Pontifícia Universidade Católica
do Rio de Janeiro

Edirlei Everson Soares de Lima

Video-Based Interactive Storytelling

TESE DE DOUTORADO

DEPARTAMENTO DE INFORMÁTICA
Programa de Pós-Graduação em Informática

Rio de Janeiro
August 2014

Pontifícia Universidade Católica
do Rio de Janeiro

Edirlei Everson Soares de Lima

Video-Based Interactive Storytelling

TESE DE DOUTORADO

Thesis presented to the Programa de Pós-Graduação em Informática of the Departamento de Informática, PUC-Rio as partial fulfillment of the requirements for the degree of Doutor em Informática

Advisor: Prof. Bruno Feijó

Rio de Janeiro
August 2014

Pontifícia Universidade Católica
do Rio de Janeiro

Edirlei Everson Soares de Lima

Video-Based Interactive Storytelling

Thesis presented to the Programa de Pós-Graduação em Informática, of the Departamento de Informática do Centro Técnico Científico da PUC-Rio, as partial fulfillment of the requirements for the degree of Doutor.

Prof. Bruno Feijó
Advisor
Departamento de Informática - PUC-Rio

Prof. Simone Diniz Junqueira Barbosa
Departamento de Informática - PUC-Rio

Prof. Helio Côrtes Vieira Lopes
Departamento de Informática - PUC-Rio

Prof. Angelo Ernani Maia Ciarlini
UNIRIO

Prof. Sean Wolfgand Matsui Siqueira
UNIRIO

Prof. José Eugenio Leal
Coordinator of the Centro Técnico Científico da PUC-Rio

Rio de Janeiro, August 4th, 2014

Edirlei Everson Soares de Lima

Graduated in Computer Science at Universidade do Contestado (2008), and received his Master Degree in Computer Science from Universidade Federal de Santa Maria (2010). He joined the Doctorate program at PUC-Rio in 2010, researching on interactive storytelling, games, artificial intelligence and computer graphics. In 2011 and 2012, his research on video-based interactive storytelling received two honorable mentions from the International Telecommunication Union (ITU).

Bibliographic data

Lima, Edirlei Everson Soares de

Video-Based Interactive Storytelling / Edirlei Everson Soares de Lima ; Advisor: Bruno Feijó – 2014.

218 f. : il. (color.) ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2014.

Inclui bibliografia

1. Informática – Teses. 2. Storytelling Interativo. 3. Dramatização Baseada em Vídeo. 4. Composição de Vídeo. 5. Cinematografia Virtual. I. Feijó, Bruno. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Acknowledgments

First and foremost, I would like to express my deepest and sincerest gratitude to my advisor, Prof. Bruno Feijó, for his constant guidance, support and incentive during all these years of research. Besides being an outstanding advisor, he has been like a father and a very good friend to me. My appreciation for his continuous support in all the aspects of my research is immeasurable.

This thesis would also not have been possible without the expert guidance and support of many of my unofficial advisors. First, I would also like to thank Prof. Antonio Furtado for constantly guiding me with his extensive knowledge and enthusiasm. My grateful thanks also to Prof. Cesar Pozzer for his constant guidance and insightful discussions. Special thanks to Prof. Simone Barbosa for her precious advices on Human-Computer Interaction. And I would also like to thank Prof. Angelo Ciarlini for his support and collaboration on many of my research works.

I would also like to thank all the people that have been involved in the production of the prototype video-based interactive narratives. Special thanks to Marcelo Feijó for his excellent work in writing the scripts and directing the production of the interactive narratives; and to Bruno Riodi for his great work in editing and preparing the video material.

I would also like to thank CAPES (Coordination for the Improvement of Higher Education Personnel, linked to the Ministry of Education) and CNPq (National Council for Scientific and Technological Development, linked to the Ministry of Science, Technology, and Innovation) for the financial support for this research. Special thanks to the staff of the Department of Informatics (PUC-RIO) and to the ICAD/VisionLab for providing an excellent research environment.

Finally, I would also like to thank my parents for their constant support and encouragement.

Abstract

Lima, Edirlei Everson Soares de. **Video-Based Interactive Storytelling.** Rio de Janeiro, 2014. 218p. DSc Thesis - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The generation of engaging visual representations for interactive storytelling represents a key challenge for the evolution and popularization of interactive narratives. Usually, interactive storytelling systems adopt computer graphics to represent the virtual story worlds, which facilitates the dynamic generation of visual content. Although animation is a powerful storytelling medium, live-action films still attract more attention from the general public. In addition, despite the recent progress in graphics rendering and the wide-scale acceptance of 3D animation in films, the visual quality of video is still far superior to that of real-time generated computer graphics. In the present thesis, we propose a new approach to create more engaging interactive narratives, denominated "Video-Based Interactive Storytelling", where characters and virtual environments are replaced by real actors and settings, without losing the logical structure of the narrative. This work presents a general model for interactive storytelling systems that are based on video, including the authorial aspects of the production phases, and the technical aspects of the algorithms responsible for the real-time generation of interactive narratives using video compositing techniques.

Keywords

Interactive Storytelling; Video-Based Dramatization; Video Compositing; Virtual Cinematography.

Resumo

Lima, Edirlei Everson Soares de. **Storytelling Interativo Baseado em Vídeo.** Rio de Janeiro, 2014. 218p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A geração de representações visuais envolventes para storytelling interativo é um dos desafios-chave para a evolução e popularização das narrativas interativas. Usualmente, sistemas de storytelling interativo utilizam computação gráfica para representar os mundos virtuais das histórias, o que facilita a geração dinâmica de conteúdos visuais. Embora animação tridimensional seja um poderoso meio para contar histórias, filmes com atores reais continuam atraindo mais atenção do público em geral. Além disso, apesar dos recentes progressos em renderização gráfica e da ampla aceitação de animação 3D em filmes, a qualidade visual do vídeo continua sendo muito superior aos gráficos gerados computacionalmente em tempo real. Na presente tese propomos uma nova abordagem para criar narrativas interativas mais envolventes, denominada "Storytelling Interativo Baseado em Vídeo", onde os personagens e ambientes virtuais são substituídos por atores e cenários reais, sem perder a estrutura lógica da narrativa. Este trabalho apresenta um modelo geral para sistemas de storytelling interativo baseados em vídeo, incluindo os aspectos autorais das fases de produção e os aspectos técnicos dos algoritmos responsáveis pela geração em tempo real de narrativas interativas usando técnicas de composição de vídeo.

Palavras-chave

Storytelling Interativo; Dramatização Baseada em Vídeo; Composição de Vídeo; Cinematografia Virtual.

Contents

List of Figures

List of Tables

1
Introduction

Since immemorial times, humans have been telling stories. What started out as short stories about hunts and tales of ancestors, soon evolved to myths and legends. Over centuries, stories played an important role in human society and were used to teach, inspire, and entertain. With the advent of new technologies, new forms of storytelling were created. Currently, stories are told through several types of media such as books, movies and games.

Video games were introduced to the general public in the early 1970s and quickly became a form of digital storytelling that added interactivity to the traditional stories, allowing players to be the protagonists in a new form of digital entertainment. With the advancement and popularization of video games, a long discussion about the relationship between games and narratives has taken place (Jensen 1988; Juul 1998; Adams 1999; Costikyan 2000; Jenkins 2003). Meanwhile, a new research topic exploring the combination of storytelling with interactivity emerged (Meehan 1981; Loyall and Bates 1991; Szilas 1999). Soon it became the field of research that today is known as Interactive Storytelling or Interactive Narrative.

The first interactive narratives date back to the 1970s (Klein et al. 1973; Meehan 1977) and important experiments on agent-based storytelling systems took place in early 1990s (Loyall and Bates 1991). In the 2000s we can find the most influential research works on interactive storytelling systems (Cavazza et al. 2002; Mateas 2002). In more recent years, we have been exposed to new demands for richer interactive experiences in storytelling, such as transmedia storytelling (Cheshire and Burton 2010), social interaction between groups (Williams et al. 2011), and interactive TV storytelling (Ursu et al. 2008).

In parallel with the evolution of interactive narratives, cinema has been promoting new forms of immersive experience since the advent of projected motion pictures in the late 19th century. Cinema has evolved from the silent black-and-white film to the high-definition stereoscopic 3D projection. Recently,

films with interactive plots have been proposed as a new experience (Činčera et al. 1967; Pellinen 2000; Ursu et al. 2008; Jung von Matt 2010). However, most of these experiences are based on the concept of branching narrative structures (Samsel and Wimberley 1998), which are known in the area of interactive storytelling as having several limitations, such as the authoring complexity and the lack of story diversity. Research on interactive storytelling has been exploring the generation of interactive narratives since the 1970s and may provide the proper foundation for the creation of a new form of interactive cinema. However, this research area has few works oriented to motion pictures.

The most robust forms of interactive narratives rely on artificial intelligence techniques, such as planning (Ghallab et al. 2004), to dynamically generate the sequence of narrative events rather than following predefined branching points. The techniques that support the dynamic generation of stories are also useful to maintain the coherence of the entire narrative. Moreover, they support the propagation of changes introduced by the users, allowing them to effectively interact and change the unfolding stories. Although artificial intelligence techniques can help to improve the diversity of stories, they face the challenge of generating in real-time a visual representation for a story that is not known beforehand. In branching narratives, all the possible storylines are predefined by the author, and the system is prepared to represent them in the best possible way. On the other hand, in systems based on planning techniques, stories are created by the planning algorithm, guided to some extent by the user interactions, and it is not easy to predict all the possible storylines that can emerge. These unpredictable outcomes require intelligent systems capable of adapting themselves to represent emergent narratives.

Despite the large amount of research works in the field of interactive storytelling, there are still some open issues (Karlsson 2010; Zhao 2012). One of the main challenges, and the focus of this thesis, is the generation of engaging visual representations for interactive narratives. Interactive storytelling systems usually employ 2D or 3D computer graphics to represent the virtual story worlds (Mateas 2002; Cavazza et al. 2002; Ciarlini et al. 2005; Pizzi and Cavazza 2007). This approach provides a visual medium that facilitates the dynamic generation of visual content, providing the freedom to move the characters and cameras to any place in the virtual world, allowing the system to show the story events from any

angle or perspective. Furthermore, virtual characters may have their properties easily changed, such as shape, facial expressions, clothes, and behaviors. However, this freedom usually sacrifices the visual quality of the real-time narrative. Despite the recent progress in graphics rendering and the wide-scale acceptance of 3D animation in films, the visual quality of video is still far superior to that of real-time generated computer graphics.

Although animation is a powerful storytelling medium, live-action films are still attracting more attention from the general public. A promising approach that may be the first step to bring interactive narratives to the big screens is the replacement of 2D/3D virtual characters by video sequences in which real actors perform predetermined actions. This approach, which we propose to call "video-based interactive storytelling", has showed some interesting results in recent years (Ursu et al. 2008; Porteous et al. 2010; Piacenza et al. 2011; Jung von Matt 2010). However, most of those results either are domain-specific applications based on branching narrative structures or do not have the intention of presenting a general approach to handle all problems of a generic video-based interactive story.

The main problem of using videos to dramatize an interactive narrative is the lack of freedom occasioned by immutable prerecorded segments of videos, which reduces interactivity, limits story diversity, and increases production costs. For example, let us suppose a story that includes a kidnap event, where the victim can be kidnapped by the villain in different locations depending on the previous events of the story and user interventions. If only prerecorded scenes are used for dramatization, every possible variation of the kidnap has to be filmed. Consequently, the production costs of the interactive film will be multiplied by the number of storylines that can be generated by the system. Usually, the number of possible stories ends up being limited in order to reduce production costs.

This thesis proposes a new approach to video-based interactive narratives that uses real-time video compositing techniques to dynamically create video sequences representing the story events – rather than proposing a simple method that merely assembles prerecorded scenes. This approach allows the generation of more diversified stories and reduces the production costs. However, it requires the development of fast and intelligent algorithms, capable of applying cinematography techniques to create cinematic visual representations for the story events in real-time.

Advances of interactive storytelling technology towards new media, such as television and cinema, also require the development of new interaction mechanisms that consider the characteristics of the media platform that supports the story. In television, for instance, interaction systems should consider one or few local viewers (in the same room) or thousands of viewers sharing the same story at different places. In movies, the audience is restricted to the theater space, but is still a multi-user environment. Video-based interactive narratives designed either for TV or cinema require new interaction mechanisms that support multi-user interactions. These interaction issues are also addressed by the present thesis.

1.1. Objectives

This thesis aims at the generation of more engaging visual representations for interactive narratives by using video segments with real actors to represent story events. We argue that this approach can generate interactive narratives that resemble traditional movies while keeping the possibilities of user interaction even when the plot generation algorithms produce scenes that were not foreseen during the production stage.

The main objective of this thesis is to propose a general model for video-based interactive storytelling, including the organizational aspects of the production pipeline, the technical aspects of the algorithms responsible for the real-time generation of video-based interactive narratives, and the usability of the user interfaces.

1.2. Contributions

The main contributions of this thesis are summarized below (more details on specific contributions are presented in Chapter 9):

- Proposes a model for video-based interactive storytelling based on cinematography theory;
- Presents new algorithms and techniques for real-time video compositing and editing in video-based interactive narratives;

- Proposes new interaction mechanisms that support multi-user interaction in video-based interactive narratives;
- Reduces the gap between interactive storytelling systems and film directors by proposing a guide and computational tools for the production of video-based interactive narratives.

1.3. Thesis Structure

Chapter 2 presents the main concepts of interactive storytelling and a bibliographic review of the main interactive storytelling systems, emphasizing the methods used by these systems to visually represent interactive stories. It also includes a detailed description of Logtell, which is the interactive storytelling system used as basis for developing the proposed video-based dramatization model.

Chapter 3 reviews some essential concepts of cinematography that are important for the development of a video-based interactive storytelling system. The cinematography theory provides the basic principles and background for the creation of attractive and engaging video-based visual representation of interactive stories.

Chapter 4 presents the proposed architecture of the video-based interactive storytelling system from a software engineering perspective. It also discusses related work and describes the main differences between the proposed system and previous work.

Chapter 5 presents the proposed process of production of interactive narratives, describing how to write and how to film an interactive story. This chapter also describes some tools that were developed to assist the author during the production process.

Chapter 6 describes the technical details about the implementation of the proposed video-based dramatization system, including the proposed algorithms for real-time video compositing and editing.

Chapter 7 describes the technical details about the implementation of the user interaction module of the video-based interactive storytelling system, including the proposed multi-user interaction mechanisms.

Chapter 8 describes the interactive narratives that were produced to validate the proposed system and presents some technical tests to evaluate the algorithms used in the video-based interactive storytelling system.

Chapter 9 presents the conclusions remarks, summarizes the contributions and suggests topics for future research work.

2
Interactive Storytelling

Interactive storytelling is a form of digital entertainment based on the combination of interactivity and storytelling. Interactive storytelling systems aim to create dramatic and engaging narrative experiences for users, while allowing them to intervene with ongoing plots and change the way that the story unfolds. One of the key challenges in the development of such systems is how to balance a good level of interactivity with the consistency of the generated stories.

Although at first glance interactive storytelling may seem similar to digital games, there is a clear difference between them. In games, stories are essentially used to create challenges for the players, whereas in interactive storytelling stories are created to surprise and to entertain spectators. In both forms of media, users are able to intervene with the ongoing stories in some way. Usually, in interactive storytelling the interaction occurs only at specific points of the story and does not require much effort and attention from users. Although digital games and interactive storytelling are two different forms of media, the field of research in interactive storytelling may offer the opportunity for the development of new forms of digital games. The recent commercial success of the game Heavy Rain (2010) reveals that integrating interactive stories into the gameplay creates a successful experience for players.

Since the advent of the first interactive storytelling systems, a number of techniques and applications have been proposed in an effort to create engaging narrative experiences for users. Particularly, two models, character-based and plot-based, have been widely used in most current interactive storytelling systems. In the character-based approach (Meehan 1977; Cavazza et al. 2002; Aylett et al. 2005; Lima et al. 2014A), a simulation of a virtual world is created and the stories result from the real-time interaction among virtual autonomous agents that inhabit the virtual world and incorporate a deliberative behavior. The main advantage of a character-based model is the ability of anytime user intervention. As a result of such strong intervention, there is no way to estimate what decisions or actions will

be made by the virtual actors, which is likely to lead the plot to unexpected situations that can violate the coherence of the story. The second model corresponds to the plot-based approach (Grasbon and Braun 2001; Paiva et al. 2001; Spierling et al. 2002; Magerko and Laird 2003; Magerko 2005), where characters incorporate a reactive behavior, which follows rigid rules specified by a plot. The plot is usually built in a stage that comes before dramatization. This approach ensures that actors can follow a predefined script of actions that are known beforehand. The plot may be automatically generated through planning algorithms or manually built by the author of the story. However, a plot-based approach restrains the user's freedom of interaction. Usually, the user has to wait until the storyline reaches some predefined points to be able to intervene. There are also some hybrid systems (Mateas 2002; Ciarlini et al. 2005; Cai et al. 2007; Si et al. 2008) that combine characteristics of both character-based and plot-based approaches using authoring goals and modeling the behavior of the virtual character in an attempt to reduce the shortcomings of both approaches.

A narrative is generally defined as a series of events that tells a story, either fictional or non-fictional. Interactive narratives can be entirely created manually by an author or automatically generated by planning techniques and simulations. The first case represents the more simple form of interactive stories. Usually, a hand-crafted structure of nodes, often in the form of a graph, defines the possible storylines. Each node of the graph includes a finely-crafted description of the plot event and the connections between nodes represent the possible paths that the story can follow. The user is given the ability to navigate through the graph, and the resulting sequence of nodes constitutes the experience of the narrative. Every possible storyline is manually authored, which ensures the author's vision is precisely preserved. However, the amount of narrative content that must be authored can grow exponentially with the number of user choices and the authoring process of large graphs quickly becomes intractable.

Figure 2.1 shows an example of a complex graph that defines every possible narrative trajectory of the Choose Your Own Adventure book *The Mystery of Chimney Rock* (Packard 1979), which contains 36 possible endings. Choose Your Own Adventure books were originally created by Edward Packard in the 1970s. These books contain a finite number of plot lines and narrative paths. At critical moments in the story, the reader is prompted to play the role of the characters and

make a choice between several possible actions, which will direct him/her to another page of the book. Examples of manually authored interactive narratives also include some adventure games and some recent interactive experiences in TV and Cinema.

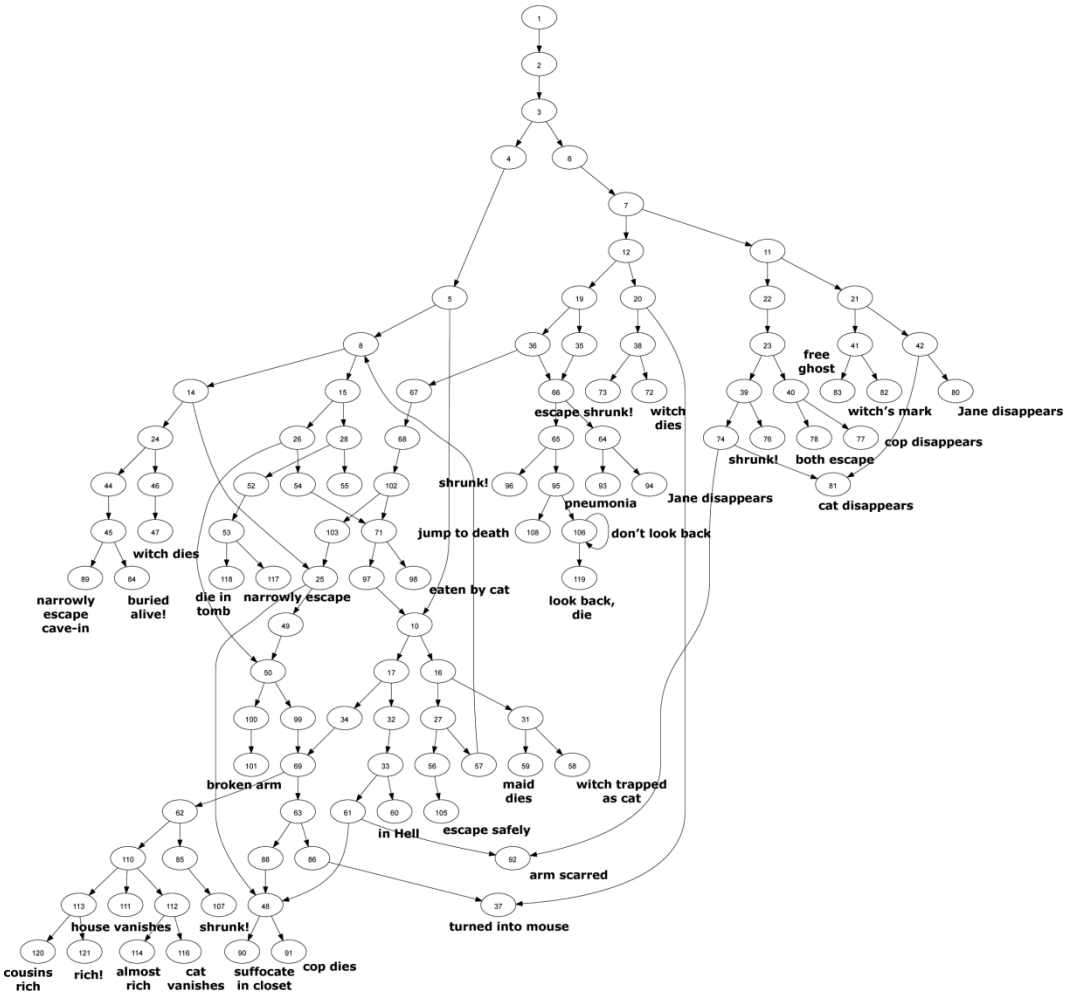


Figure 2.1: Story graph of The Mystery of Chimney Rock (Packard 1979).

