. (3) Creating Computer Graphics (CG) movie: Whole movie is completely made by computer directly. Because these approaches are all time-consuming and/or expensive, it is still impossible for us to produce personal movie readily within short time when we come up with an idea for movie. Our research aims at developing an automation technique by which anyone can easily make and deliver his own movie.

Movies generated by the ways (2) and (3) mentioned above are considered "real" digital ones. Digital video production using DV camera encompasses acquisition, storage, selection/editing, and composition of video data. 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. For the production of CG Movie, supposing the existence of a library that stores 3D models and actions mentioned in the script, it is possible to combine objects and actions according to the screenplay and to choose optimal placement for the camera automatically. Therefore automatic edition and computer animation are feasible.

The desktop moviemaking system *DMP* (Digital Movie Producer) 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 [1-4]. The remainder of this paper is structured as follows. Section two introduces the system structure of DMP after analysing the related methods 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 of knowledge representation (KR) and reasoning strategies respectively. 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 Related Work

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 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 modelling for intelligent agent was employed by Szarowicz et al to solve the same cinematic problem [11]. However few efforts were made in encoding the cinematic knowledge base (KB) and use it to automate the procedure of digital movie making from screenplay. Our KB system DMP aims to create digital motion picture and decide the temporal order of video clips through the rule-based approach.

2.2 System Architecture

The architecture of DMP with the capabilities of intelligent reasoning is generalized in figure 1.

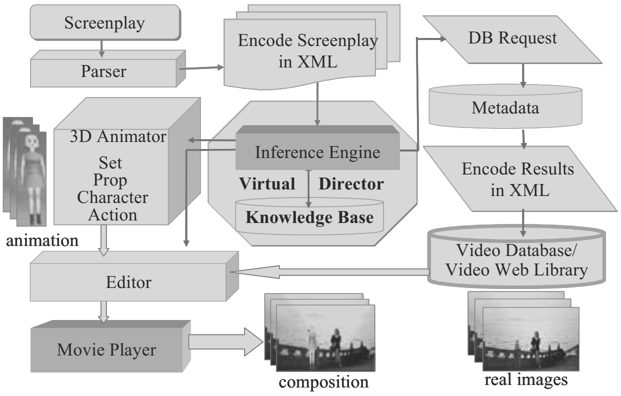


Figure 1: DMP system architecture

Virtual director The virtual film director is responsible for the visual aspect of screenplay dependent on knowledge of plot structure in KB and screenplay world. 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 between 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 to describe structured information and its properties. Video data may also be encoded in XML because XML acts as a description language for multimedia well. Video data is also tracked by the MPEG-7 standard because MPEG-7 coded data is 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 modelling and query mechanism with the hierarchical structure (figure 2) constructed from the perspective of filmmaker for DMP to realize video reusing.

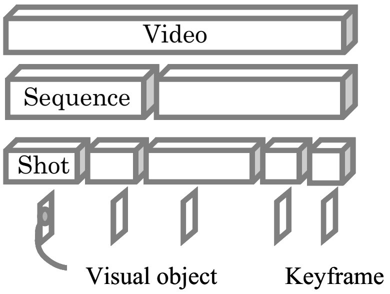


Figure 2: A hierarchy structure of video

3 Knowledge Representation and Reasoning Strategies

3.1 Available Filmmaking Techniques

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 within the digital domain of computer & computer networks. In the real world human uses and controls camera with optical system to shoot, 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. Its location and aiming direction spatially determine what image of the space is displayed [12].

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 *mise-en-scène* (what to shoot), *cinematograph* (how to shoot it), *montage* (how to present the shots), and *sound edition* (how to present the sounds) from film theory.

Those aspects of film such as setting, lighting, figures, movement, appearance, and costumes within the frame created by computers are all considered as part of the mise-en-scène. Cinematography comprises camera angles, mobile framing and camera movements. 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. Image-related sounds include *dialog, music, background sound, and sound effect*s. Sound editing refers to the mixing of all sound tracks and the integration of the final images with the sound track.

For example, to shoot the scene where two people are talking, over-the-shoulder shot (of camera1 or 2 in figure 3) created by *triangle principle* is usually used.

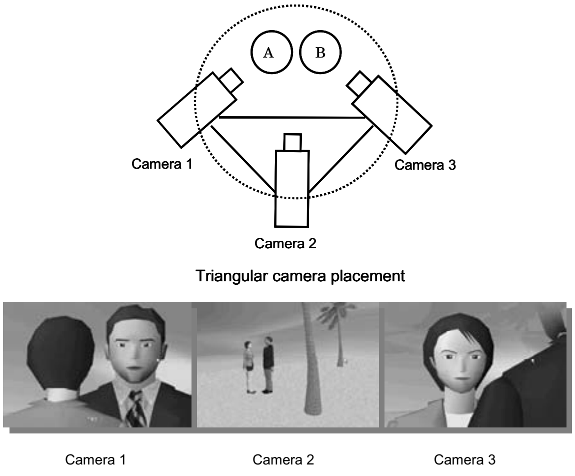


Figure 3: Triangle principle

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 another 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 [13, 14].