Interactive narratives that have plots entirely or partially generated by planning techniques or simulations are the most robust forms of interactive storytelling. In a plot-based approach, the author usually defines only the story characters, a set of narrative events with preconditions and effects, and an initial state of the world. Then, a planning algorithm is responsible for finding coherent sequences of narrative events that will form the story. The advantage of this approach is that the planning algorithms can guarantee the story coherence and

avoid the author from having to handle complex graph structures. A drawback is the exponential complexity of the planning algorithms, which require optimization techniques to allow the generation of plots in real-time. In a character-based approach, the author has to encode the behavior, personality traits and intents of virtual autonomous characters in a virtual environment. Then, a simulation process occurs and the storylines result from the interaction between characters and users. The advantage of this approach is the large number of stories that can emerge from the characters interactions, but there is no easy way to guarantee that all of these stories will be complex enough to create an interesting drama. Character-based approaches can also incorporate planning techniques in the simulation of the virtual world to define the behavior of the autonomous characters, as in Cavazza et al. (2002), where each character consults pre-compiled plans encoded as a Hierarchical Task Network (HTN) in order to decide the actions to be performed.

The field of research on interactive storytelling can be divided into three lines of research: story generation, user interaction, and story dramatization. Story generation aims at creating methods for the generation and management of coherent and diversified stories. User interaction aims at designing interaction mechanisms and interfaces that allow users to intervene with ongoing plots and change the way that the story unfolds. Story dramatization aims at generating attractive and engaging visual representations for the narratives.

The origins of the research topic about story generation date back to the 1970s (Meehan, 1977) and several techniques to accomplish this task have been proposed throughout the years. Usually, the process of generating stories involves narrative theories and planning algorithms. The narrative theory gives the formalism on how the story is structured and the planning algorithms are responsible for generating coherent sequences of events to compose the story plot. The formalism proposed by Propp (1968) is an example of narrative theory adopted by several interactive storytelling systems (Prada et al. 2000; Grasbon and Braun 2001; Ciarlini et al. 2005; Szilas 2007]. Propp examined 100 Russian fairy tales, and showed that they could all be described by 31 typical narrative functions, such as villainy, hero’s departure, reward, etc. Propp also showed that these functions have a chronological order that defines the basic structure of a fairy tale. This formalism is commonly used by story generator systems to

produce the basic structure of the narratives. The planning techniques commonly used in interactive storytelling systems include Hierarchical Task Networks (HTN) (Cavazza et al. 2002), Heuristic Search Planners (HSP) (Pizzi and Cavazza 2007; Lima et al. 2013), and first order logic planners based on the STRIPS formalism (Szilas 1999; Ciarlini et al. 2005).

The research topic about user interaction investigates new ways for users to intervene with interactive narratives. Several approaches to handle user interactions have been proposed through the years. The forms of interaction vary from traditional GUI interfaces (Grasbon and Braun 2001; Ciarlini et al. 2005) to more complex interaction mechanisms, such as speech recognition (Cavazza et al. 2002; Cavazza et al. 2009), body gestures combined with speech (Cavazza et al. 2004; Cavazza et al. 2007; Lima et al. 2011B), hand-drawn sketches (Kuka et al. 2009; Lima et al. 2011A), physiological inputs (Gilroy et al. 2012), and interaction through social networks (Lima et al. 2012B). One of the key challenges in the development of a user interface for interactive storytelling is how to balance a simple and transparent interface that does not distract users from the dramatic content of the narrative with the need of a robust interaction mechanism that does not restrict the user creativity.

The research topic about story dramatization investigates new forms to visually represent interactive narratives. Text was the most common form of dramatization used by the first interactive storytelling systems like Tale-Spin (Meehan 1977; Meehan 1981), Universe (Lebowitz 1984; Lebowitz 1985) and Ministrel (Turner 1992). Usually, the process of representing stories through text consists of translating the story events into written natural language. The second generation of interactive storytelling systems that emerged in early 2000s was based on 2D or 3D computer graphics (Bates 1994; Mateas 2002; Cavazza et al. 2002; Aylett et al. 2005; Magerko 2006; Mott and Lester 2006; Pizzi and Cavazza 2007; El-Nasr 2007). In this form of dramatization, the story events are represented through animations of 3D models or 2D images. Other forms of dramatization include videos (Mateas et al. 2000; Ursu et al. 2008; Porteous et al. 2010; Lima et al. 2012A), augmented reality (Dow et al. 2006; Zhou et al. 2008; Lima et al. 2011A), and comics (Lima et al. 2013).

2.1. Story Dramatization

The following sections present a bibliographic review of the methods employed by the most relevant interactive storytelling systems to visually represent interactive stories.

2.1.1. Text-Based Dramatization

Tale-Spin (Meehan 1977; Meehan 1981) was one of the first computer programs created to automatically generate stories. The system is based on a character-based approach and it generates narratives by simulating a virtual world where characters try to reach their goals. Stories are created by a planning algorithm that is responsible for generating a plan that will be used by the characters. Once generated, this plan is translated into written natural language and then presented to the user. An example of story generated by Tale-Spin is shown in Figure 2.2.

Once upon a time George ant lived near a patch of ground. There was a nest in an ash tree. Wilma bird lived in the nest. There was some water in a river. Wilma knew that the water was in the river. George knew that the water was in the river. One day Wilma was very thirsty. Wilma wanted to get near some water. Wilma flew from her nest across the meadow through a valley to the river. Wilma drank the water. Wilma wasn't thirsty anymore.

George was very thirsty. George wanted to get near some water. George walked from his patch of ground across the meadow through the valley to a river. George fell into the water. George wanted to get near the valley. George couldn't get near the valley. George wanted to get near the meadow. George couldn't get near the meadow. Wilma wanted to get near George. Wilma grabbed George with her claw. Wilma took George from the river through the valley to the meadow. George was devoted to Wilma. George owed everything to Wilma. Wilma let go of George. George fell to the meadow. The end.

Figure 2.2: Example of story generated by Tale-Spin (Meehan 1981).

In Tale-Spin the process of translating the story events into natural language consists of putting the actor, action and object in the correct position of the sentence (Lee 1994). For each verb used by the program, there is a lexical entry which details the corresponding conjugations of the verb for singular and plural in the past, present and future tenses. The process is entirely ad hoc and is not based on any recognized linguistic theory (Meehan 1981). Therefore, the range of sentences which can be generated is limited.

Even though Tale-Spin is able to generate some interesting stories, most of them are not dramatically interesting. Despite the characters being coherent, stories have no structure and can turn out to be too short (Lee 1994). The translation of the story events into natural language generates very simple sentences that do not attract the reader's attention and are somewhat difficult to read.

Universe (Lebowitz 1984; Lebowitz 1985) is another interactive storytelling system that follows the same idea of Tale-Spin. However, instead of using characters' goals, the planning process is based on authorial goals. This is done by using plot fragments that contains a list of roles to be filled up by the characters, a set of restrictions and consequences, and an ordered list of sub-goals that must be achieved to satisfy the plot fragment. The characters of the stories are defined by personality traits, stereotypes, relations to other characters, and are responsible for assuming roles in the plot fragments. A large number of characters ensures that most situations will have a suitable character. As happens in Tale-Spin, once a plan for the story is generated it is translated into written natural language and then shown to the user. An example of story generated by Universe is shown in Figure 2.3.

Universe does not deal with the problem of natural language generation. The process of translating the story events into text is based on the use of pre-defined phrases and templates to generate the story output. This approach produces good results because the phrases and templates are prepared and written by humans. However, for every new story context, new templates have to be created.

Minstrel (Turner 1992) is another story generation system based on planning. However, different from Tale-Spin and Universe, it uses a planning technique called Case-Based Reasoning (Aamodt and Plaza 1994), where pieces of previously known or pre-generated stories are used in the generation of new

ones. The process is based on four types of authorial goals: thematic, dramatic, consistency and presentation goals. Thematic goals are concerned with the selection and development of the theme and purpose of the story. Dramatic goals are concerned with keeping the story interesting by generating suspense, tragedy, presages, etc. Story consistency concerns the credibility of the actions performed by the characters. Finally, presentation goals are concerned with how the story is presented in natural language. An example of story generated by Minstrel is shown in Figure 2.4.

Liz was married to Tony. Neither loved the other, and, indeed, Liz was in love with Neil. However, unknown to either Tony or Neil, Stephana, Tony's father, who wanted Liz to produce a grandson for him, threatened Liz that if she left Tony, he would kill Neil. Convinced that he was serious by a bomb that exploded near Neil, Liz told Neil that she did not love him, that she was still in love with Tony, and that he should forget about her. Eventually, Neil was convinced and he married Marie. Later, when Liz was finally free from Tony (because Stephana had died). Neil was not free to many her and their troubles went on.

Figure 2.3: Example of story generated by Universe (Lebowitz 1985).

Minstrel uses a phrasal parser (Arens 1986; Reeves 1989) to generate written natural language from the story events. The parser integrates syntactic and semantic information into a lexicon of phrases that pair concepts and words. To translate the story events into natural language, the concepts from the story are matched with the lexical entries, and when a match is found, the corresponding words are outputted.

Tale-Spin, Universe and Minstrel are the most relevant text-based interactive storytelling systems found in the literature; however, there are others interesting systems such as Brutus (Bringsjord and Ferrucci 1999), which writes short stories about pre-defined themes, and Mexica (Pérez y Pérez 1999; Pérez y Pérez and Sharples 2001), which produces short stories based cycles of engagement and reflection.

It was the spring of 1089, and Lancelot returned to Camelot from elsewhere. Lancelot was hot tempered. Once, Lancelot had lost a joust. Because he was hot tempered wanted to destroy his sword. Lancelot stuck his sword. His sword was destroyed.

One day a lady of the court named Andrea wanted to have some berries.

Andrea went to the woods. Andrea had some berries because Andrea picked some berries. Lancelot's horse moved Lancelot to the woods. This unexpectedly caused him to be near Andrea. Because Lancelot was near Andrea Lancelot saw Andrea. Lancelot loved Andrea.

Some time later. Lancelot's horse moved Lancelot to the woods unintentionally, again causing him to be near Andrea. Lancelot knew that Andrea kissed a knight called Frederick because Lancelot saw that Andrea kissed with Frederick Lancelot believed that Andrea loved Frederick. Lancelot loved Andrea. Because Lancelot loved Andrea, Lancelot wanted to be the love of Andrea. But he could not because Andrea loved Frederick. Lancelot hated Frederick. Because Lancelot was hot tempered, Lancelot wanted to kill Frederick. Lancelot went to Frederick. Lancelot fought with Frederick. Frederick was dead.

Andrea went to Frederick. Andrea told Lancelot that Andrea was siblings with Frederick. Lancelot wanted to take back that he wanted to kill Frederick. But he could not because Frederick was dead. Lancelot hated himself. Lancelot became a hermit. Frederick was buried in the woods. Andrea became a nun.

Figure 2.4: Example of story generated by Ministrel (Turner 1992).

2.1.2. 2D/3D Dramatization

The second generation of interactive storytelling systems that emerged in early 2000s was based on 2D or 3D computer graphics. In this form of dramatization the story events are represented through animations of 3D models or 2D images. However, this process is not an easy task; representing a narrative graphically involves several challenges. The characters must be believable and attract the attention of the audience, the environment must be rich and coherent with the story, and the camera must be intelligent to correctly show the scenes and improve the dramatic content of the narrative.

The game industry deals with interactive computer graphics since the advent of the first video games. However, interactive narratives are different from traditional games. Usually in a game every little detail is planned by the game designer and created by the artists and programmers to be always attractive and works as expected. In real emergent interactive narratives all the possible storylines that can be generated based on the user choices are not easily predictable. Different from traditional games, interactive narratives demand a high degree of adaptability and intelligent methods to produce attractive and engaging visual representations for the narratives.

The first graphical interactive storytelling system was developed in the Oz Project,[1] as part of their experiments on agent-based storytelling (Loyall 1997). The Edge of Intention (Bates 1992) (Figure 2.5 - *a*) is a virtual world that contains three autonomous animated creatures, called Woggles. Each Woggle has goals, emotions, and personality, and expresses these through movement and facial expression. They also engage in simple social games, exhibit aggression, play, sleep, and perform several other behaviors.

One of the main components of the Edge of Intention is a agent language called HAP (Loyall and Bates 1991). The language directly supports goal-directed action producing behaviors and continuously chooses the agent's next action based on perceptions, current goals, emotional state and aspects of an internal state. The HAP architecture also allows the parallel execution of multiple actions and the early production of next action to allow smooth animations and more believable and engaging dramatizations (Loyall and Bates, 1993).

Based on the ideas of the Oz Project, Mateas (2002) developed another interactive storytelling system called Façade. The main goal of Façade was to provide a complete real-time dramatic experience with a highly interactive story. The story generation in Façade is based on small plot units called *beats*. Each beat consists of a set of pre-conditions, a pre-scripted sequence of events, and a set of effects. The beats are sequenced in such a way as to be responsive to user interactions while providing story structure (Mateas 2002). The stories are represented in a 3D environment through a first-person perspective. An example of scene from Façade is shown in Figure 2.5 - *b*.

[1] OZ Project - https://www.cs.cmu.edu/afs/cs/project/oz/web/

Façade's real-time rendered 3D story world is implemented in C++ with OpenGL (Mateas and Stern 2003). The animation of character's body and face is done through a mixture of procedural animation and layered keyframe animation data. Each of the characters contains a library of behaviors. The pre-scripted beats control the animation of a character's body by issuing commands (e.g. "play this animation script" or "assume this facial expression").

Façade is known as the most successful attempt to create a real interactive drama (Crawford 2004). However, its architecture requires great authorial effort to create new interactive narratives. The authors spent 2 years to create the narrative that has only one scene, two characters and takes about 20 minutes to complete (Mateas and Stern 2003).

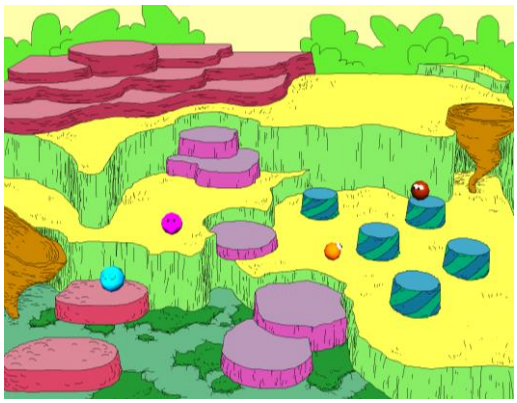
Following a different approach, Cavazza et al. (2002) formalize the concepts of character-based interactive storytelling and presented a system where each character uses a Hierarchical Task Network (HTN) in order to decide the actions to be performed. The HTN consists of a tree-like structure that describes the actions a character can perform in order to achieve his goals. Users can interact with the characters and navigate through their environment or they can verbally interact with them using a speech recognition system. The stories are represented in a 3D environment, but different from Façade, the user assumes a third-person perspective. An example of scene from Cavazza et al.'s system is shown in Figure 2.5 - *c*.

The system was developed using the Unreal Tournament game engine[2] and the story events are represented through animation and subtitles corresponding to the characters' dialogue or important events (Cavazza et al. 2002; Charles and Cavazza 2004). The system incorporates an intelligent camera control to decide which event should be shown to the user when different events occur at different locations at the same time (Charles et al. 2002). This decision is based on the type of event, participating characters, and emotional information.

In a more recent work, Pizzi and Cavazza (2007) present another version of their interactive storytelling system based on a different planning approach. They aim at reconciling narrative actions with the psychological state of characters. The best actions to be applied for a given character are provided by a function of

[2] Unreal Engine - http://www.unrealengine.com/

his/her current feelings, its beliefs, and the world's state of affairs. A multithreaded heuristic search planner is used for controlling each character independently and 3D animations are generated from the grounded actions produced by the planner. During story visualization, the system accepts Natural Language input, which is analyzed to update characters' beliefs and emotional state, thus altering the evolution of the narrative. The Unreal Tournament game engine is used to generate the real-time 3D graphics. An example of scene from Pizzi and Cavazza interactive storytelling system is shown in Figure 2.5 - *d*.



(*a*) Edge of Intention (1992)

(*b*) Façade (2002)

(*c*) Cavazza et al. System (2002)

(*d*) Pizzi and Cavazza System (2007)

Figure 2.5: Graphical interactive storytelling systems. Image (*a*) shows a scene from the Edge of Intention; image (*b*) shows a scene from Façade; image (*c*) shows a scene from Cavazza's interactive storytelling system; and image (*d*) shows a scene from Pizzi and Cavazza interactive storytelling system.

The story worlds created by Mateas (2002) and Cavazza et al. (2002) are the most relevant graphical interactive storytelling systems present in the literature, however there are other important systems like FearNot! (Aylett et al. 2005),

which follows a pure character-based approach using a cognitive and emotional model of human behavior. The narratives presented by FearNot! consist of educational stories about bullying, in which users are able to interact with the victims, giving them advices and observing the results in a 3D environment (Figure 2.6 - *a*). Another example is IDA (Interactive Drama Architecture) (Magerko 2006), which is based on an author-centric approach and uses a story director agent to maintain the progression of the stories. A similar approach is adopted in U-Director (Mott and Lester 2006), where a utility-based director agent monitors the stories according to narrative objectives, user states and story world states. In both systems the stories are represent through a 3D game engine (Figure 2.6 - *b* and Figure 2.6 - *c*). Another example of 3D system is called Mirage (El-Nasr 2007), which uses an architecture based on a set of dramatic techniques for story dramatization (Figure 2.6 - *d*).

(*a*) FearNot!

(*b*) IDA

(*c*) U-Director

(*d*) Mirage

Figure 2.6: Other graphical interactive storytelling systems. Image (*a*) shows a scene from FearNot! system; image (*b*) shows a scene from IDA system; image (*c*) shows a scene from U-Director system; and image (*d*) shows a scene from Mirage system.

The Logtell system (Pozzer 2005; Ciarlini et al. 2005), which is the base for the development of this thesis, is also an important example of interactive storytelling system that has a 3D dramatization system that represents the stories through a third person perspective. More details on Logtell system will be presented in Section 2.2.

Other examples of graphical interactive storytelling systems includes Mimesis (Young 2001), Dramachina (Donikian 2003), IDTension (Szilas 2003), Gadin (Barber and Kudenko 2007), ISRST (Nakasone and Ishizuka 2007), and PaSSAGE (Thue at al. 2007).

2.1.3. Video-Based Dramatization

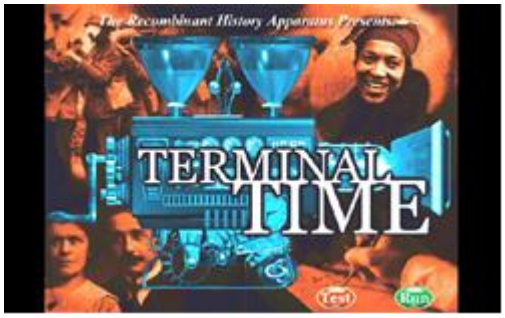
The idea of using videos as a form of visual representation of interactive narratives is not entirely new. The first attempts to use prerecorded video segments to represent some form of dynamic narrative appeared with the first experiences for interactive cinema (Činčera et al. 1967; Bejan 1992), and the academic research works on this topic date back to the 1990s (Chua and Ruan 1995; Davenport and Murtaugh 1995; Ahanger and Little 1997).

Terminal Time (Mateas et al. 2000) (Figure 2.7 - *a*) is an example of narrative system that uses videos to produce historical documentaries based on the audience's appreciation of ideological themes. It focuses on the automatic generation of narrative video sequences through a combination of knowledge-based reasoning, planning, natural language generation, and an indexed multimedia database. In this system, video clips are subsequently selected from the multimedia database according to keywords associated with the documentary events and annotated video clips.

Following a different approach, Ursu et al. (2008) explore the idea of a generic framework for the production of interactive narratives. The authors present the ShapeShifting Media, a system designed for the production and delivery of interactive screen-media narratives based on prerecorded video segments. However, the system does not incorporate any mechanism for automatic story generation. Essentially, their approach is to empower the human-centered authoring of interactive narratives rather than attempting to build systems

that generate narratives themselves. The variations of the narrative content are achieved by the automatic selection and rearrangement of atomic elements of content into individual narrations.

The applications developed with the ShapeShifting Media system include *My News & Sports My Way* (Figure 2.7 - *b*), in which the content of a continuous presentation of news is combined in accordance with users' interest, and the romantic comedy *Accidental Lovers* (Figure 2.7 - *c*), in which users can watch and influence a couple's relationship. In *Accidental Lovers*, viewers are able to influence the ongoing story by sending mobile text messages to the broadcast channel. Changes in the emotional state of the characters and their relationships depend on the existence of some specific keywords found in the viewer's messages. *Accidental Lovers* was broadcasted several times on Finnish television in late December 2006 and early January 2007 (Williams et al. 2006).



(*a*) Terminal Time

(*b*) My News & Sports My Way

(*c*) Accidental Lovers

(*d*) Last Call

Figure 2.7: Video-based interactive storytelling systems. Image (*a*) shows a scene from the Terminal Time system; image (*b*) shows a scene from My News & Sports My Way; image (*c*) shows a scene from Accidental Lovers; and image (*d*) shows a scene from Last Call.

There are also some examples of video-based interactive narratives developed for TV, Cinema and Web. Last Call (Jung von Matt 2010), for example, is an interactive advert for the 13th Street TV Channel exhibited experimentally in movie theaters. In Last Call, the audience interacts with the actress talking to her via cell phone (Figure 2.7 - *d*). Based on the audience voice commands, the system selects the sequence of videos to be presented according to a fixed tree of pre-recorded video segments.

A more complete and detailed review of the previous works on video-based interactive storytelling is presented in Section 4.1.

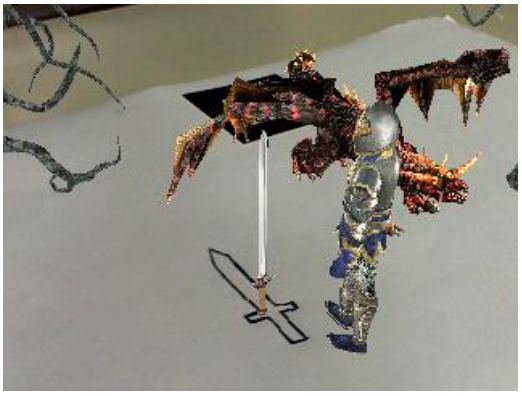
2.1.4. Other Forms of Dramatization

The current advances in virtual and augmented reality have motivated the development of other forms of story dramatization. Cavazza et al. (2007) present an interactive storytelling system where the narrative unfolds as a real-time stereoscopic 3D animation in an immersive CAVE-like system, where characters express themselves using speech synthesis as well as body animations (including elementary facial animations with lip synchronization) and the user can interact with them naturally, using speech and attitudes, as if acting on stage (Figure 2.8 - *a*). The system follows a character-based approach and the character's actions are driven by their feelings. The immersive narrative, as perceived by the user, is composed of a succession of real-time animations showing the characters moving around on stage, performing actions and expressing themselves through utterances, body attitudes and gestures. All these animations are generated by elementary actions associated to planning operators.

Lima et al. (2014) explore the use of an augmented reality visualization interface combined with a sketch-based interaction interface and presents an interactive storytelling system able to dramatize interactive narratives in augmented reality over conventional sheets of paper. Users can freely interact with the virtual characters by sketching objects on the paper, which are recognized by the system and converted into objects in the 3D story world (Figure 2.8 - *b*). In the system, stories are graphically represented in augmented reality over the paper, which creates the illusion that the sheet of paper is a virtual world

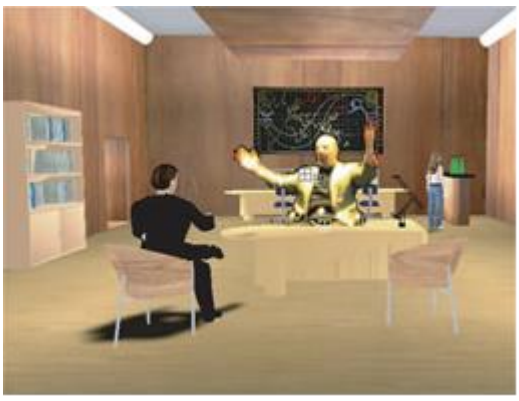
populated by virtual characters. The entire world may comprise several sheets of paper, each one representing a different location in the virtual world. Users can switch between places by changing the paper shown to the camera or by pointing the camera to other sheets of paper. They also have the freedom to move the camera and watch the scenes from different angles. Moreover, like film directors, they have the freedom to change the perspective of the stories simply by choosing to focus on a different virtual place, which generates different storylines.

(*a*) Madame Bovary on the Holodeck



(*b*) Paper and Pencil

(*c*) AR Façade



(*d*) Cavazza et al. System

Figure 2.8: Interactive storytelling systems that explore other forms of story dramatization. Image (*a*) shows a scene from Madame Bovary on the Holodeck system; image (*b*) shows a scene from Paper and Pencil interactive storytelling system; image (*c*) shows a scene from AR Façade system; and image (*d*) shows a scene from Cavazza et al. system.

Other examples of immersive systems include the interactive storytelling application presented by Cavazza et al. (2004), which uses a camera to capture the

user image to then insert him/her into a virtual world populated by virtual actors. Users are able to interact with virtual actors using body gestures and natural speech (Figure 2.8 - *d*). Other example is the augmented reality version of the desktop based interactive drama Façade (Mateas 2002) presented by Dow et al. (2006), where players can move through a physical apartment and interact with two autonomous characters using gestures and speech (Figure 2.8 – *c*).

2.2. Logtell

The present thesis is part of the Logtell Project,[3] which is a research project that aims at the development of integrated tools for managing both the generation and representation of dynamic interactive stories. The Logtell interactive storytelling system was used as basis for developing the video-based dramatization model proposed in this thesis.

Logtell is an interactive storytelling system based on temporal modal logic (Ciarlini et al. 2005) and planning under nondeterminism (Silva et al. 2010). It uses a hybrid planner that combines partial-order planning and task decomposition to efficiently deal with nondeterministic events, i.e. events that can have more than one outcome. Logtell conciliates plot-based and character-based approaches by logically modeling how goals can be brought about by previous situations and events.

In Logtell, stories are generated in chapters. In each chapter, goals to be achieved are specified either by rules or by user interventions, and the planner tries to achieve them. Situations generated by the planned events and user interventions that occur while the chapter is being dramatized influence the next chapter and so on. The chapters are represented as contingency trees, where the nodes are nondeterministic events and the edges correspond to conditions that enable the execution of the next event. A nondeterministic event e_i is executed by a nondeterministic automaton (NDA) (Doria et al. 2008) composed of actions a_i (Figure 2.9). The automaton contains information about possible sequences of actions and is open to audience's interventions.

[3] Logtell Project - http://www.icad.puc-rio.br/~logtell/

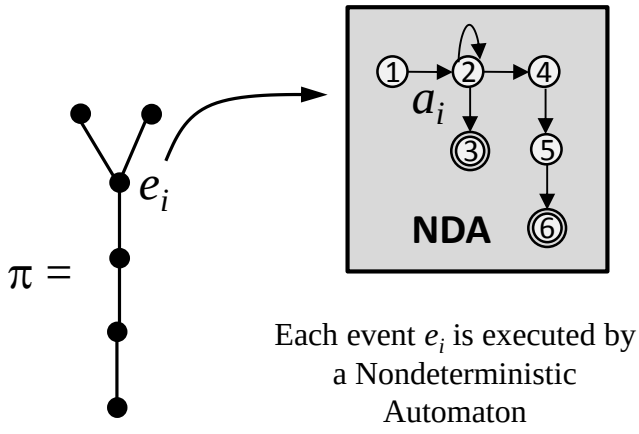


Figure 2.9: Overview of the story generation process. Plot π, events e_i, and nondeterministic automata with actions a_i. Double circles mean final states.

The Logtell system has a client/server architecture (Camanho et al. 2009) that supports multiple users sharing and interacting in the same or different stories. The client-side is responsible for user interaction and dramatization of stories. At the application server side there is a pool of servers sharing the responsibility of creating and controlling multiple stories, which are presented in different clients. The audience can interact with the story by suggesting events to the planning algorithm (to be incorporated in the next chapter) and/or interfering in the nondeterministic automata in a direct or indirect way.

The Logtell system comprises a number of distinct modules to provide support for generation, interaction and visualization of interactive plots (Figure 2.10). In the Logtell architecture, story contexts are stored in a database of contexts (Context Database), where each context contains a description of the genre according to which stories are to be generated, and also the intended initial state specifying characters and the environment at the beginning of the story. The Simulation Controller is responsible for: (1) informing the Drama Manager, at the client side, the next events to be dramatized; (2) receiving interaction requests and incorporating them in the story; (3) selecting viable and hopefully interesting suggestions for users who are intent on performing strong interactions; and (4) controlling a number of instances of the Nondeterministic Interactive Plot Generator (NDet-IPG), which is responsible for the generation of the plan to be used as input to the dramatization process. The Chapter Controller is responsible for generating the plot, chapter by chapter, including the treatment of

nondeterminism and the parallel processing of multiple valid alternatives at any given moment. The Interface Controller controls the user interventions and centralizes the suggestions made by the users. On the client side, the user interacts with the system via the User Interface, which informs the desired interactions to the Interface Controller placed at the server side. The Drama Manager requests the next event to be dramatized from the Simulation Controller, and controls actor instances for each character in a 3D environment running on the Graphical Engine.

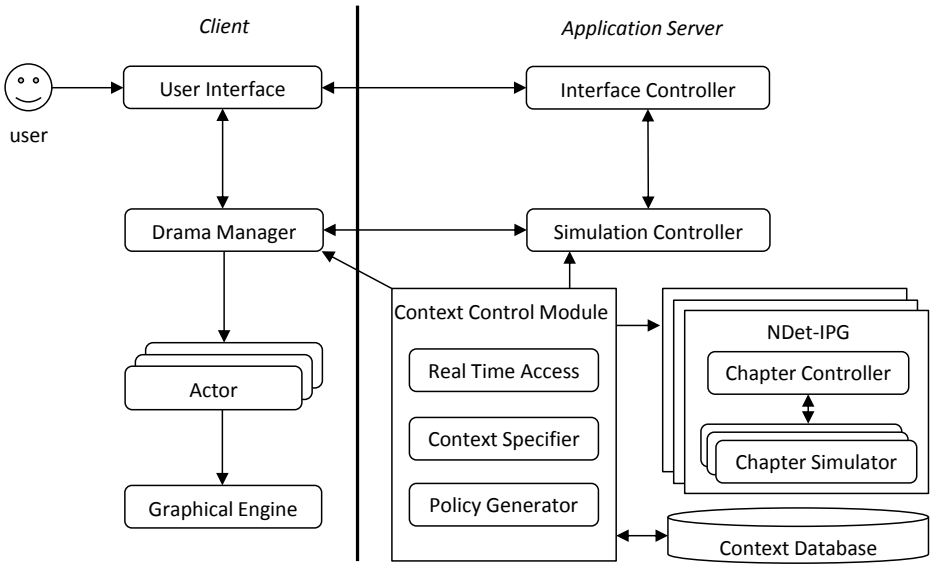


Figure 2.10: Logtell architecture. Source: adapted from (Silva 2009).

2.2.1. Story Generation

The idea behind the story generation in Logtell is to capture the logics of a genre through a temporal logic model and then verify what kind of stories can be generated by simulation combined with user intervention. In this way, the Logtell focuses not simply on different ways of telling stories but on the dynamic creation of plots. The temporal logic model is composed of typical events and goal-inference rules. Inspired by Propp's ideas on the typical functions of a narrative (Propp 1968), Logtell extensively uses planning to generate alternative stories in real-time. The logical specification defines the events that can occur and rules establishing goals to be pursued by characters when certain situations occur. Plots are generated by multiple cycles of goal-inference, planning and user intervention.

The planning process is divided into two phases. In the first phase, a partial-order planner (Ciarlini and Furtado 2002) generates a sketch of the plot by inferring new goals through the goal-inference rules or by incorporating user interventions. This sketch serves as an initial HTN for the second phase, in which a nondeterministic HTN-planner decomposes complex events into basic ones in order to obtain an executable plan. This process creates a contingency tree in which the nodes are basic events and the edges contain conditions to be tested after the event to choose the branch to be followed during dramatization.

Each basic event is modeled as a nondeterministic automaton (Doria et al. 2008), where situations observed in the world are associated to states, and actions that virtual actors can perform are associated to the transitions. In general, there is always a set of states that can be reached after the execution of an action; the selection of which transition must occur could be a user choice or randomly chosen according to weights associated to the transitions. Figure 2.11 shows an example of automaton created to represent the possibilities for the dramatization of an event where a villain kidnaps a victim.

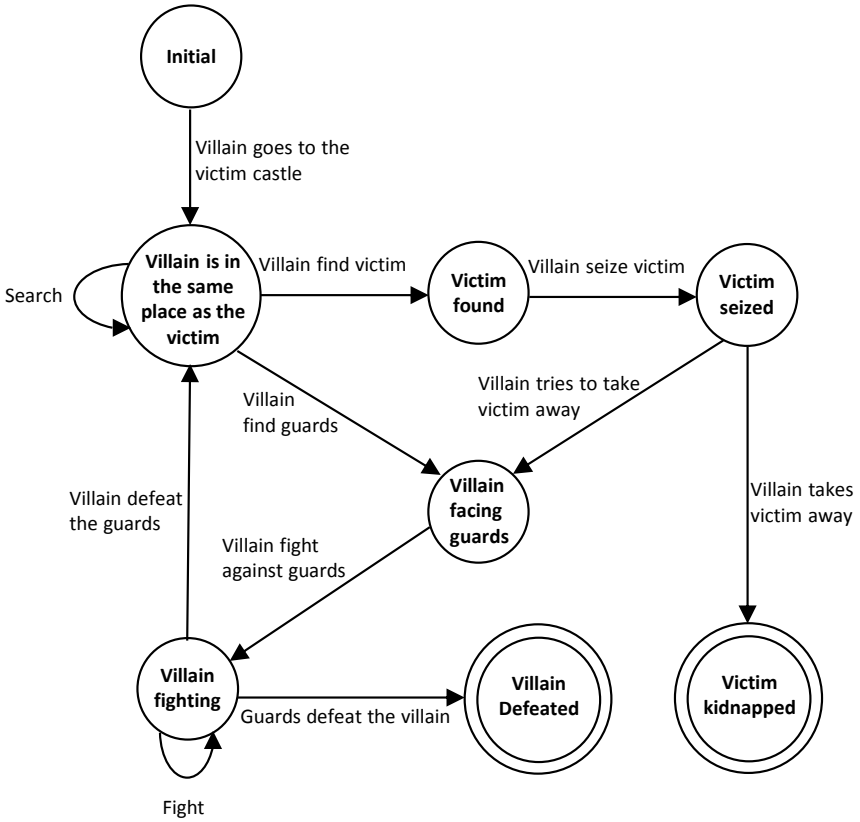


Figure 2.11: Example of automaton representing the possibilities for the dramatization of a kidnap event. Source: adapted from (Doria et al. 2008).

During the dramatization of the nondeterministic automata, the system is also able to automatically select policies that can either accelerate or extend the presentation time. The generated policies can be weak, strong with cycles or strong. In weak policies, there is at least one path from the initial state to a goal state, but states from which it is not possible to reach the goal according to the policy can be reached. When a policy is strong with cycles, the goal state is always reachable but cycles can occur, so that the time to reach the goal might be virtually infinite. Strong policies guarantee that, from any state in the policy, the goal state is reached at some moment. The use of policies allows the system to coordinate plot generation and dramatization in parallel.

More details about the story generation model of the Logtell system can be found elsewhere (Ciarlini and Furtado 2002; Ciarlini et al. 2005; Doria et al. 2008; Silva et al. 2010).

2.2.2. User Interaction

The Logtell system offers two forms of user interaction: (1) weak interventions, where users can select alternatives that are automatically generated by the planning algorithms; and (2) strong interventions, where users can try to force the occurrence of events or specific situations in the story. Figure 2.12 shows the user interface of the Logtell system. The interaction window occupies a small part of the screen and appears in parallel with the main window, where the story is dramatized.

Weak interventions occur by means of the commands “rewind” and “another” (Camanho et al. 2009). The rewind command allows users to “return” to the start of the selected chapter. The chapter is presented again and the user has the opportunity to interact with the system again and select alternatives for the next chapters. When the command is executed, the system retrieves a logical snapshot of the selected chapter and resumes the simulation from this point, discarding the snapshots of the next states, which will be generated again in accordance with the user’s interactions. The “another” command is used to ask the system to provide an alternative for the selected chapter. In response to the command, the planning system generates another solution for the goals that were

reached. In this way, a different combination of events can be generated for the chapter, wherefrom a completely different continuation of the story can be developed.

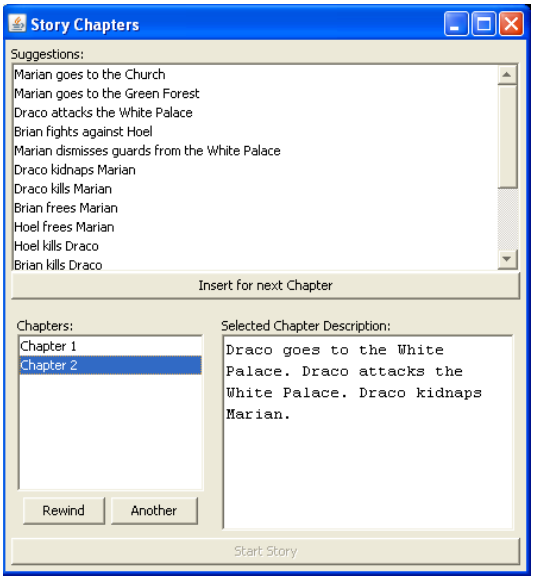


Figure 2.12: User interface for continuous interaction. Source: (Camanho et al. 2009).

Strong interventions enable users to indicate events and situations that should occur in the next chapters. The suggested situations are considered as goals to be achieved at a certain time, and events can have unfulfilled pre-conditions that might demand the insertion of more events. In such cases, the system has to plan a chapter with additional events and constraints that make the user intervention consistent with the plot and the rules of the genre. If this is not possible, the user intervention is simply rejected. The system also includes a mechanism in which viable strong interventions are automatically suggested to users, so that they can simply select the one that better suits their tastes. The list of suggestions is updated whenever the presentation of a new chapter starts.

More details about the user interaction mechanisms of the Logtell system can be found elsewhere (Pozzer 2005; Ciarlini et al. 2005; Camanho et al. 2009).

2.2.3. Dramatization

Logtell represents the stories generated by the planning system in a 3D environment (Figure 2.13), where characters are represented through 3D models

and their actions through animations (Pozzer 2005; Lima 2009). The virtual world is represented by a hand-built 3D environment that is consistent with the logical definition of the story context, which means that it contains all the locations where the story events can happen. Similarly, all characters described in the story context are associated with a 3D actor in the dramatization system.

Figure 2.13: Scenes from the 3D dramatization module of the Logtell system.

The system provides a set of parameterized actions that can be used to visually represent the generated stories. These actions correspond to the basic actions described in the nondeterministic automata and are represented by the virtual 3D actors trough animations controlled by the dramatization system. The behavior of the actors is determined by the sequence of actions that must be dramatized in order to represent the story events.

The dramatization system has the goal of emphasizing the dramatic content of the scenes and presenting them in a more attractive and engaging way to the viewers. Its architecture is composed of a set of cinematography-inspired autonomous agents that controls the dramatization, actors, cameras, lights and music. The dramatization system is capable of automatically placing the cameras in the virtual environment, selecting the best camera angle to show the action, and selecting the best visual effects and sound tracks to the scenes of the narrative.

More details about the dramatization process of the Logtell system can be found elsewhere (Pozzer 2005; Ciarlini et al. 2005; Lima 2009; Lima et al. 2009; Lima et al. 2010).

2.3. Conclusion

This chapter introduced the main concepts of interactive storytelling and presented a brief review of the main interactive storytelling systems, emphasizing the methods used by these systems to visually represent interactive stories. Table 2.1 summarizes the results of this study by showing a list of the main interactive storytelling systems and their respective story generation models and dramatization methods. We can observe a variety of dramatization methods that vary from simple textual stories to complex virtual environments. While early interactive storytelling systems were mainly based on textual descriptions of stories, more recent system have been exploring new forms of dramatization, such as immersive mixed reality environments and video-based representations.

System	**Story Generation Model**	**Dramatization Method**
Kinoautomat (1967)	Branching Points	Video
Tale-Spin (1977)	Planning	Text
Universe (1984)	Planning	Text
The Edge of Intention (1991)	Agent-Based Planning	2D
Minstrel (1992)	Planning	Text
I'm Your Man (1992)	Branching Points	Video
Façade (2002)	Agent-Based Planning	3D
Cavazza et al. (2002)	HTN Planning	3D
Pizzi and Cavazza (2007)	Emotional Planning	3D
FearNot! (2005)	Agent-Based Planning	3D
Logtell (2005)	Nondeterministic Planning	3D
Terminal Time (2000)	Planning	Video
Accidental Lovers (2007)	Branching Points	Video
Last Call (2010)	Branching Points	Video
Cavazza et al. (2007)	Emotional Planning	Immersive System
Paper and Pencil (2011)	Agent-Based Planning	Augmented Reality

Table 2.1: List of the main interactive storytelling systems and their respective story generation models and dramatization methods.

Based on this study, we can observe that the most robust interactive storytelling systems adopt a story generation model based on planning, while most of the previous works on video-based interactive storytelling are still based on branching narrative structures. In addition, previous works are entirely based on immutable pre-recorded segments of video, which reduces interactivity, restricts story diversity, and increases the productions costs.

3
Cinematography

The cinematography theory provides the basic principles and background for the creation of attractive and engaging visual representation for interactive stories. This chapter reviews some essential concepts of cinematography that are important for the development of the proposed video-based interactive storytelling system.

Cinematography can be defined as the "art of film making" (Brown 2011). The term was created in the film industry to describe the process of creating images on film, and it covers all aspects of camera work, including the creative process of making aesthetically pleasing images and the technical aspects involved with using cameras, lights, and other equipment (Newman 2008). More specifically, the cinematography theory describes a set of principles and rules to effectively use cameras, actors, illumination and soundtracks to visually tell a story. Cinematography involves taking the narrative ideas, actions, emotional meanings and all other forms of non-verbal communication and rendering them in visual terms. It provides ways to add dramatic emphasis where required, to communicate additional information, and ultimately to evoke emotional responses in the audience. The successful application of cinematography concepts results in a coherent and attractive visual narrative (Kneafsey 2006).

The first step to comprehend the basic concepts of cinematography is analyzing the structure of a film (Figure 3.1). A film consists of a linear sequence of scenes, where each scene is composed of several shots. A scene defines the place or setting where the action happens. A shot consists of a continuous view of the scene filmed by one camera without interruption (Mascelli 1965). Each shot consists of a linear sequence of image frames that compose the complete moving picture. The transition between one shot to another is known as cut.

The smallest element of interest in a film is a single frame, and for some filmmakers, each frame represents a masterpiece that is carefully planned and composed (Brown 2011). The combination of several frames filmed without

interruptions constitutes a shot. The type of camera angle used in a shot defines how much the camera, and therefore the viewer, engages with the event depicted in the shot (Kneafsey 2006). The camera may be stationary or it may employ simple or complex movements depending on the content of the scene and the mood that is to be established. The combination of a number of shots creates a scene, which is a single setting where a set of events take place during a particular time period. Independently of type of shot, camera movement or transition between shots or scenes, it is always important to keep the view's mind oriented in time and space (Mascelli 1965). The cinematography theory provides basic guidelines on how to position and move the cameras and actors, on how to perform cuts and transitions between shots and scenes, and especially, on how to keep the spatial and temporal continuity of the film.