3.2 Knowledge Representation

Different types of KB have different roles for different goals [15]. When representing human expertise, it is usual to use *rules* to model domain knowledge. We built a filmmaking knowledge base having the representation, contents, and inference strategies. Each rule encodes a piece of filmmaking knowledge. Besides rules, knowledge representation still implies human behaviour of intelligent reasoning, i.e. determining which rule will be applied [16]. Representation and reasoning are inextricably intertwined with representation. We employed *forward-chaining* inference mechanics for the same way as in which human makes a plan.

Expert system shells using Rete algorithm include OPS5, ART, Rete++, CLIPS, Jess, and so forth. CLIPS (C Language Integrated Production System) is an expert system language with the properties of natural language style, uniform structure, and good extensibility, suitable to modelling human knowledge or expertise. It 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 fulfil the needs for high-level tool to program the generation of movie.

3.2.1 Rule-based Paradigm

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 can be written in *defrule* construct as:

```
(defrule shot-type “A rule of intention shot”
    ; track two persons in half length size,
    ; make a two-shot and a medium shot."
    (track-two-half -front)                      ; If
    =>                                           ; Then
    ; activate over the shoulder shot
    (assert (TS) (MS) (DS)))
    ;(assert (two-shot) (medium-shot) (dolly-shot)))
(defrule shot-select “A rule of shot selecting”
    ; there a conversation between two person
    ;one is talking
    (dialogue exist) (speak one)
    =>
    ;activate over the shoulder shot
    (assert (action OTS)))
```

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* (TS) (MS) (DS) that contain function calls. The example rule of shot selecting will be activated when the two facts (dialogue exist) and (speak one) appear in the knowledge base. When the rule executes, or *fires*, the function (action OTS) is called.

3.2.2 Object-based Paradigm

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)
     (parameter FLOAT) (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 Procedural-based Paradigm

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 *defule* sentence to express this meaning:

```
(defrule two-talk-distance
     character (name [Rose] (send ? stop)
     character (name [Tom])
     (send ? length   &: (< ? length (+ ? stop 2)
                           &: (> ? length (- ? stop 2)
     =>
     (send [Tom] walk 0)
```

3.3 Moviemaking Reasoning Procedure

See the usual steps in linear animation generation (figure 4). Camera works and sound can be set up at any stage.

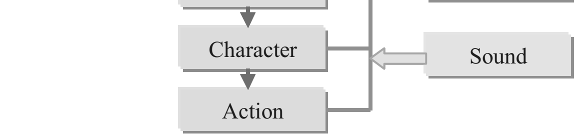


Figure 4: Control flow of moviemaking

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 modelling them in advance. If necessary, composition of animation and real image will be made.

4 Implementation with Example

A film is made up of shots arranged in sequence. The virtual director establishes a point of view on the action that helps to determine the selection of shots and camerawork through rule-based planning, timing out every shot 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 satisfy some conditions evaluated by objective function. The main data flowing along the moviemaking pipeline from the textual input to clip output is "screenplay → elements (event, sound etc.) → shots → shot sequence → scene".

4.1 Virtual Camera Works

Our virtual camera is modeled with seven Degrees of Freedom - three for Cartesian position, three for orientation, and FOV 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.

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). 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. In mathematical formula the camera operation can be expressed as:

Camera (x, y, z, vx, vy, vz, xadj, yadj, zadj,
roll, tilt, dolly, pan, rack, zoom, speed, transition)

where x, y, z are coordinates, vx, vy, vz are visual angles, xadj yadj, zadj are relative distances of movement, roll, tilt, pan dolly, track, zoom 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

For instance, OTS (Over-The-Shoulder) involves the techniques of mise-en-scène and cinematography.

Spatial constraints

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 line 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 facing directly on to character A.
   - The field of view should be between 20 and 60 degrees;