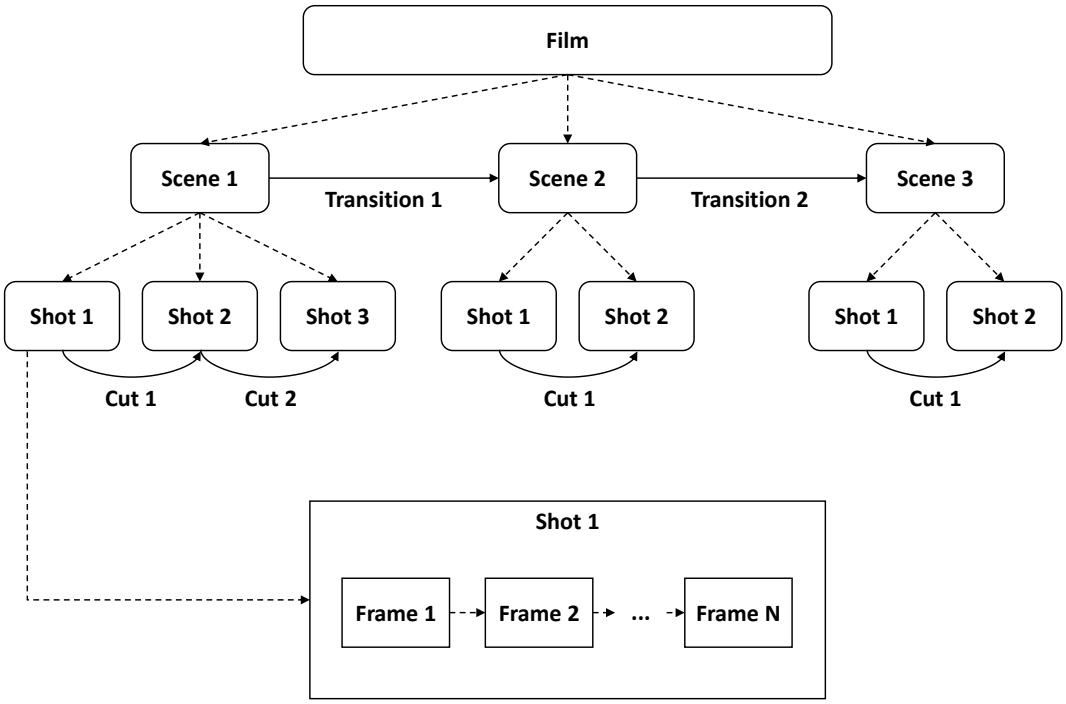


Figure 3.1: The structure of a film.

3.1. Shot

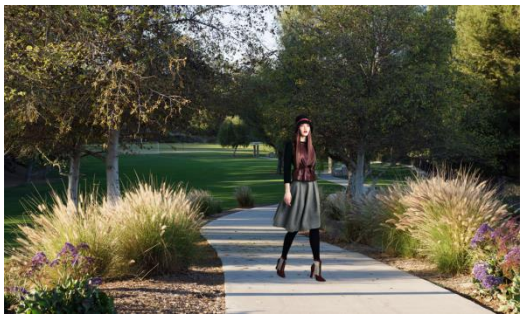
Bowen and Thompson (2009) define a shot as the smallest unit of visual information captured at one time by the camera that shows a certain action or event from a specific point of view. A shot creates a continuous visualization of a situation and it is characterized by position, orientation, and movement of the

camera and of the participating characters and objects of the scene (Hornung 2003). Each shot requires placing the camera in the best position for viewing characters, setting and action at that particular moment in the narrative (Mascelli 1965). According to Brown (2011), camera placement is a key decision in storytelling. More than just "where it looks good," it determines what the audience sees and from what perspective they see it.

A shot can be described by three main characteristics: (1) camera angle, which determines the viewer's point of view (objective, subjective or point-of-view); (2) shot type, which defines how much of the scene and the subject will be visible in the shot; and (3) camera height, which defines the height of the camera and, consequently, influences the viewer's psychological relationship with the scene characters.

The camera angle has a strong effect on the dramatic impact of the story. A more subjective camera angle places the viewer into the scene, while a more objective angle provides a general view of the scene (Mascelli 1965). The type of shot determines the size of the subject in relation to the overall frame and also has dramatic effects on the story (Mascelli 1965; Kneafsey 2006). Scenes are often opened with establishing shots (extreme long shots, very long shots or long shots), which helps viewers to understand the situation that will be presented (Brown 2011). When more detailed information is required, median shots are used to bring the audience closer to the action, and close-ups are used to add dramatic emphasis to the facial expressions of characters. Figure 3.2 illustrates the most common types of shot.

The height of the camera can also be manipulated to add dramatic and psychological overtones to the narrative. According to Mascelli (1965), eye-level camera angles are best for shooting general scenes that should be presented from a normal eye-level. High and low camera angles are usually chosen for esthetic, technical or psychological reasons. High camera angles put the audience into an elevated and powerful position, making the subject seems smaller and weaker. Low camera angles create the opposite impression, showing the might and power of the subject.

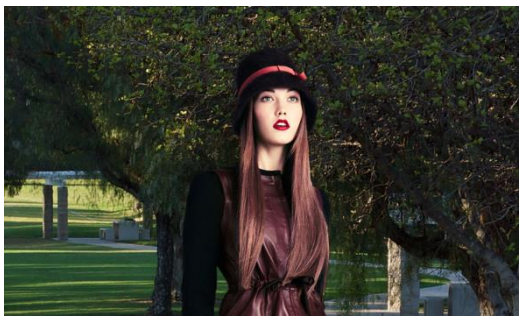


(*a*) *Very Long Shot*

(*b*) *Long Shot*

(*c*) *Medium Long Shot*

(*d*) *Medium Shot*



(*e*) *Medium Close-up*

(*f*) *Close-up*

Figure 3.2: Shot types.

3.2. Camera Movements

Motion is the primary aspect that differentiates film from photography and painting (Hawkins 2005). All characteristics of the camera movement (style, trajectory, pacing and timing in relation to the action), contribute to the mood and feel of the narrative. According to Brown (2011), camera movements can enhance the scene and add a layer of meaning beyond the shots themselves. They can add an additional emotional content, a sense of energy, joy, menace, sadness, or any other emotional overlay. There are several basic camera movements, and combinations of these movements can create more complex shots.

Brown (2011), Katz (1991) and Hawkins (2005) classify the camera movements in four fundamental types: (1) pan, which consist in the movement of rotating the camera on a vertical axis; (2) tilt, which is the movement of rotating the camera on a horizontal axis perpendicular to direction in which the camera is pointing; (3) dolly, which consist in the movement of the camera along the horizontal plane; and (4) crane, which is the movement of the camera along the vertical and horizontal planes.

3.3. Continuity

A film can create its own time and space to fit any particular storytelling situation. Time may be compressed or expanded; speeded or slowed; remain in the present, go forward or backward. Space may be shortened or stretched; moved nearer or farther; presented in true or false perspective; or be completely remade in to a setting that may exist only on film (Mascelli 1965). Independently of all spatial and temporal awkwardness a filmmaker may purposely create in a film, it is always important to keep the viewer's mind oriented in time and space.

Continuity, with respect to motion pictures, refers to the logical consistency of the story, dialogs, position and movements of cameras, characters and objects present in the narrative. According to Brown (2011), when the audience becomes aware of continuity errors, they simultaneously become conscious that they are watching a movie, which breaks the storytelling illusion. Over the years, film directors and cinematographers have developed several rules and principles to maintain the basic spatial and temporal continuity of a film. Some of these rules can be applied when the film is being shot and others during the editing process. Mascelli (1965), Thompson and Bowen (2009) and Brown (2011) describe the following main rules to maintain the film continuity during the shooting process:

1. **Line of Action:** for each scene, a line of action must be established. The line of action or action axis consists of an imaginary line connecting the most important elements or directing the focus of the action in a scene. The audience unconsciously observe these lines and cinematographers use this phenomenon to help establishing narrative meaning and shot

composition, and to reinforce spatial relationships within the film space. The line of action is also used in the 180 degree rule to preserve a consistent screen direction and spatial continuity.

2. **180 Degree Rule:** the rule determines that when shooting a scene, the camera must be placed only at one side of the line of action. The placement of the camera in different angles for new shots of the scene must occur only within the 180 degree arc. The 180 degree rule helps to maintain the lines of attention, the screen direction and the visual continuity of consecutive shots, which prevents the audience from getting confused about where someone is in the scene.
3. **30 Degree Rule:** the rule determines that when shooting two consecutive shots of the same subject from inside of the 180 degree arc, the camera angle for the new shot must be at least 30 degree from the angle of the previous shot. In this way, the two shots can be considered different enough to avoid jump cuts, which is an undesirable effect that causes visual jumps in either space or time of the film. The same rule applies when using zoom to produce the new shot – it must have at least 20% of difference relative to the previous shot.
4. **Reciprocating Imagery:** the rule determines that when shooting two separate subjects with single shots in the same scene, some characteristics of the camera must match in both shots. Changes in the camera height, type of the lens, focal length and illumination may compromise the film continuity.

A well-defined line of action for each scene is a key factor to maintain the spatial continuity of the whole film. According to Kneafsey (2006), for moving subject the line of action is usually defined by the motion path of the subject at a given instant. In this way, the subject will always be moving towards the same side of the screen. For stationary subjects, the line of action is often drawn in the direction the subject is facing. For two or more subjects, the line of action is defined by a line connecting the two most important characters of the scene. The continuity will be maintained as long as the camera remains within the 180º horizontal arc of one side of the line of action during a cut (Brown 2011). However, this rule only applies when one shot ends and the next one begins.

During the same shot, the camera can freely move and cross the line without breaking the film continuity because the audience can observe the movement and the spatial relation of the subjects.

3.4. Filming Methods

The sequence of shots that produce a scene is one of the most important elements in the process of creating visually aesthetical images and conveying consistent visual interpretations that do not contradict the intended narrative meaning (Hornung 2003). These shots can be filmed in several different ways depending not only on the style of the cinematographer but also on whether or not the action is controllable and can be repeated several times for multiple takes (Kneafsey 2006). The cinematography theory describes several ways to film a scene. The three most common filming methods are the master scene, the triple take (overlapping) and the plan-scene (in one).

According to Mascelli (1965) and Brown (2011), the master scene is the most common filming method used in narrative films. The method involves filming the entire scene with a master shot (a shot that includes the whole film set and uses a view angle that is different from other cameras) along with coverage shots (shots that reveal different aspects of the action). In this way, the editor has always two shots of each scene. If a continuity problem is detected in the coverage shots, a new shot of the same action can be extracted from the master scene. Furthermore, this method gives to the editor the freedom to creatively cut and alter the pacing, the emphasis, and even the point of view of the scenes. Filming a scene with the master scene method can be done using a single camera or multiple cameras. If filmed with a single camera, the action is repeated several times to obtain the coverage shots and the master shot. If filmed with multiple cameras, the coverage shots and the master shot are filmed simultaneously.

3.5. Editing

Films, especially narrative feature films, are made up of a series of individual shots that filmmakers connect in a formal, systematic, and expressive

way (Sikov 2009). Editing a film involves more than just assembling the shots one after the other. It involves the creative process of organizing, reviewing, selecting, and assembling the various picture and sound elements captured during production process so that it creates a coherent and meaningful visual presentation that comes as close as possible to achieving the goals behind the original intent of the work (Thompson and Bowen 2009). According to Mascelli (1965), only a good editing can bring life to a motion picture.

The most common editing method is called continuity editing (Mascelli 1965). It is described by a set of editing practices that establish spatial and temporal continuity between shots and keep the narrative moving forward logically and smoothly, without disruptions in space or time (Brown 2011). Continuity editing is used to join shots together to create dramatic meanings. With an effective editor, the audience will not notice how shots of various frame sizes and angles are spliced together to tell the story. The best editing is usually the unobtrusive editing (Mascelli 1965), that is, the one in which the audience does not notice that the editor joined the shots.

Each cut must always be unobtrusive and sustain the audience's attention on the narrative (Mascelli 1965). One way of complying with this editing principle is by avoiding jump cuts. A jump cut is often regarded as a mistake in classical editing (Butler 2002). It usually occurs when two very similar shots of the same subject are joined together by a cut, producing the impression that the subject "jumps" into a new pose. A jump cut produces a disorientation effect, confusing the spectators spatially and temporally. The best way to avoid jump cuts is respecting the 30 degree rule in consecutive shots during the shooting process. However, a good editor must always check the final sequence of shots to ensure that no jump cuts occurs.

Another important cinematography principle used by conventional editors to join and maintain the continuity between segments of videos is the use of adequate scene transitions. There are four basic ways to transit from one shot to another (Thompson and Bowen 2009):

1. **Cut:** Consists of an instantaneous change from one shot to the next. It is most often used where the action is continuous and when there is no change in time or location.

2. **Dissolve:** Consists of a gradual change from the ending of one shot into the beginning of the next shot. The dissolve is correctly used when there is a change in time or location, the time needs to be slowed down or speeded up, and when there is a visual relationship between the outgoing and the incoming images.
3. **Wipe:** Consists of a line or shape that moves across the screen removing the image of the shot just ending while simultaneously revealing the next shot behind the line or shape. The wipe is correctly used where there is a change in the location and when there is no strong visual relationship between the outgoing and the incoming frames.
4. **Fade:** Consists of a gradual change from a fully visible image into a solid black screen (fade-out) and a gradual change from a solid black screen into a fully visible image (fade-in). The fade is used at the beginning/end of a film, scene, or sequence.

Each one of these four scene transitions carries with it its own meanings (Thompson and Bowen 2009). The cut is the most frequently used transition, and, when it is made at the correct moment, it is not consciously noticed by the audience. The dissolve is the second most common scene transition, and unlike the straight cut, it attracts the audience's attention on purpose. Dissolves are more often used to indicate the passage of time (few seconds or many years). Other common uses of dissolves include: flashbacks, flashforwards, parallel actions and dreams (Barbash and Taylor 1997). Fade transitions are similar to dissolves, but they tend to be more emphatic, and usually express a more substantial rupture of time, space, theme, or plot. Wipes were very common in early films to indicate change in place, but nowadays they are rarely used. Some directors, however, are known for their extensive use of wipe transitions, like George Lucas in his Star Wars films (Caldwell 2011).

3.6. Matting and Compositing

Compositing is the process of assembling multiple visual elements from different sources into a single piece of motion picture (Lanier 2009). The goal of a

digital composite is to create the illusion that all the visual elements always existed in the same location or scene. The matting process consists in the extraction of the visual elements from the background so they can be used in the compositing process. According to Sawicki (2011), chroma key (also referred as green screen or blue screen) is the most common matting technique used in the film industry today. The chroma key involves shooting the visual elements in front of a green or blue screen, then using an algorithm to remove the colored screen from the shot and replace it with the substitute background during the compositing process (Aronson 2006; Foster 2010).

Early film productions traditionally used blue screens for the background. The main reason for this choice was because blue is complementary to the skin tone, and its wavelength can be isolated while still getting a fairly acceptable color rendition for faces (Sawicki 2011). However, green is currently the most popular background color. According to Foster (2010), this occurs because the image sensors of modern digital video cameras are most sensitive to green, due to the Bayer pattern, which allocates more pixels to the green channel. In addition, the green color requires less light to illuminate the background, because of its higher luminance and sensitivity in the image sensors. Green and blue are the most common colors used for backgrounds, but in theory any color can be used. Red is usually avoided due to its prevalence in normal human skin pigments, but can be used for other objects.

Figure 3.3 shows an example of compositing process using a green screen background. Initially, the wolf is captured in front a green screen and then composed with the actor and the background. The compositing result creates the illusion that both wolf and actor were in the same place.

Matting and compositing techniques can also be used to add computer generated images and 3D objects to the scenes, merging virtual and real words (Wright 2010). These objects can constitute the whole environments or other components such as character, furniture or other elements. Another example of element that can be added to the scenes are matte paintings, which are a painted representation of a landscape that allows filmmakers to create the illusion of fantasy environments or to represent scenarios that would be too expensive or impossible to build or visit (Okun and Zwerman 2010). Figure 3.4 shows an

example of scene where the green screen background is replaced by a matte painting of mountains and sky.

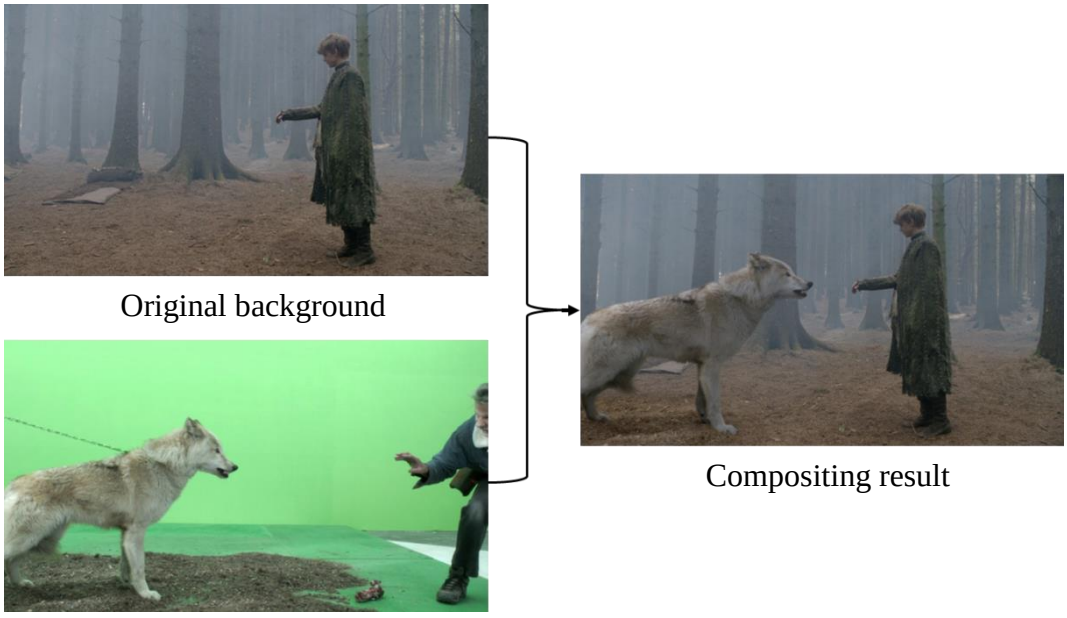


Figure 3.3: The compositing process using the chroma key technique. Source: (ArtOfVFX 2013). Copyrighted images reproduced under “fair use” policy.

(*a*) Original scene (*b*) Compositing result

Figure 3.4: Example of scene created using the chroma key and matte painting techniques. Source: (ArtOfVFX 2013). Copyrighted images reproduced under “fair use” policy.

Matting and compositing techniques are crucial operations in visual effects production, allowing filmmakers to create a world of fantasy by combining live action and visual effects (Foster 2010). The term visual effect is used to describe

any imagery created, altered, or enhanced for a film that cannot be accomplished during live-action shooting. In other words, it refers to the process of adding and modifying the visual content of the film during the post-production phase. According to Okun and Zwerman (2010), there are three main reasons for using compositing techniques and visual effects: (1) when there are no practical ways to film the scenes required by the script or the director; (2) for safety reasons, when the scene could be done practically, but may cause personal injuries; and (3) for cost efficiency, when it is more economical or practical to use visual effects than filming the real scene.

Compositing techniques are also useful to reduce the number of actors required to represent complex scenes such as war sequences, which are known in the film industry for requiring huge investments in terms of money and manpower. By using compositing techniques, it is possible to multiply the numbers of actors, animals or other objects during the actual shot into as many as necessary to fill the screen. Figure 3.5 shows an example of war sequence, where the few actors present in the original scene were multiplied during the post-production phase in order to create an army.

(*a*) Original scene (*b*) Compositing result

Figure 3.5: Example of scene created using the chroma key techniques. Source: (ArtOfVFX 2013). Copyrighted images reproduced under “fair use” policy.

3.7. Light and Color

Light is one of the most important elements to create the mood and the atmosphere of a film (O'Brien and Sibley 1995). Scenes with a lot of darkness and shadows increase the impact of emotions such fear and foreboding. Horror or

thriller films use low-key lighting to increase the sense of fear in the audience (Sullivan et al. 2008). A high-key lighting is more calming, it can evoke beauty, innocence, tranquility, and romance. Scenes with bright light increase the sense of well-being.

Color is an important aspect provided by light. It can affect the psychological perception the audience has of the scenes, change its mood and transmit emotions. Color is a powerful storytelling tool and an important factor to express emotions through images (Brown 2011). According to LoBrutto (2002), colors can be very subjective, but particular hues and palettes do represent, indicate, and communicate narrative messages to the audience. Warm colors tend to represent tenderness and humanity. Cool colors represent cold, lack of emotion, and distant feelings. Hot colors represent sexuality, anger, and passion. A monochromatic palette is a limited range of colors that can establish a colorless world, sameness, masked emotion, or a sense of simplicity.

3.8. Music

Music also is a powerful tool to express emotions. In a film, music can change the feel of a scene, bring out the emotions and enhance the reaction of the audience. According to Davis (2010), music is a fundamental element of a film. It creates the connection between the emotional content of the narrative and the visual events presented on the screen. Music is a communication tool, and it represents and communicates the narrative in a non-verbal way, filling the narrative gaps by being able to say more than the visual image can, particularly in terms of emotions (Ferreira 2012).

Several characteristics have been suggested that might influence the emotion of music. According to Gabrielsson and Lindstrom (2001), major keys and rapid tempos cause happiness, whereas minor keys and slow tempos cause sadness, and rapid tempos together with dissonance cause fear. The choice of instrumentation, whether soothing or obnoxious, will have an effect. Music can set the stage and place spectators in a different world, a different country, or a different time. Music is primarily designed to create a certain atmosphere or feeling for the scenes. It can create a dark and mysterious world, adding tension

and desperation to reinforce the seriousness of a situation. Music can express emotions and feelings so successfully because it works beneath our conscious level. It can cue us as to how to respond to the film or to a particular scene of the film without taking up additional screen time or space (Miller 1997). As music can enhance a scene, it can also ruin a scene. Incorrect type of music during a particular scene can nullify the emotions expressed by the actors.

3.9. Film Crew

Producing a film involves several professionals, and each member of the crew has specific roles and tasks in different phases of the production process. LoBrutto (2002), Kodak (2007) and Zettl (2012) describe the principal professionals and their main roles in the creation of a film:

- **Screenwriter (or Scriptwriter):** is responsible for creating the original story, or adapting a book, or other form of narrative for use as a script for the film. He/she must create a compelling and coherent story, and decide how to structure the narrative for presentation as a film;
- **Director:** is responsible for translating the script into a visual presentation. He/she controls the overall aspects of the film, including the content and flow of the narrative events, the performance of the actors, the organization and selection of the locations in which the film will be shot, and the management of technical details such as the position of cameras, the use of lights, and the content of soundtracks. The director is responsible for the artistic and dramatic aspects of the film. He/she must visualize the whole script and guide the technical crew and actors to fulfill his/her vision of the narrative;
- **Director of Photography:** is responsible for the visual quality and the cinematic look of the film. The director of photography transforms the screenwriter's and director's concepts into visual images. Using his/her knowledge of lighting, lenses, cameras, colors, and emotions, creates the

appropriate mood, atmosphere, and visual style of each shot to evoke the emotions that the scene must express;

- **Camera Operator (or Cameraman):** is responsible for the general operation of the camera. The cameraman works very closely with the director. He/she helps the director to translate his/her vision of the narrative onto film by suggesting possible camera placements, lens, movements, lighting and composition;
- **Editor:** during the post-production phase, the editor is responsible for selecting shots from the raw footage, and combining them into sequences to create a finished motion picture. The editing process is often called "the invisible art", because when it is well-done, the audience become so engaged in the narrative that they do not consciously notice the editor's work;
- **Compositor (Visual Effects):** works with the visual effects team and is responsible for compositing live action videos, computer-generated elements, and other resources from different sources to create the final image of the film. He/she is responsible for the aesthetic integrity and technical quality of the composed scenes.
- **Music Director:** during the post-production phase, the music director is responsible for combining music with the visual media of the film. Using his/her knowledge of music, the music director creates the mood and atmosphere of each scene based on the emotions and feelings that the scenes must express. He/she must have a wide knowledge of music and must know the effects that music has on the audience.

3.10. Conclusion

This chapter presented a brief overview of some important concepts of cinematography that are essential for the development of a video-based interactive storytelling system. The next chapter will present the proposed video-based dramatization model and will discuss how cinematography theory must be applied to maintain the film continuity and preset the story events in an attractive and engaging manner.

4
Video-Based Interactive Storytelling

This thesis proposes a new approach to video-based interactive narratives that uses real-time video compositing techniques to dynamically create video sequences representing the story events generated by planning algorithms. The proposed approach consists of filming the actors representing the characters of the story in front of a green screen, which allows the system to remove the green background using the chroma key matting technique and dynamically compose the scenes of the narrative without being restricted by static video sequences. In addition, both actors and locations are filmed from different angles in order to provide the system with the freedom to dramatize scenes applying the basic cinematography concepts during the dramatization of the narrative. A total of 8 angles of the actors performing their actions are shot using a single or multiple cameras in front of a green screen with intervals of 45 degrees (forming a circle around the subject). Similarly, each location of the narrative is also shot from 8 angles with intervals of 45 degrees (forming a circle around the stage). In this way, the system can compose scenes from different angles, simulate camera movements and create more dynamic video sequences that cover all the important aspects of the cinematography theory.

The proposed video-based interactive storytelling model combines robust story generation algorithms, flexible multi-user interaction interfaces and cinematic story dramatizations using videos. It is based on the logical framework for story generation of the Logtell system, with the addition of new multi-user interaction techniques and algorithms for video-based story dramatization using cinematography principles.

This chapter discusses the related works and describes the main differences between the proposed system and previous work. It also presents an overview of the architecture of the video-based interactive storytelling system from a software engineering perspective.

4.1. Related Work

The idea of using videos as a form of visual representation of interactive narratives is not entirely new. The first attempts to use prerecorded video segments to represent some form of dynamic narrative date back to the 1960s (Činčera et al. 1967; Bejan 1992; Chua and Ruan 1995; Davenport and Murtaugh 1995; Ahanger and Little 1997) and several other interactive narrative experiences using videos have been developed through the years. The game industry was the first to explore the use of videos as a form of interactive content. During the early 1980s a new class of games, known as full motion video (FMV) based games or simply by interactive movies, emerged and became very popular. The main characteristic of these games is that their content was mainly based on pre-recorded video segments rather than sprites, vectors, or 3D models.

The first game to explore the use of full motion videos as the game content was Dragon's Lair (1983). Although the genre came to be associated with live-action video, its first occurrence is an animated interactive movie. In Dragon's Lair, the player has the role of a sword fighting hero who needs to win many fights and gather items to finally free a princess from a dragon. The gameplay consists of making decisions by using a joystick to give directions to the virtual character. If the player chooses the right action and its respective button is pressed at the right moment, the obstacle is overcome. If not, the character dies and the player loses a life. Space Ace (1984), another game from the same production team of Dragon's Lair, used a similar idea, but improved on its predecessor with an expanded storyline, with multiple branch points and selectable skill levels.

FMV-based games were considered the cutting edge technology at the beginning of the 1990s and were seen as the future of the game industry. However, as the consoles of that time evolved, the popularity of these games decreased drastically. Today, they are known as one of great failures of the game industry (Wolf 2007). The main problem was the lack of interactivity. The gameplay of most part of them was based on pressing a sequence of buttons in pre-determined moments to keep the narrative moving forward. The narratives also had a very limited branching factor, because every action, every movement, every success and every failure had to be either pre-filmed or pre-rendered.

Obviously it was expensive in terms of production, so the designers had to reduce the interaction options to reduce costs. At that time, FMV-based games failed in the attempt of creating a link between games and films.

In the same time, academic researchers begin to explore the capabilities of videos as an interactive content. Davenport and Murtaugh (1995) present a method to maintain temporal continuity between segments of videos by scoring metadata associated with all available scenes. In their application, users are able to navigate through a collection of documentary scenes describing theme, time and location. Terminal Time (Mateas et al. 2000) is another example of narrative system that uses videos to produce historical documentaries based on the audience's appreciation of ideological themes. It focuses on the automatic generation of narrative video sequences through a combination of knowledge-based reasoning, planning, natural language generation, and an indexed multimedia database. In their system, video clips are subsequently selected from the multimedia database according to keywords associated with the documentary events and annotated video clips. In a similar approach, Bocconi (2006) presents a system that generates video documentaries based on verbal annotations added to the audio channel of the video segments. A model of verbal relations is used to automatically generate video sequences for user-specified arguments. In another work, Chua and Ruan (2005) designed a system to support the process of video information management: segmenting, logging, retrieving, and sequencing. Their system semi-automatically detects and annotates shots for later retrieval. The retrieving system uses rules to retrieve shots for presentation within a specified time constraint.

Ahanger and Little (1997) present an automated system to compose and deliver segments of news videos. In the system, content-based metadata and structure-based metadata are used to compose news items. The composition process is based on knowledge about the structure of a news item (introduction, body, and end) and how various types of segments fit into the structure. Within restrictions imposed by the composition grammar, segments belonging to the body can be presented in any order if their creation times are within a small range. Related segments can be included or excluded to meet preference to time constraints without sacrificing continuity. The authors also present a set of metrics to evaluate the quality of news videos created by the automated editing process.

These metrics include thematic continuity, temporal continuity, structural continuity, period span coverage, and content progression (Ahanger and Little 1998). In a similar approach, but not focusing on news videos, Nack and Parkes (1997) present a method to establish continuity between segments of videos using rules based on the content of the segments. Their application is capable of automatically generating humorous video sequences from arbitrary video material. The content of the segments is described with information about the characters, actions, moods, locations, and position of the camera.

Hypervideo, or hyperlinked video, is another form of media that explores interactivity by including embedded user-clickable anchors into videos, allowing the user to navigate between video and other hypermedia elements. HyperCafe (Sawhney et al. 1996) is one of the first hypervideo examples that were primarily designed as a cinematic experience of hyper-linked video scenes. Currently, hypervideo research is mainly focused on the efficient definition of interactive regions in videos. VideoClix (2014) and ADIVI (2014) are examples of authoring tools for defining flexible hyperlinks and actions in a video. However, they do not directly support the generation of interactive narratives.

Another research problem closely related to automatic video editing is video summarization, which refers to the process of creating a summary of a digital video. This summary, which must contain only high priority entities and events from the video, should exhibit reasonable degrees of continuity and should be free of repetition. A classical approach to video summarization is presented by Ma et al. (2002). The authors present a method to measure the viewer's attention without fully semantic understanding of the video content. As result, the system could select the high priority video events based on the evoked attention.

The idea of a generic framework for the production of interactive narratives is explored by Ursu et al. (2008). The authors present the ShapeShifting Media, a system designed for the production and delivery of interactive screen-media narratives. The productions are mainly made with prerecorded video segments. The variations are achieved by the automatic selection and rearrangement of atomic elements of content into individual narrations. The system does not incorporate any mechanism for the automatic generation of stories. Essentially, their approach is to empower the human-centered authoring of interactive narratives rather than attempting to build systems that generate narratives by

themselves. The applications developed with the ShapeShifting Media system include My News & Sports My Way, in which the content of a continuous presentation of news is combined in accordance with users' interest, and the romantic comedy Accidental Lovers, in which users can watch and influence a couple's relationship. In Accidental Lovers, viewers are able to interact with the ongoing story by sending mobile text messages to the broadcast channel. Changes in the emotional state of the characters and their relationships depend on the existence of some specific keywords found in the viewer's messages. Accidental Lovers was broadcasted several times on Finnish television in late December 2006 and early January 2007 (Williams et al. 2006). Another example of system for the production of interactive narratives is presented by Shen et al. (2009). Their system helps users to compose sequences of scenes to tell stories by selecting video segments from a corpus of annotated clips.

Another example of interactive narrative automatically edited and broadcasted by a TV channels is Akvaario (Pellinen 2000). Similarly to Accidental Lovers, in Akvaario viewers also can influence the mood of the protagonists through mobile text messages. The system uses a large database of clips (approximately 5000), and relies on many features of the database organization to choose the adequate video segments based on the content of the viewer's messages (Manovich 2001).

There are also some examples of video-based interactive narratives for cinema. *Kinoautomat* (Činčera et al. 1967) is one of the first interactive films produced for cinema (Hales 2005). The film comprises nine interaction points, where a human moderator appears on stage and asks the audience to choose between two scenes. Then, the public votes on their desired option by pressing colored buttons installed on the theater seats. Based on the audience votes, the lens cap of two projectors is manually switched to project only the selected scene. Kinoautomat was exhibited for six months and attracted a public of more than 67 thousands of viewers. Following a similar approach, the short interactive film *I'm Your Man* (Bejan 1992) was also exhibited on movie theaters and allowed the audience to interact in six points of the story by choosing between three different options. A more recent interactive experience is Last Call (Jung von Matt 2010), which is an interactive advert for the 13th Street TV Channel exhibited experimentally in movie theaters. In Last Call, the audience interacts with the

actress talking to her via cell phone. Based on the audience voice commands, the system selects the sequence of videos to be presented according to a fixed tree of prerecorded video segments.

In a recent research work, Porteous et al. (2010) present a video-based storytelling system that generates multiple story variants from a baseline video. The video content is generated by an adaptation of video summarization techniques that decompose the baseline video into sequences of interconnected shots sharing a common semantic thread. The video sequences are associated with story events and alternative storylines are generated by the use of AI planning techniques. Piacenza et al. (2011) present some improvements to these techniques using a shared semantic representation to facilitate the conceptual integration of video processing and narrative generation. However, continuity issues are not tackled by their approach. As these video segments can be joined in different orders, several continuity failures may occur, in particular because their system uses video segments extracted from linear films. The planning algorithm ensures only the logical continuity of the narrative, but not the visual continuity of the film.

Another recent research that explores the use of videos in interactive storytelling is presented by Müller et al. (2013). Those authors describe a system for the production and delivery of interactive narratives, whose web-based client interface represents stories using short video snippets. However, as other previous works, their system relies only on static video segments and cinematography principles are not applied to ensure the consistency of presented video stories.

In general, the previous works surveyed here focus basically on the creation of stories by ordering pre-recorded video, without using cinematography concepts. The interactive narratives broadcasted by TV channels and exhibited in theaters are entirely based on predefined branching narrative structures. Moreover, previous works adopt only immutable pre-recorded videos, which reduce interactivity, story diversity, and increase the productions costs. None of the previous works uses video compositing techniques to generate video-based interactive narratives in real-time. The proposed thesis differs from the aforementioned works because it proposes a general model for video-based interactive storytelling based on planning and cinematography theory. The proposed approach uses video compositing techniques in order to create video

sequences representing story events generated by a planning algorithm in real-time.

4.2. System Requirements

Based on the cinematography and interactive narrative theories, we established some basic requisites for a video-based interactive storytelling system:

1. **Interactivity:** Interactivity is the key element of interactive narratives. It differentiates interactive narratives from simple linear stories. However, the level of interaction must be carefully planned. The audience must keep the attention on the narrative content without being distracted by the interaction interface. A video-based interactive narrative must handle user interactions and present the results of the user interventions without breaking the continuity of the narrative. In addition, the interaction interface must support multi-user interactions and be unobtrusive to users that just want to watch the narrative without interactions.
2. **Flexibility:** One of the main challenges when developing the dramatization module of an interactive storytelling system is how to make it generic, flexible and adaptable for the presentation of different story domains. A video-based dramatization system must be flexible and independent of story domain.
3. **Autonomy:** In interactive storytelling, stories are usually generated in real-time and the system must be capable of representing all the stories without human intervention. A video-based dramatization system must be capable of:
    - **a.** Automatically compose the scenes to represent the story events;
    - **b.** Autonomously control the behavior of the characters participating in the action;
    - **c.** Automatically select the best shots during the compositing process;

   d. Automatically select the best music and illumination to express the emotions of the scenes;

4. **Real-Time Performance:** The ability of generating and presenting narratives in real-time is crucial to any interactive storytelling system. In a video-based interactive narrative, the system must be capable of composing video sequences to represent the story events in real-time, without noticeable delays and keeping the visual continuity of the film.
5. **Continuity:** In a film, continuity means keeping the narrative moving forward logically and smoothly, without disruptions in space or time. When the audience becomes aware of continuity errors, they simultaneously become conscious that they are watching a movie, which breaks the storytelling illusion. A video-based interactive storytelling system must be capable of keeping the visual and temporal continuity of the narrative.
6. **Expressing emotions:** Expressing and evoking emotions is a key factor to engage the audience in a narrative. The cinematography theory describes several ways to emphasize emotions by using specific camera shots, camera movements, light and music. A video-based interactive narrative must emphasize the dramatic content of the story by correctly employing cinematography principles according to the emotional content of the narrative to create an attractive and engaging visual representation of the story.

4.3. Operating Cycle and System Modules

Similarly to previous interactive storytelling systems, the main operating cycle of a video-based interactive storytelling system can be divided in three main processes: *story generation*, *user interaction* and *story dramatization*:

1. The *story generation* phase makes use of planning algorithms to create and update the story plot;

2. The *user interaction* phase allows users to intervene in the narrative in a direct or indirect way;
3. The *story dramatization* phase represents the events of the story plot using videos.

The main difference between previous systems and a video-based interactive storytelling system lies in the *story dramatization* phase, which uses videos with living actors to present the story events instead of computer generated 2D or 3D animations.

Each phase of the operating cycle is implemented in a different module. The proposed system is composed of three main modules (Figure 4.1): Story Generator, User Interaction and Story Dramatization, which implement their respective phases in the operating cycle (*story generation*, *user interaction* and *story dramatization*). Each module integrates a dedicated controller in charge of handling the network communication between the components: a Planner Controller for the Story Generator, a Drama and an Interaction Controller for the Story Dramatization, and a Global and a Local Interaction Controller for the user Interaction module. Each controller is responsible for interpreting and managing the messages received from other modules.

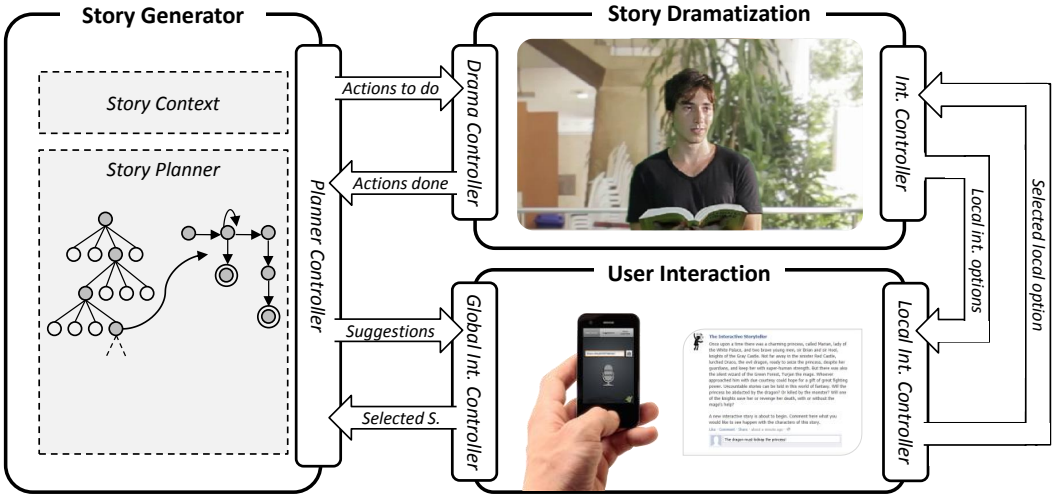


Figure 4.1: System modules.

The three modules were designed to work independently and to run on separate computers, which reduces the computational overhead of running a

complex planning task together with time consuming image processing algorithms for video dramatization. The system adopts a client/server architecture, where the story generator and user interaction modules are both servers, and the story dramatization module is the client interface. This architecture allows several instances of the story dramatization module to be connected with the story generator and the user interaction servers, allowing several users to watch and interact with the same or different stories. The communication between the modules is done through a TCP/IP network connection.

The system adopts the story generation algorithms of Logtell, and consequently follows its approach of generating stories in chapters, which are represented as contingency trees, where the nodes are nondeterministic events and the edges correspond to conditions that enable the execution of the next event. As illustrated in Figure 4.2, a nondeterministic event e_i is executed by a nondeterministic automaton composed by basic actions a_i. The basic actions correspond to the primitive actions that can be performed by the virtual characters during dramatization.

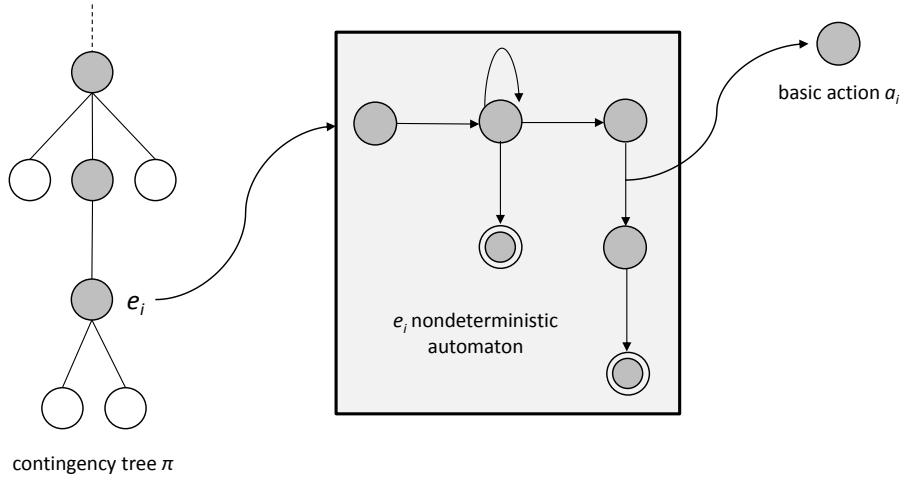


Figure 4.2: Overview of the story generation process.

The system offers two types of user interactions: global and local. In global user interactions, users are able to suggest events to next story chapters, directly interfering in the generation of the contingency trees for the chapters. Such interactions do not provide immediate feedback, but can directly affect the narrative plot. Local user interactions occur during the execution of the nondeterministic automaton and are usually more direct interventions, where users

have to choose between the available options in a limited time. In this type of intervention, users can observe the results of their choices immediately, but such interventions only affect the story plot when the decision leads the execution of the nondeterministic automaton to a different final state.

The system has a dynamic behavior with several tasks running in parallel. Figure 4.3 presents an overview of the behavior of the whole system through an activity diagram, where thick black bars indicate parallel activities. Initially, the story generator module creates the first chapter of the story, according to the initial state of the world, while the dramatization module exhibits an overture. In parallel with the dramatization process, the user interaction module is continuously collecting all the suggestions sent by the users (G facts) and combining them with the facts added (F^+) and removed (F^-) from the current state of the world by the story planner. When the end of a chapter is reached, the facts that are more frequently mentioned by the users and that are not inconsistent with the ongoing story are then incorporated into the story plot. During dramatization, if a local decision point is reached, the user interaction module collects opinions of users to decide the course of the dramatization.

The next sub-sections describe in more details the main operating cycle of the three modules of the system and their respective tasks.

4.3.1. Story Generation

The Story Generator module is in charge of creating and updating the story plot according to user interactions. In every operating cycle, a new chapter of the story is generated. The story generation phase main cycle is composed of six steps: (1) Request Reception; (2) Suggestions Retrievement; (3) Chapter Generation; (4) Automaton Transmission; (5) Suggestions Generation; and (6) Suggestions Transmission.