Expressing constrains

Write the above rules in CLIPS correspondingly:

```
(deftemplate character1
   "Information of character A"
   (slot name (type STRING) (default ?DERIVE))
   (slot location (type SYMBOAL) (default top-left))
   (slot orientation (type Number) (default 0)))
(deftemplate character2
   "Information of character B"
   (slot name (type STRING) (default ?DERIVE))
   (slot location (type SYMBOAL) (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

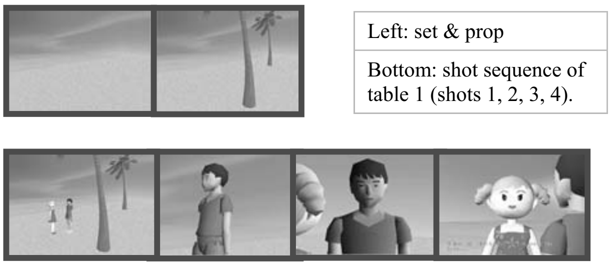
Heuristics about making a sequence of shots involves the techniques of montage and sound related to image, and another unit in film named event. *Event* is an important primitive action unit in camera planning procedure such as "a private conversation between two characters" (two-talk). Shot sequence 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: c*lose-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.

For example, to stage face to face two-talk, the virtual director determines five basic shots, 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 figure 5) while some may be never used (e.g. Angular shot).

Table 1: Staging dialogue sequence for two characters

Inference Procedure		
Premises	Actions	
Two talk	Set of basic shots	Rule
	two-shot & full shot (FS) profile-shot & close-up (CU) over-the-shoulder shot (OTS) point-of-view shot (POV) & close-up angular-shot & close-up	(a)
Silence	Shot sequence	Rule
	1. two-Shot FS	(b)
B talks	Shot sequence	Rule
	1. two-Shot FS 2. OTS (facing B)	(d) (e)
B talks	Shot sequence	Rule
	1. two-Shot FS 2. angular-shot CU 3. OTS (facing B)	(d) (f)
A talks	Shot sequence	Rule
	1. two-Shot FS 2. angular-shot CU 3. OTS (facing B) 4. OTS (facing A)	(c) (e)



Left: set & prop

Bottom: shot sequence of table 1 (shots 1, 2, 3, 4).

Figure 5: An example scene of two talk

5 Conclusion

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 recent years, filmic techniques have been extended to a degree possible with live actors shot in real time, first by means of animated film as seen in drawn or puppet animation, later by means of computer-generated images and animations. Current activities point to an interesting market potential for shared virtual environments where a knowledge-based approach to graphics generation is always essential. Cinematic knowledge base extracted from the group of film craftsmen can give great aid for the process of creating high quality motion pictures.

6 Acknowledgements

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.

7 References

[1] SHEN Jinhong, Seiya MIYAZAKI, Terumasa AOKI, Hiroshi YASUDA, "The Framework of an Automatic Digital Movie Producer", *2002 AVM Conference of IEICE, IEICE Technical Report,* Vol. 102, No. 517, Nagoya, Japan, (2002).

[2] Shen Jinhong, Seiya Miyazaki, Terumasa Aoki, Hiroshi Yasuda, "Virtual Camera Works with Planning Capabilities, *The 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI'03),* Vol. XV, Orland, Florida, USA (2003)

[3] Shen Jinhong, Seiya Miyazaki, Terumasa Aoki, Hiroshi Yasuda, "A Prototype of Cinematic Rule-based Reasoning and Its Application", *The 9th International Conference on Information Systems Analysis and Synthesis: ISAS '03 (CCCT2003)*, Vol. VI, Florida, USA (2003)

[4] Seiya Miyazaki, Shen Jinhong, Terumasa Aoki, Hiroshi Yasuda, "A System from Script to Motion Picture", *The 9th International Conference on Information Systems Analysis and Synthesis: ISAS '03 (CCCT2003),* Vol. IV, Florida, USA (2003)

[5] Christianson, Anderson, Wei-he, Salesin, Weld, and Cohen, "Declarative Camera Control for Automatic Cinematography", *AAAI/IAAI,* Vol. 1, Portland, Oregon, pp148-155 (1996)

[6] Li-wei He, Michael F. Cohen, David H. Salesin, "The virtual cinematographer: a paradigm for automatic real-time camera control and directing", *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, New Orleans, Louisiana, United States, pp217-224 (1996)

[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)

[8] Szarowicz, A., Amiguet-Vercher, J., Forte, P., Briggs, J., Gelepithis, P.A.M., Remagnino, P. "The Application of AI to Automatically Generated Animation", *Australian Joint Conference on Artificial Intelligence*, *AI'01, AI 2001:Advances in Artificial Intelligence,* Adelaide, pp487-494 (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)

[10] Stefan Kopp, Ipke Wachsmuth, "A Knowledge-based Approach for Lifelike Gesture Animation", *In W. Horn, editor, ECAI 2000 Proceedings of the 14th European Conference on Artificial Intelligence,* IOS Press, Amsterdam, pp120-133 (2000)

[11] John Funge, Xiaoyuan Tu, Demetri Terzopoulos, "Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters", *Computer Graphics Proceedings*, Siggraph 1999, Los Angeles, USA, pp29-38 (1999)

[12] Foley, Van Dam, Feiner, Hughes, "Computer Graphics: Principles and Practice", *2nd edition in C*, Addison-Wesley, (1996)

[13] Thompson R., *Grammar of the Shot*, Oxford: Focal Press, (1998)

[14] Steven D. Katz, *Film Directing Shot by Shot: Visualizing from Concept to Screen*, Michael Wiese Productions and Focal Press (1992)

[15] Malgorzata Ochmanska & Mieczyslaw L. Owoc**,** **"**Verification of Different Knowledge Bases, Knowledge Acquisition and Distance Learning for Supporting Managerial Issues", Proc. of the International Seminar in Krzyzowa, Malardalens University (2001)

[16] Randall Davis, Howard Shrobe, & Peter Szolovits, "What is a Knowledge Representation?", *AI Magazine*, 14(1), pp17-33 (1993)