The first step of the story generation phase is triggered by the reception of a request from the story dramatization module, which can be a request for: (1) the first chapter of a new story; (2) the next chapter of an ongoing narrative; or (3) the next basic event of an ongoing chapter. In the case of a request for the first or the next chapter, the story generator module retrieves all the suggestions given by

users and starts a new planning process in order to generate the story events for the next chapter using the users' suggestions to guide the development of the narrative. Once the chapter has been generated, a new message containing the nondeterministic automaton for the first basic event of the contingency tree of the chapter is constructed and sent back to the story dramatization module. Then, a new set of possible suggestions, based on the possible outcomes of the story, is created and sent to the user interaction module. Otherwise, if the story generator receives a request for the next basic event of an ongoing chapter, the module only creates and sends back a new message containing the nondeterministic automaton for the next basic event of the contingency tree of the current chapter.

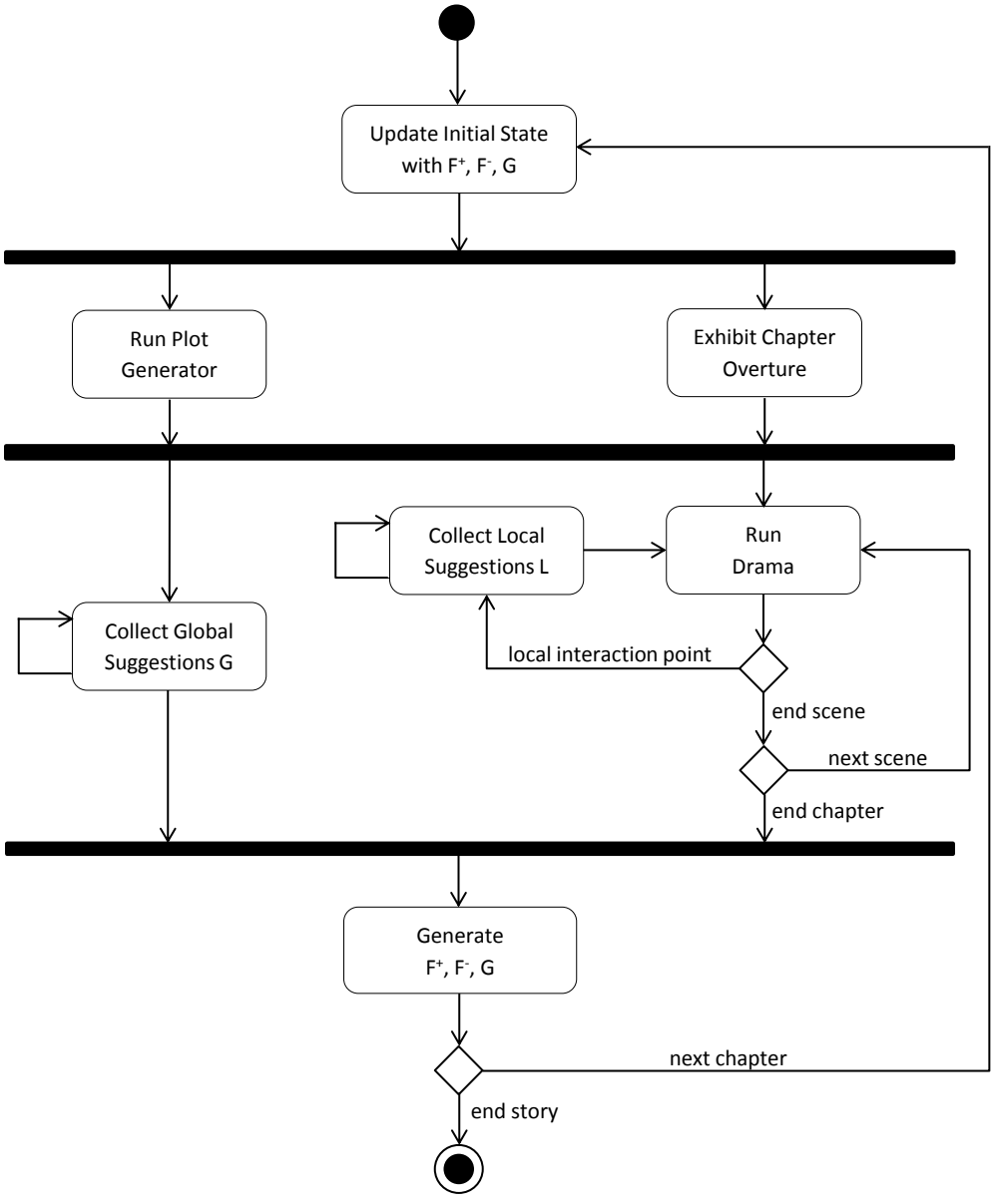


Figure 4.3: Activity diagram of the proposed system.

When a new chapter is requested, the story planner must check the coherence of the user suggestions and compute the story events for the next chapter considering the possible consequences of the user interventions in the rest of the story. However, this is not a trivial task and may become excessively time-consuming. In order to synchronize the process of generation and dramatization, stories are strategically divided into chapters. While a chapter is being dramatized, the story planner can already start generating the future chapters. When user interventions are coherent, they are incorporated in the next chapters. In this way, the system keeps the plot generation some steps ahead of the dramatization, so that chapters are continuously generated and dramatized. While the story is being dramatized, the system tries to anticipate the effects of possible user interventions, so that future chapters will be ready when necessary (Camanho et al. 2009). If the system detects that more time is needed for generating the next chapter, a message is sent to the dramatization module in order to extend the duration of the remaining events in the current chapter, as detailed by Doria et al. (2008).

4.3.2. User Interaction

The user interaction module is in charge of handling and managing multi-user interactions. The user interaction phase cycle is composed of three steps: (1) Suggestions Reception; (2) Vote Collection; and (3) Selected Suggestion Transmission.

The first step of the user interaction phase is triggered by the reception of interaction suggestions, which can be global suggestions generated by the story generator module, or local interaction options received from the story dramatization module. After parsing the suggestions, the process of collecting votes from users starts. Although there is a set of valid global suggestions, users are free to suggest any event for the story. The user interaction module maintains a list of user's desires, which contains the number of votes for each suggestion, even if it is not in the current set of valid suggestions. In this case, if it appears in the set of valid suggestions during a future chapter, it will already have the amount votes previously accumulated.

Global suggestions are continuously collected by the system. When the story generator module requests the results of user interactions, a new message containing the most voted current global suggestion is created and sent back to the story generator module. Local user interventions occur in parallel with the global user interaction. When the system receives local interaction options from the story dramatization module, it shows and collects user votes for the local decision point. In this type of intervention, users are more restricted and have to choose between the available options in a limited time. When the dramatization module requests the results of the local intervention, a new message containing the most voted current local option is created and sent back to the dramatization module. Meanwhile, the system is still collecting global suggestions for the next chapters.

4.3.3. Story Dramatization

The story dramatization is the third process in the main cycle, and is handled by the Story Dramatization module. The dramatization phase cycle is composed of three steps: (1) Automaton Reception; (2) Automaton Execution; and (3) Confirmation Transmission.

The dramatization process is initiated by the story dramatization module after receiving a new automaton with basic actions to perform. The nondeterministic automaton is executed starting from the initial state until it reaches a final state. As previously detailed in Section 2.2, in each automaton, states are described by situations observed in the world, and the transitions between states are associated with basic actions that virtual actors can perform. The basic actions are parsed during the execution of the automaton and delegated to their respective actors. The execution of the automaton progresses to the next state when an actor finishes its performance of a basic action.

In general, there is always a set of states that can be reached after the execution of a basic action and the selection of which transition must occur is based on local user interaction. When starting the dramatization of an action that leads to a decision point, the dramatization module creates a new message containing the local interaction options and sends it to user interaction module. After finishing the execution of the action, the dramatization module retrieves the

most voted option and selects the next action to continue the execution of the automaton. When the execution of the automaton reaches a final state and all the basic actions have been successfully executed, a confirmation message is sent back to the planner controller indicating the end of the dramatization of the current automaton, and requesting a new automaton to continue the narrative based on the final state reached during the dramatization of the current automaton.

4.4. Architecture

The architecture of the proposed video-based interactive storytelling system comprises three main modules: story generator, user interaction and story dramatization.

4.4.1. Story Generator

The story generator module is based on the third version of Logtell, which incorporates the basic temporal modal logic of the first version (Pozzer 2005; Ciarlini et al. 2005), the client/server architecture of the second version (Camanho et al. 2009), and planning under nondeterminism (Silva et al. 2010) combined with the use of nondeterministic automata to control the dramatization of events (Doria et al. 2008) that were introduced in the third version of the Logtell system. Only a few relevant modifications were made in the original architecture and implementation of Logtell story generation module. The main modification is the introduction of a new module called Planner Controller to manage and centralize the communication of the story generator module with the other modules of the system.

Figure 4.4 shows the architecture of the story generator server. In the architecture, story contexts are stored in a database of contexts (Context Database), where each context contains a description of the genre according to which stories are to be generated, and also the intended initial state specifying characters and the environment at the beginning of the story. The Context Control Module stores and provides real-time access to all data of the Context Database. The Simulation Controller is responsible for informing the dramatization module,

at the client side, the next events to be dramatized; receiving interaction requests and incorporating them in the story; selecting viable and hopefully interesting suggestions for users who are intent on performing global interactions; and controlling a number of instances of the Nondeterministic Interactive Plot Generator (NDet-IPG), which is responsible for the generation of the plan to be used as input to the dramatization process. The Chapter Controller is responsible for generating the plot, chapter by chapter, including the treatment of nondeterminism and the parallel processing of multiple valid alternatives at any given moment. The Interface Controller controls the user interventions and centralizes the suggestions made by the users.

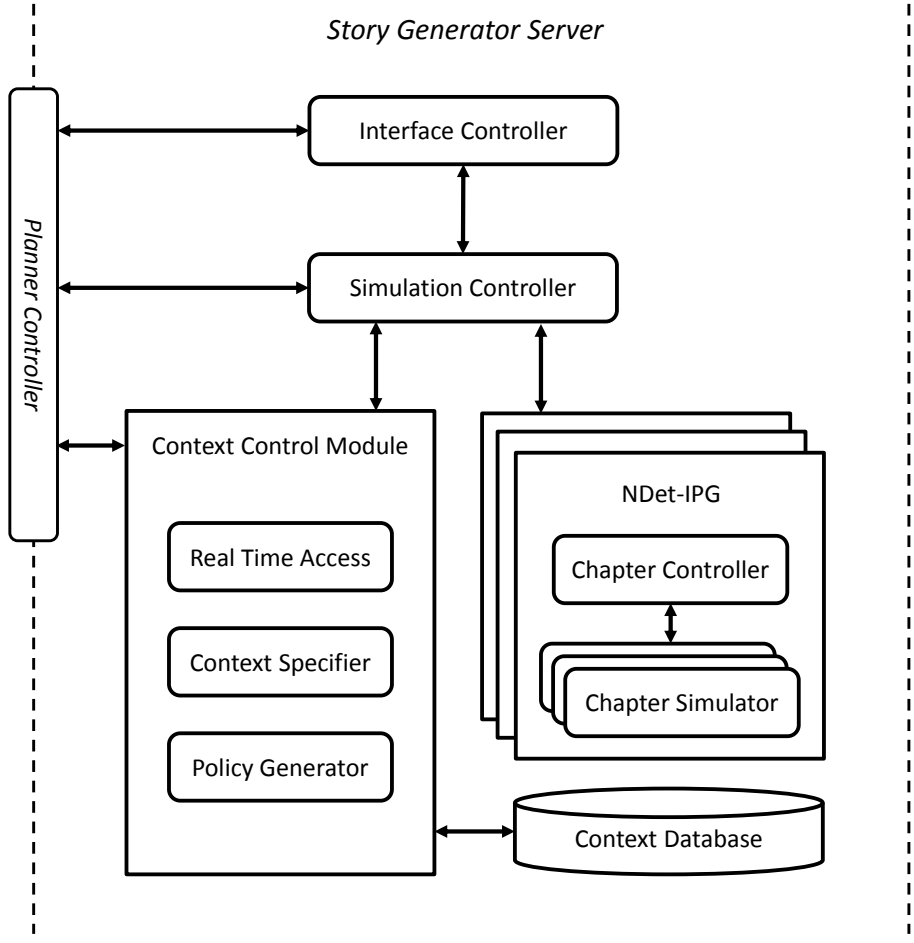


Figure 4.4: The new architecture of the story generator server of Logtell.

More details about the architecture of the Logtell are available in (Pozzer 2005; Ciarlini et al. 2005; Camanho et al. 2009; Silva 2010).

4.4.2. User Interaction

The user interaction module of the proposed system is the result of several studies on user interaction mechanisms for interactive storytelling that were conducted during the development of this thesis (Lima et al. 2011B; Lima et al. 2012B; Lima et al. 2012C). The user interaction module works as a multimodal and multi-user interaction server that supports the integration of several interaction mechanisms based on suggestions (Figure 4.5). In this architecture, the Suggestion Manager is the main module that controls the interaction mechanisms, centralizes the users' suggestions and translates them into valid story suggestions. Each interaction mechanism acts as a multi-user server that has its own client interface, allowing several users to be connected in the same interaction network.

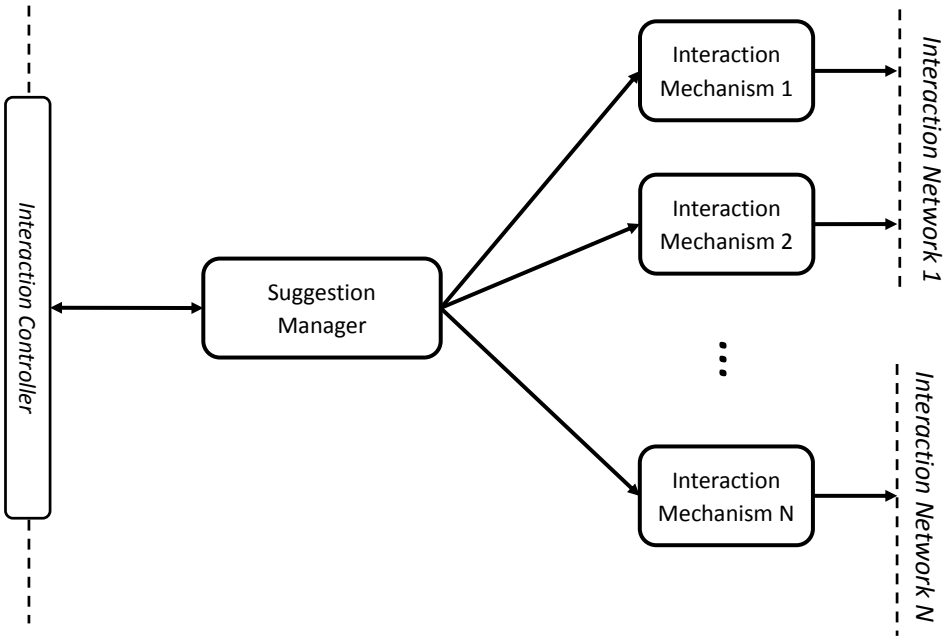


Figure 4.5: Multimodal interaction architecture.

The architecture of the user interaction module integrates two interaction mechanisms: social networks and mobile devices. The first method is based on the idea of using social networks (such as Facebook, Twitter and Google+) as an interaction interface. Three basic ways of interacting with the stories using social networks are: (1) interaction by comments – where users explicitly express their desires through comments in natural language; (2) interaction by preferences – where users express satisfaction or state preferences; and (3) interaction by poll –

where a poll is created and users vote in what they want. In the proposed architecture (Figure 4.6), the modules Interaction by Comments, Interaction by Preferences and Interaction by Poll implement their respective methods of social interaction and are responsible for accessing the social networks looking for user interactions and informing the Suggestion Manager about the user's choices. The second interaction mechanism combines the use of mobile devices (such as smartphones and tablets) with natural language to allow users to freely interact with virtual characters by text or speech. In the architecture, the Mobile Interaction module is responsible for receiving and translating user's advices into valid story suggestions, and informing the Suggestion Manager about the user's interventions.

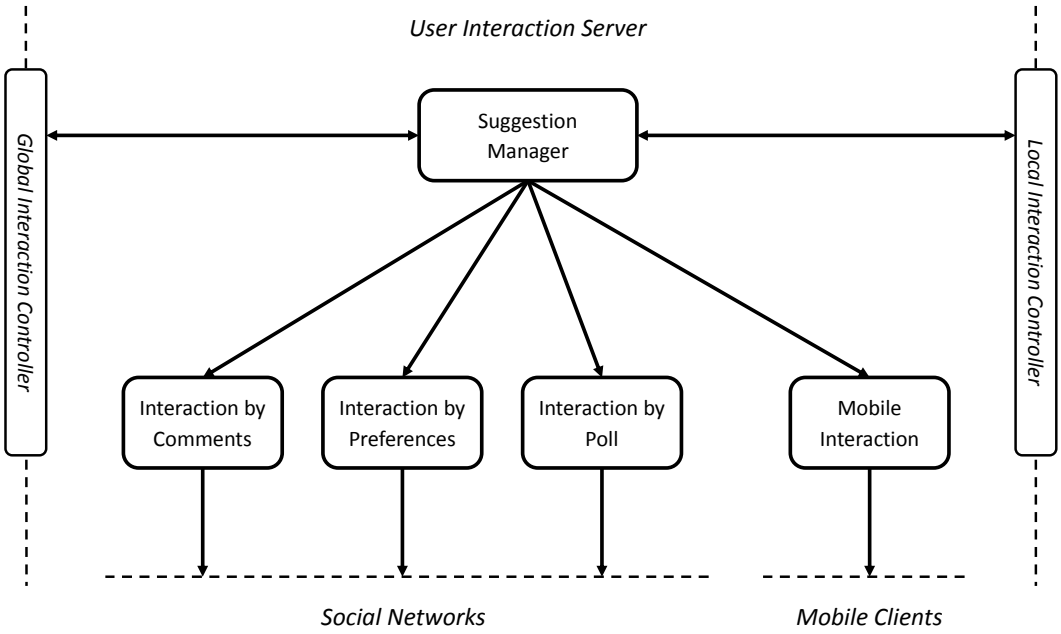


Figure 4.6: The architecture of the user interaction server.

More details about the implementation of the proposed interaction mechanisms are presented in Chapter 7.

4.4.3. Story Dramatization

The architecture of the proposed video-based dramatization module is inspired by the theory of cinematography, and the tasks of the system are assigned

to agents that perform the same roles played by the corresponding filmmaking professionals. This approach has been previously used in the 3D dramatization module of the second and the third version of the Logtell system (Lima 2010), and it has proved to be a good strategy to organize and maintain the modules of the dramatization system.

The video-based base dramatization architecture is composed of several cinematography-based agents (Figure 4.7). The agents share the responsibility of interpreting and presenting the narrative events using videos with living actors. In the proposed architecture, the Scriptwriter is the agent responsible for receiving and interpreting the automata of story events generated by story planner. The Director is responsible for controlling the execution of the nondeterministic automata and the dramatization of the basic actions, and for defining the location of the scenes, actors and their roles. The Scene Composer, using real-time compositing techniques, is responsible for combining the visual elements (video sources) that compose the scenes. The Cameraman is responsible for controlling a virtual camera and suggesting the possible shots (e.g. close-up, medium shot, long shot) for the scenes. The Editor agent, using cinematography knowledge of video editing, is responsible for selecting the best shot for the scenes and keeping the temporal and spatial continuity of the film. The Director of Photography is responsible for defining the visual aspect of the narrative, manipulating the illumination and applying lens filters to improve and create the emotional atmosphere of the scenes. A similar task is performed by the Music Director, which is responsible for creating and manipulating the soundtracks of the film to create the adequate mood and atmosphere of each scene. The communication between the Story Dramatization Client and the other modules of the system is handled by the Drama Controller and the Interaction Controller.

The video-based story dramatization module was specially designed to employ video compositing techniques to generate a visual presentation for the story events. However, it also supports the use of static video sequences to represent scenes. Thus, the system can dramatize both prerecorded and dynamically composed video-based interactive narratives. It can also blend both modalities and use static videos to represent complex scenes that cannot be dynamically composed by the system.

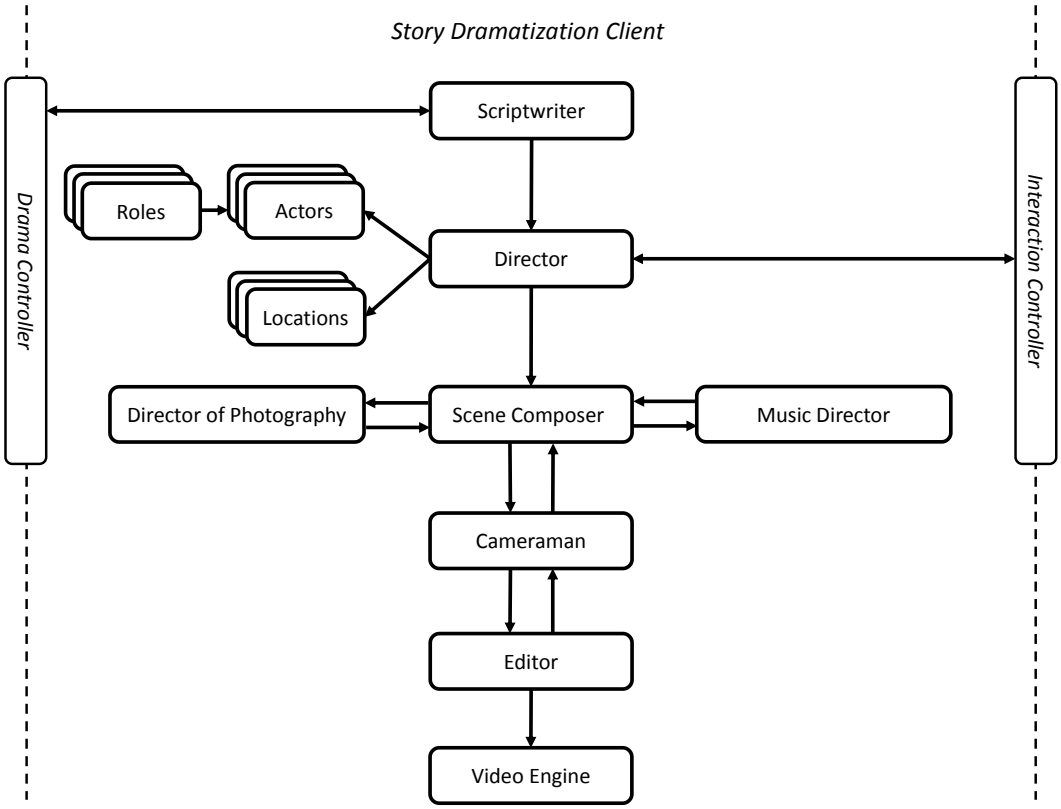


Figure 4.7: The architecture of the video-based story dramatization system.

More details about the implementation of the video-based dramatization system are presented in Chapter 6.

4.5. Conclusion

This chapter presented the general idea of the proposed approach to video-based interactive storytelling, related works, and the architecture of the proposed system together with a high level description of its components and operating cycles. The next chapters present the process for the production of video-based interactive narratives and describe in detail each of the system components.

5
Interactive Film Production

Current advances in interactive storytelling technology have been based mostly on the development of computational methods for the generation, interaction, and dramatization of interactive narratives – focusing mainly in the audience side of the storytelling process. However, in any form of storytelling, the author is the key component for a successful story. The concept of authoring in interactive storytelling has its origins with the own research field, but only recently it has attracted the attention of the research community (Medler and Magerko 2006; Spierling et al. 2006; Pizzi and Cavazza 2008; Riedl et al. 2008; Swartjes and Theune 2009). However, even today, there is a clear imbalance between the number of technical proof-of-concept prototypes and the number and scale of actual interactive narratives.

In an attempt to reduce the gap between interactive storytelling systems and film directors, this chapter presents a general guide on how to write and film interactive stories, and describes some computational tools developed for the production of video-based interactive narratives.

The process of production of a video-based interactive narrative is divided into three phases: (1) pre-production – where the story is logically defined by the author and the shooting script describing how to shoot the film elements is automatically generated; (2) production – where the film set is constructed and the raw elements (actors' actions and locations) are recorded; and (3) post-production – where the raw material is edited, the background is removed, and the video files are associated with the corresponding characters' actions and locations.

5.1.
Pre-production

The first step of the process of production of a video-based interactive narrative is the pre-production phase. It involves the logical definition of the story, including the definition of the characters, their attributes and relations, the

locations where the story take place, the possible events that can occur during the story and the way these events are represented.

5.1.1. Story Definition

Interactive stories are generated according to a story context, which comprises a logical description of the mini-world where the narrative takes place, a definition of the events that can be enacted by the characters, and a description of the motives that guide their behavior.

The process of creating the story context involves work from both story writers and programmers. Initially, the author of the story creates a draft of the general story idea, describing the diverse story elements (e.g. characters' psychology, representative scenes and environment description). Based on these drafts and using cognitive modelling and programming skills, the programmer describes the story world and the various events of the story according to their validity conditions and their consequences.

The elements that compose the story context can be divided in:

- **Static schema**: describes the mini-world wherein the plots take place (characters, relations, places…);
- **Dynamic schema**: describes the possible events in which the characters of the story can participate;
- **Behavioral schema**: describes the motives that guide the behavior of the characters;
- **Detailed schema**: details the description of the story events in terms of actions.

5.1.1.1. Static Schema

The formal specification of the static schema is based on the standard syntax and semantics of first order languages and follows the formalism of the Entity-Relationship model (Batini et al. 1992). The static schema is composed by a set of facts indicating the existence of entities, the values of the attributes of entities, the

existence of relationships, the values of the attributes of the relationships, or the assignment of roles to entities. An entity can represent anything of interest by itself, material or abstract, animate or not. A set of facts holding at a given instant of time constitutes a state.

The clause patterns used in the specification of the static schema are given below. In conformity with Prolog conventions, square brackets are used for conjunctive lists (with "," as separator) and round brackets for disjunctions (with ";" as separator).

```
entity(<entity-class>,<identifier>).
relationship(<relationship-class>,[<entity-class>,...,
                                          entity-class>]).
attribute(<entity-class>,<attribute>).
attribute(<relationship-class>,<attribute>).
boolean(<attribute>).
composite(<attribute>,[<attribute-part>,...,
                                       <attribute-part>]).
is_a(<more-specialized-entity-class>,
                                <more-general-entity-class>).
role(<role>,(<entity-class>;...;<entity-class>)).
```

In the context of a narrative, the major entity classes are usually the characters and places where the story takes place. The characters are identified by their names and may have some attributes describing their physical and emotional characteristics. Similarly, locations are also identified by a name and may have some attributes. Between characters and places there may be some relationships, for example, indicating the home or the current place of a character. Similarly, relationships between characters are also accepted, for example, indicating the affection of one character to another or indicating that two characters are married.

The static scheme is created according to the general idea of the story plot. For example, considering the following plot idea:

"*A young boy named Peter falls in love with a girl named Anne who he met at the university. Advised by two imaginary creatures, a little angel and little devil, Peter tries to know more about Anne by hacking her Facebook page. After getting some information, Peter manages to go out with Anne on a date after a*

party, but she finds out that he invaded her social network account. After so many lies, Anne does not know whether she forgives Peter."

There are four characters that participate in this story (Peter, Anne, Angel and Devil) and at least four different locations (university, party, Peter's house and Anne's house). Based on these assumptions, it is possible to identify five entity classes:

- creature – all the characters are living creatures;
- person – both Peter and Anne are persons;
- adviser – both Angel and Devil are advisers;
- student – both Peter and Anne are students;
- place – all the locations where the story happens are places.

Every student is a person and every person and adviser is a creature. A person has a gender and every living creature has a level of joy and affection for other creatures. The story has protagonists (students) and supporting characters (advisers). The complete entity-relationship diagram of these various components and the connections among them are shown in Figure 5.1.

Based on the definition of the entities and relationships, the static scheme can be specified in a concrete notation:

```
/* Static Schema */
entity(creature, name).
entity(person, name).
entity(adviser, name).
entity(student, name).
entity(place, place_name).
is_a(person, creature).
is_a(student, person).
is_a(adviser, creature).
attribute(person, gender).
attribute(creature, joy).
attribute(acquaintance, affection).
relationship(home,[creature, place]).
relationship(current_place, [creature, place]).
```

```
relationship(acquaintance,[creature,creature]).
relationship(know, [creature, creature]).
relationship(know_more, [creature, creature]).
relationship(seduced, [person, person]).
relationship(debunked,[creature, creature]).
relationship(forgiven,[person, person]).
role(protagonist, (student)).
role(supporting, (adviser)).
```

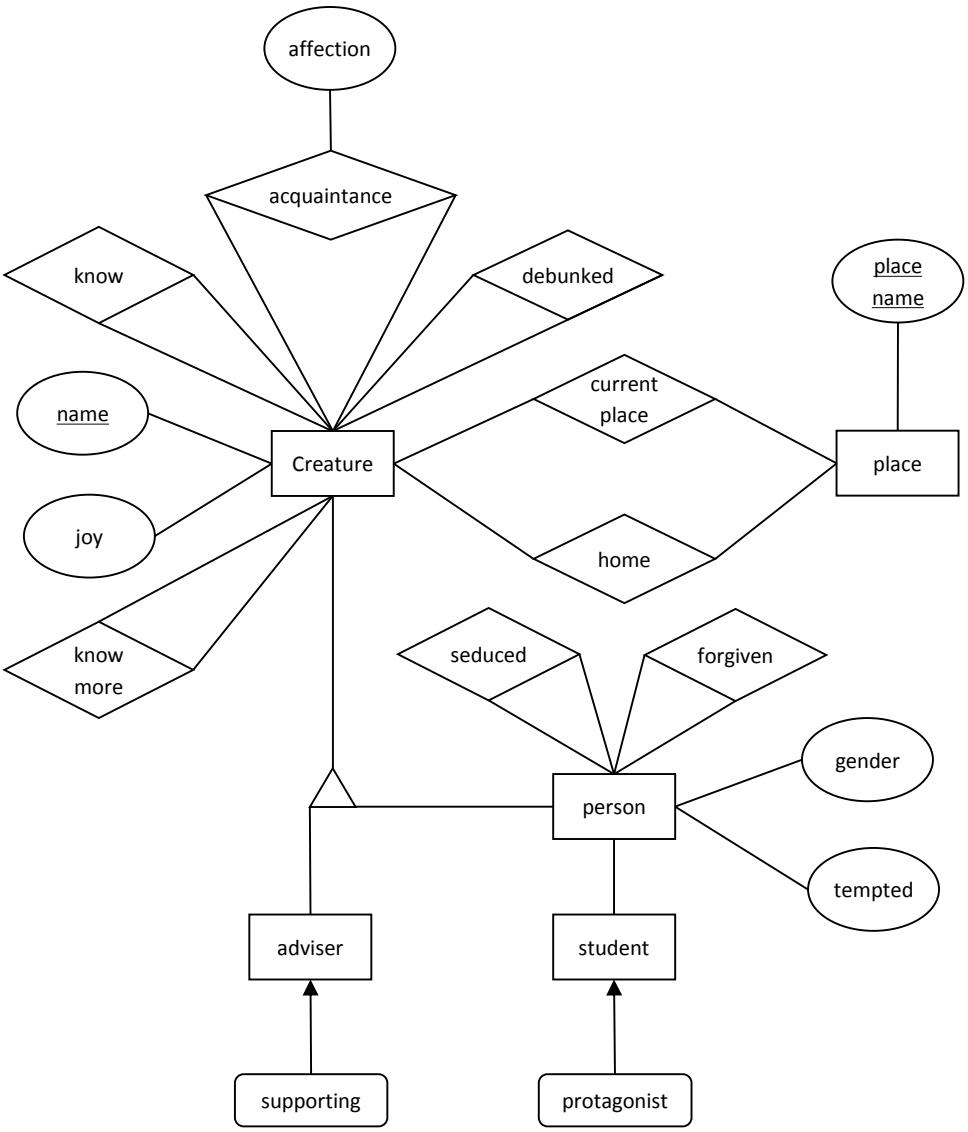


Figure 5.1: Entity-Relationship diagram of the static scheme.

The notation for representing facts is a straightforward consequence of the schema notation. The terms `db(<fact>)`, i.e. database facts, denote the component entries of the initial state of the story. Based on the story plot and the static schema, the initial state of the narrative can be defined:

```
/* Database Facts – Initial State*/
db(protagonist('Peter')).
db(protagonist('Anne')).
db(supporting('Angel')).
db(supporting('Devil')).
db(student('Peter')).
db(student('Anne')).
db(adviser('Angel')).
db(adviser('Devil')).
db(gender('Peter', male)).
db(gender('Anne', female)).
db(joy('Peter', 5.0)).
db(joy('Anne', 10.0)).
db(place('University')).
db(place('Party')).
db(place('PeterHouse')).
db(place('AnneHouse')).
db(home('Peter', 'PeterHouse')).
db(home('Anne', 'AnneHouse')).
db(current_place('Peter', 'University')).
db(current_place('Anne', 'University')).
db(affection(['Peter', 'Anne'], 0.0)).
db(affection(['Anne', 'Peter'], 0.0)).
```

The state of the world at a given instant consists of all facts about the existing entity instances and their properties (attributes and relationships) holding at that instant. A genre is compatible with an ample choice of (valid) initial states. Different initial states lead to the development of possibly very different narratives, all of which are constrained to remain within the limits of the defined genre.

5.1.1.2. Dynamic Schema

After defining the static schema, the next step in the specification of the story context is the definition of the dynamic schema, which includes the declaration of the basic operations.

A narrative event is defined by a transition from a valid world state S_i to another state S_j, which should also be valid. Changes in the world state must be

limited to what can be accomplished by applying a limited repertoire of operations, which will define what kind of events can occur in the narrative. The operations are formally specified using the STRIPS formalism (Fikes and Nilsson 1971). Each operation is defined in terms of its pre-conditions and post-conditions. Pre-conditions are conjunctions of positive or negative database facts, which must hold at the state in which the operation is to be executed. Post-conditions (effects) consist of two sets of facts: those to be asserted and those to be retracted as a consequence of executing the operation. The events can be either deterministic or nondeterministic. While a deterministic event has only one possible outcome, a nondeterministic event can have multiple outcomes, which are defined by adding alternative sets of effects (post-conditions) to the operations. Guaranteeing the transition between valid states depends on a careful adjustment of the interplay among pre- and post-conditions over the entire repertoire of operations.

The dynamic schema is specified with the following syntax, wherein each operation is defined by an operation frame and an operation declaration:

```
operator_frame(<operator-id>, <operator-name>,
              [<case>:(<entity class or role>;...;
                       <entity class or role>),...,
               <case>:(<entity class or role>;...;
                        entity class or role>]).

operator(<operator-id>,<operator-name>(<parameter list>),
         [<pre-conditions>],
         [<effects>],
         <estimated cost of operation>,
         [<main effects>],[],[]).
```

The operations are specified according to the possible events that can occur on the narrative. In the example given in the previous section, we can easily identify two possible events for the narrative in the first sentence of the plot: "*A young boy named Peter falls in love with a girl named Anne that he met at the university*". Clearly, there is a "meet" event, where a character meets another character, and there is a "fall in love" event, where a character falls in love with another character. We can also think about some pre-conditions that must hold to

these events to occur: in order to meet someone, both characters must be at the same place; and to fall in love, the characters must know and have some affection for each other. Similarly, some post-conditions (effects) can be established: after meeting someone, the character gets to know the other one; and after falling in love, the character begins to love the other one. Both events are deterministic and have only one possible outcome.

Based on these ideas, we can formally specify the operations in a concrete notation:

```
operator(1, meet(CH1,CH2),
        [
           current_place(CH1,PL),
           current_place(CH2,PL),
           not(know([CH1,CH2]))
        ],
        [
           know([CH1,CH2])
        ],
        [],10, [know([CH1,CH2])],[],[]) :-
           db(protagonist(CH1)),
           db(protagonist(CH2)),
           dif(CH1,CH2).

operator(2, fallinlove(CH1,CH2),
        [
           know([CH1,CH2]),
           affection([CH1,CH2],A),
           {A > 5}, {A < 10}
        ],
        [
           not(affection([CH1,CH2],A)),
           affection([CH1,CH2],10)
        ],
        [],10, [affection([CH1,CH2],10)],[],[]) :-
           db(protagonist(CH1)),
           db(protagonist(CH2)),
           dif(CH1,CH2).
```

Apart from the primitive domain operators, an operator can be either an abstraction of one or more specific operators and/or a composition of partially ordered sub-operators. In both cases, they are called complex operators. Abstraction relates a parent operator to one or more child operators. If the parent operator is also a composite operator, its children will inherit its composition. Children operators can also add new operators to the composition, and even include new operators amongst the inherited sub-operators.

Considering the plot example presented in the previous section, we can imagine a possible event where a character attempts and fails to seduce the other one. There are several ways a character can try to seduce someone: a direct and polite approach; a more casual and ironic approach; or even a more aggressive approach. This event can be modeled as a generic operator that has three different specializations (Figure 5.2).

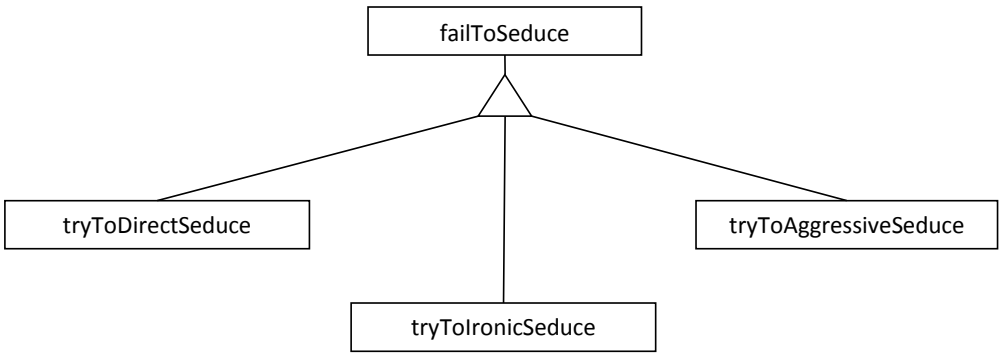


Figure 5.2: Example of generic operator that has three different specializations.

According to these definitions, the specializations of the genetic operator `failToSeduce` can be formally specified in a concrete notation:

```
specialise(tryToDirectSeduce(CH1,CH2),
                             failToSeduce(CH1,CH2)).
specialise(tryToIronicSeduce(CH1,CH2),
                             failToSeduce(CH1,CH2)).
specialise(tryToAggressiveSeduce(CH1,CH2),
                             failToSeduce(CH1, CH2)).
```

In the same plot example, we can also imagine a possible event where a character tries to find more information about another character that he/she just

met by following him/her. This event can be composed of other basic events, which may involve the character going to other places to follow its target. This event can be logically modeled as a composite operator that has four sub-operators (Figure 5.3).

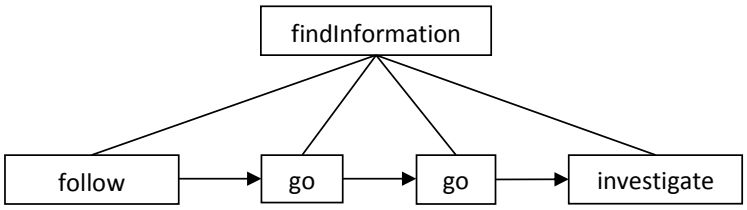


Figure 5.3: Example of composite operator that has four sub-operators.

Based on these specifications, the composite operator for the `findInformation` event can be formally defined:

```
operator(8, findInformation(CH1,CH2),
         [
           not(knowMore([CH1,CH2])),
           affection([CH1,CH2],10)
         ],
         [
           knowMore([CH1,CH2])
         ],
         [],10,[knowMore([CH1,CH2])],
         [
           (f1, follow(CH1, CH2)),
           (f2, go(CH2,PL2)),
           (f3, go(CH1,PL2)),
           (f4, investigate(CH1,PL2))
        ],
        [(f1,f2),(f2,f3),(f3,f4)]) :-
              db(student(CH1)),
              db(student(CH2)),
              dif(CH1,CH2).
```

where the two last parameters of the operator indicate the sub-operators and their partial order. It is important to notice that all sub-operators also must be specified along the dynamic schema, and they can also be composite or abstract operators.

5.1.1.3. Behavioral Schema

The behavioral scheme describes the motives that guide the behavior of the characters in the narrative. It consists of a set of goal-inference rules that capture the goals that motivate the characters' actions when certain situations occur during the narrative. Whenever goals are inferred but not achieved within a plot, a plot composition procedure considers the possible extensions to the plot so that these goals are fulfilled.

An example of goal-inference rule for the narrative idea presented on previous sections could be: "A lonely protagonist wishes to fall in love with another protagonist", which can be formally specified as:

$$\Box(\text{protagonist(CH1)} \wedge \text{protagonist(CH2)} \wedge \text{affection(CH1, CH2)} < 10) \Rightarrow \Diamond(\text{affection(CH1, CH2)} \geq 10)$$

where $\Box$ and $\Diamond$ are the temporal modal operators "always holds" and "eventually holds" respectively; and $\wedge$ is the logical symbol of conjunction.

In the story context, goal-inference rules are specified in a temporal modal logic formalism using a clause of the form `rule(<situation>,<goal>)`, where the `goal` will motivate the agents when a certain `situation` occurs during the narrative. Both `situation` and `goal` are conjunctive lists of literals, denoted using square brackets and "," as separator. The following meta-predicates are used to specify the occurrence of an event or the truth value of a literal at certain times:

`h(T,L)` - the literal `L` is necessarily true at time `T`
`p(T,L)` - the literal `L` is possibly true at time `T`
`e(T,L)` - the literal `L` is established at time `T`
`o(T,E)` - the event `E` occurred at time `T`

Using this notation, the goal-inference rule "A lonely protagonist wishes to fall in love with another protagonist" can be specified in a concrete notation:

```
rule([
       e(i,protagonist(P1)),
```

```
  e(i,affection([P1,P2],A)),
   h({A<10})
],
([T2],
[
  h(T2,affection([P1,P2],10))
],true))
```

Goal-inference rules do not determine specific reactions for the characters – they only indicate goals to be pursued somehow. The events that will eventually achieve the goals are determined by the planning algorithm.

5.1.1.4. Detailed Schema

The final stage of the specification of a story context consists of defining the detailed schema, which includes the definition of the nondeterministic automata used to detail the dramatization of the narrative events.

The events of the narrative are described by means of nondeterministic automata, which are specified and associated with the basic operations defined in the dynamic schema. Each transition in the automaton corresponds to an action that maps a state into a set of possible successor states. The automaton specifies an initial state, at which the event's pre-conditions hold, and a set of final states, at which the event's post-conditions hold. In addition, facts valid at the initial state that are not modified by the post-conditions are also assumed to hold at the final state. The automaton provides therefore various alternatives for the dramatization of each event in a plot, all logically consistent due to the preservation of the correct chaining of pre- and post-conditions.

In each automaton, states are described by invariants, expressed as logical formulae involving dramatization control variables. Transitions between states correspond to basic actions that can be directly performed by the actors. States that have more than one adjacency define local decision points, where users can decide which action the actors should take.

Figure 5.4 shows an example of automaton that describes the dramatization of the event `follow(CH1,CH2)`, where a character CH_1 follows another character CH_2. In the initial state S_1, both characters are in the same place. Then CH_2 goes to

another place and CH_1 stats following CH_2, which leads the automaton to state S_2. Next, CH_2 senses that something is wrong and look back to see what is happening, which induces CH_2 to hide (states S_3 and S_4). If CH_2 noticed that someone was following him/her, he/she enters into the ladies' room (S_7) and CH_1 loses his/her track, which leads the automaton execution to the final state S_8. Otherwise, if CH_2 did not notice CH_1, he/she enters into the classroom (S_5) and CH_1 gets the information about CH_2 classroom (final state S_6). There is also a loop in state $S4$, which makes CH_1 to keep following CH_1 until he/she arrives at the classroom or notices that he/she is being followed and enters into the ladies room.

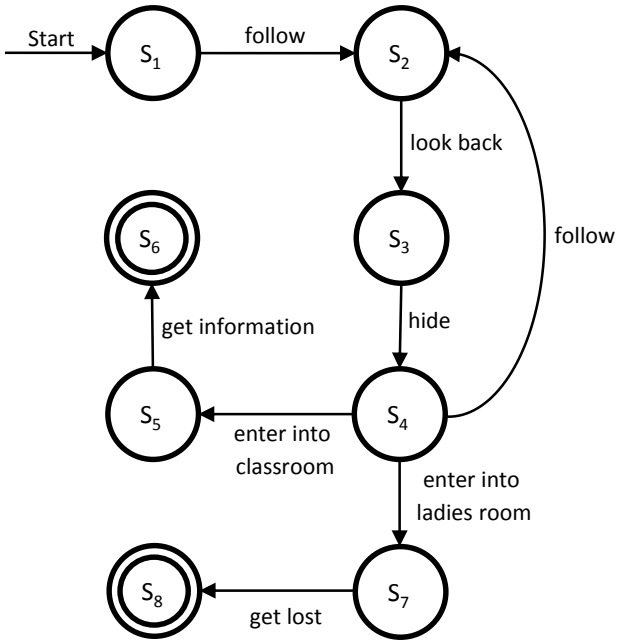


Figure 5.4: Example of automaton describing the dramatization of the event follow(CH_1, CH_2).

The nondeterministic automatons are specified in the GEXF (Graph Exchange XML Format),[4] which is an XML-based language for describing graph structures. The structure of the GEXF file that describes the automatons is illustrated in Figure 5.5.

The specification of the automata demands some authorial effort, but with few states and transitions it is possible to create very flexible dramatizations that preserve coherence within any plot containing the corresponding events.

[4] GEXF File Format - http://gexf.net/format/

```
<?xml version="1.0" encoding="UTF-8"?>
<gexf xmlns="http://www.gexf.net/1.2draft" version="1.2">
  <graph defaultedgetype="directed">
    <attributes class="edge">
      <attribute id="0" title="basicaction" type="string"/>
    </attributes>
    <attributes class="node">
      <attribute id="0" title="state" type="string"/>
    </attributes>
    <nodes>
      <node id="0" label="StateName">
        <attvalues>
          <attvalue for="0" value="StateFacts"/>
        </attvalues>
      </node>
      .
      .
      .
    </nodes>
    <edges>
      <edge id="0" source="0" target="1">
        <attvalues>
          <attvalue for="0" value="BasicAction"/>
        </attvalues>
      </edge>
      .
      .
      .
    </edges>
  </graph>
</gexf>
```

Figure 5.5: Structure of the GEXF file that describes the nondeterministic automatons of the narrative.

5.1.2. Shooting Script Generation

The logical definition of the story context provides the basic information the system needs to automatically generate plots. However, it still needs the video resources to represent them during the dramatization phase. In order to simplify the process of recording the video material required for dramatization, we developed an application that uses the logical definition of the story to automatically generate a shooting script describing how to film the necessary elements (actors and locations) to represent the video-based interactive narrative.

A shooting script is a technical document used in the film industry to specify the sequence of shots that need to be filmed during the production phase

(Swain and Swain 1998). It contains a very elaborate description of all shots, locations, character, action, sound and technical details of the film. A common format for this document is called "Two Column Shooting Script", which consists in dividing the document in two columns: one containing a description of the shot and other with the respective dialog or sound effects. Figure 5.6 shows an example of a two column shooting script used in filmmaking.

Shots	**Shot Description**	**Audio and Dialog**
1	CU of Mother pleading with the father	Mother – sad "*I can't live like this anymore!*"
2	MS of father's reaction	Father – sternly "*We have no options*"
3	LS of father and mother quarreling through a door. Son enters into the frame from the right into foreground in WA and watches them	Mother – upset "*I just can't deal with this violence*" … cries
4	CU of the father	Father – seriously "*What else can we do!*"
5	ECU of the son with an angry look on his face	
6	ECU of the pistol and his fingers on the trigger	

Figure 5.6: Example of a two column shooting script. Abbreviations: CU - close up; MS - medium shot; LS - long shot; WA - wide angle; ECU - extreme close up.

In order to generate a shooting script document, we developed an application that uses the logical definition of the story to simulate all the possible storylines that can be created based on the story context. This process generates a large tree of events, where each node corresponds to an instance of a logical operator representing a story event. The nodes of the tree are then used to instantiate their respective nondeterministic automata (initializing all variables of the basic actions). Then, all basic actions are added to a list structure that contains a logical description of all the possible basic actions that may be performed by the actors during the narrative.

The shooting script document is created based on the list of basic actions. First, the basic actions performed by each actor are grouped and duplicate actions of the same actor with the same or different parameters are removed from the list. Then, for each basic action of each actor a simple natural language sentence is created based on the logical description of the action and a simple text template. For example, the basic action `look(anne, peter)` is converted to the sentence: "*LS of Anne looking at someone*", where the predicate indicates the verb of the sentence and the first variable symbol indicates the character performing the action. The other parameters are generic and may refer to other characters or objects depending on the instance of the action, thus they are omitted and replaced by "someone" or "something". A similar procedure is adopted in dialog actions, with the addition of a step where the content of the dialog is extracted to be incorporated into the dialog column of the document.

Figure 5.7 shows a segment of shooting script automatically generated by the system, where the first column indicates the number of the shot, the second presents a description of the shot, and the third column includes the dialog when necessary. We also added a fourth column that indicates whether the action requires the production of a loop sequence or not. However, the system cannot automatically predict when a loop sequence is required based only in the story context, thus this information needs to be manually added to the document by a human expert.

Shots	**Shot Description**	**Dialog**	**Loop**
1	LS of Anne walking		Yes
2	LS of Anne smiling		No
3	LS of Anne kissing someone		No
4	LS of Anne	Anne – "*I'm waiting for someone*"	No
5	LS of Anne	Anne – "*Well, this is impossible!*"	No
6	LS of Anne	Anne – "*Today was really nice.*"	No

Figure 5.7: Segment of a shooting script automatically generated by the system.

The final shooting script document is divided into sections that separate the shots of each actor individually. In addition, the locations of the narrative that must be filmed are also automatically listed at the end of the document. An

example of a full shooting script document generated by the system can be found in (Lima and Feijó 2014).

5.2. Production

Once the story context has been logically defined and the shooting script has been generated, the next step consists in filming the video resources necessary for the dramatization of the narrative according to the instructions provided by the shooting script document.

The proposed video-based interactive storytelling system is based on the use of video compositing techniques to dynamically create video sequences to represent the story events generated by planning algorithms. In order to compose the scenes in real-time, the system uses as input videos that must pass through a pre-processing phase to remove the background from the videos. The first step of the production phase consists in defining which matting technique will be used in order to extract the visual elements from the background so they can be used in the compositing process.

The matting techniques can be divided in hardware-based (Joshi et al. 2006; Sun et al. 2006; McGuire et al. 2005) and software-based methods (Sun et al. 2004; Gastal and Oliveira 2010; Wang and Cohen 2007). Hardware-based methods usually rely on additional information provided by special equipment, which effectively enhance the efficiency, but increases the production costs. By contrast, software-based methods do not rely on special equipment; they work directly with the visual information provided by the video frames.

Examples of hardware-based matting techniques include the use of an array of cameras (Joshi et al. 2006), which uses the relative parallax between the frames produced by the aligned cameras to capture the foreground objects in front of different parts of the background; another example is the flash matting system (Sun et al. 2006), which uses flash/no-flash image pairs to extract alpha masks based on the observation that only the foreground object has the most noticeable difference between the images if the background is sufficiently distant.

Examples of software-based matting techniques include the Poisson matting algorithm (Sun et al. 2004), which operates directly on the gradient of the matte

based on the assumption that the intensity change in the foreground and background is smooth and thereby the gradient of the matte matches with the gradient of the image; another example is the Shared Matting (Gastal and Oliveira 2010), which collects samples by shooting rays from observed pixels to background and foreground, then the samples that lie along the rays are collected and used for alpha estimation. A comprehensive review on image and video matting techniques is presented by Wang and Cohen (2007).

The proposed method for video-based interactive storytelling receives as input alpha masks, which can be generated by all alpha matting techniques. For the experiments conducted during the development of this thesis, we selected the chroma key matting method, which is the most common matting technique used in the film industry today (Foster 2010). It involves shooting the visual elements in front of a green or blue screen, and then using an algorithm to remove the background from the shot based on the color range of the colored screen.

Once the matting technique has been defined, the next steps of the production phase involve the process of building the film set and filming the raw elements of the narrative (actors' actions and locations) according to the shooting script generated by the system. The next sub-sections describe these steps.

5.2.1. Set Construction and Camera Setup

The construction of the film set involves the process of defining the shooting procedure, building the green screen and placing the cameras in the set.

As previously mentioned, both actors and locations are filmed from different angles in order to give to the system the freedom to dramatize scenes from different angles and apply the basic cinematography concepts during the dramatization of the narrative. In addition, the actors are filmed in front of a green screen, which allows the system to remove the background using the chroma key matting technique and dynamically compose the scenes of the narrative without being restricted by static video sequences. A total of 8 angles of the actors performing their actions must be recorded using a single or multiple cameras in front of a green screen with intervals of 45 degrees (forming a circle around the

subject). The first step of the production phase consists of defining how these 8 angles of subject will be shot.

Four shooting procedures are proposed: (1) full circle filming setup; (2) semicircle filming setup; (3) one-quarter filming setup; and (4) single camera filming setup.

The full circle filming setup consists of using 8 cameras to record the action simultaneously (Figure 5.8). This can be done by building a cylindrical structure with the interior coated with a green screen fabric. The cameras are placed outside and around the structure with the lens embedded in small holes built in the cylindrical wall with intervals of 45 degrees. The actor is placed at the center of the structure and can perform the actions while the cameras record his performance. The cameras have to be placed outside to avoid that two facing cameras film each other. Lights, microphones and other filming equipment can be positioned over the structure.

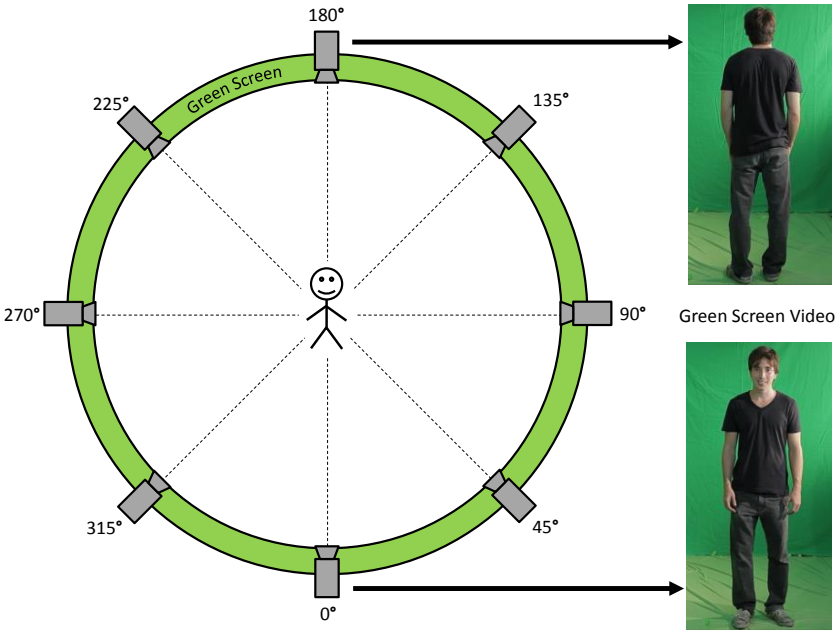


Figure 5.8: Full circle filming setup.

The main advantage of the full circle filming setup is that it makes easier and simplifies the work for both cinematographers and actors. The actors only need to perform the actions once and the cameras record all angles simultaneously without requiring adjustments in the camera setup. The main drawback is the growth of the productions costs. It requires eight equivalent cameras and the construction of the cylindrical film set.

The second shooting procedure is the semicircle filming setup (Figure 5.9). It consists of a simplified version of full circle filming setup, where only half of the circle is created. In this method, three green screen walls are built and arranged as illustrated in Figure 5.9. Inside of the walls, 5 cameras are placed forming a semicircle around the subject. The two cameras placed at straight line angles (0° and 180°) are positioned in two small holes created in the parallel walls. The actor is placed at the center of the semicircle and five angles of the actions are recorded simultaneously. In order to obtain the missing angles, there are two options: (1) the missing angles are obtained by flipping the videos of cameras 45°, 90° and 135° horizontally; or (2) the actor turns 180 degrees and repeats the action.

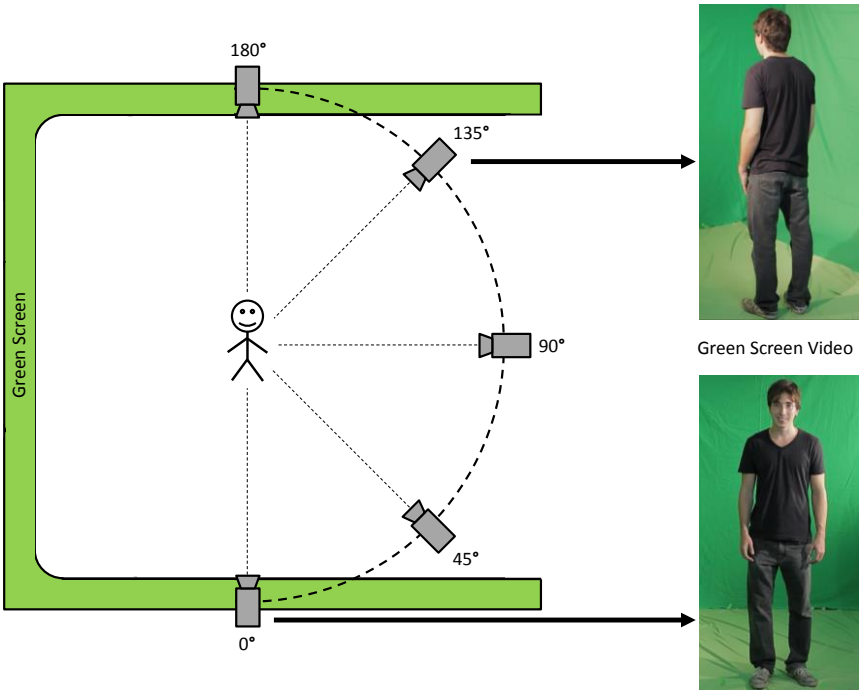


Figure 5.9: Semicircle filming setup.

The main advantage of the semicircle filming setup is the reduction of the production costs. It requires less cameras than the full circle setup and can produce similar results without increasing the work of actors and cinematographers. The main disadvantage is that the trick of flipping the videos of some cameras to produce the missing angles only work well when the actors, their clothes and props are fully symmetrical; otherwise it may break the continuity of the film. For example, if an actor is holding a handbag in his right hand when filmed from the angle of 45 degrees, in the flipped video of 315 degrees he will

appear holding the handbag in his left hand, which may confuse the audience. In order to avoid this problem, when the actors are not symmetrical, the actor has to repeat the action two times to record his performance from all camera angles.

The semicircle setup can also be used in the cylindrical structure of the full circle setup, and indeed it may produce better results regarding the quality of the green background. However, building three green screen walls may be easier that constructing the cylindrical structure.

The third proposed shooting method is the one-quarter filming setup (Figure 5.10). It consists of a more simplified version of semicircle filming setup, where only one-quarter of the circle of cameras is created. In this method, only two green screen walls are built and positioned as illustrated in Figure 5.10. Next to the walls, 3 cameras are placed forming one-quarter of a circle around the subject. In this camera setup, three angles of the actions performed by the actors are recorded simultaneously. In order to obtain the missing angles, the actor has to repeat the actions two or four times. If the actor and his clothes/props are fully symmetrical, the action is repeated two times to obtain the angles of a semicircle, and the angles of the other side of the semicircle are obtained by flipping the recorded angles. Otherwise, if the actor is not symmetrical, the action has to be repeated four times so that all angles of the actor are recorded.

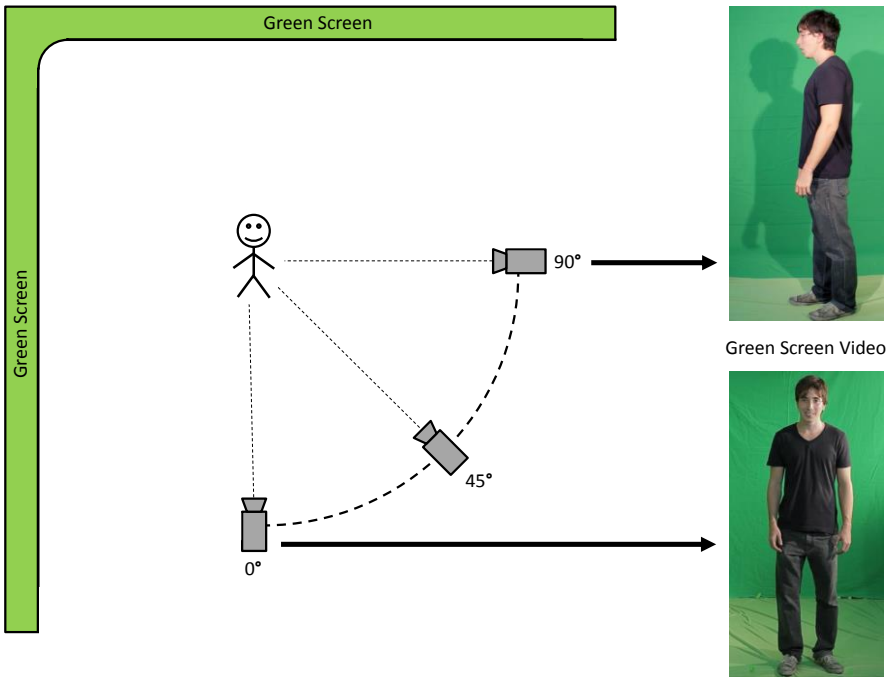


Figure 5.10: One-quarter filming setup.

The main advantage of the one-quarter filming method is the notable reduction of the production costs. However, it significantly increases the work of actors, who need to repeat the actions several times. Although there is no need to the actions to be repeated exactly the same way, it is important that they have some degree of similarity, which may be difficult to be achieved by nonprofessional actors.

The last shooting method is the single camera film setup (Figure 5.11), where all the angles of the action are recorded using only one camera. The method uses a single green screen wall and the actor is positioned between the wall and the camera (Figure 5.11). In order to record all required angles, the actor has to perform the action, turn 45 degrees, repeat the action, and do this until he completes the eight angles of the circle. Again, the number of times the actors has to repeat the actions can be reduced in half by flipping the recorded angles, but only if the actor and his clothes/props are fully symmetrical.

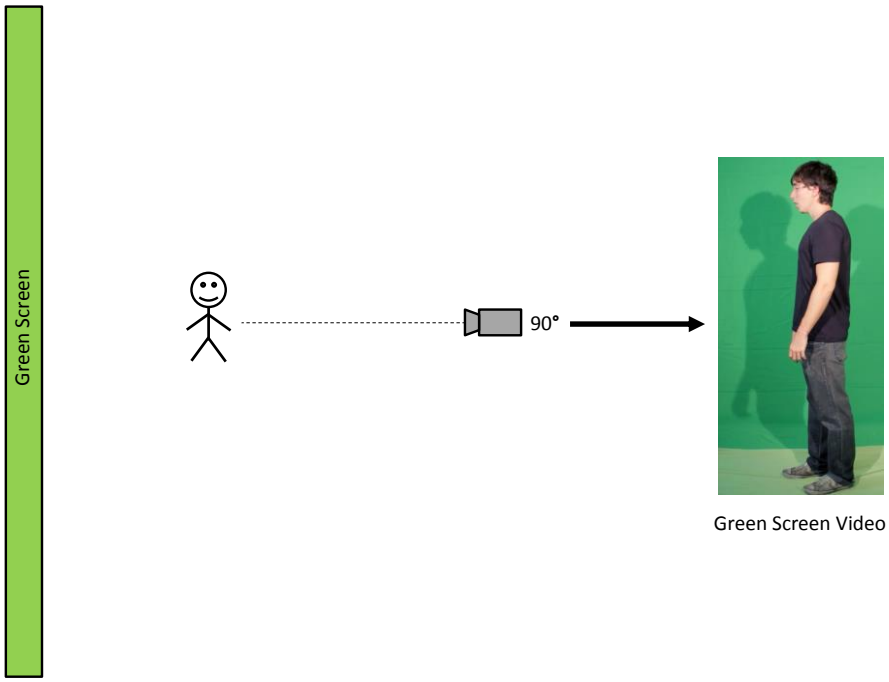


Figure 5.11: Single camera film setup.

The great advantage of the single camera setup is the huge reduction of the production costs. However, it excessively increases the number of times the actors have to repeat the actions, which is a very exhaustive task and may compromise

their performance. Although it is possible to film the videos of a full interactive narrative using this method, it is only recommended for testing purposes.

In the four shooting procedures, the distance of the cameras to the subject depends on the focal length of the camera. A longer focal length produces a narrower angle of view, whereas a shorter focal length produces a wider angle of view. All the cameras must be positioned in the film set in a way to capture the whole body of the actors in the frame.

5.2.2. Shooting Actions

Once the filming method is defined and the film set is constructed, the next step of the production phase consists of shooting the actors performing their actions. The shooting script generated by the system provides a basic description of the actions that have to be recorded, however it is a task of the film director to supervise and give more detailed instructions to the actors about the actions they need to perform.

If a full circle filming method is used, the actors only have to repeat each action once; otherwise, they have to repeat the same action more times so that all required angles are recorded. When the action needs to be repeated, there is no need to the action to be performed exactly the same as in the previous shot. The dramatization system will not switch between different angles during the exhibition of an action. The camera angle will be selected only at the beginning of the action, so the audience will not notice differences in the actors' performance. However, it is important that the performance have some degree of similarity in the different angles to avoid inconsistencies when transiting between different actions. The film director must supervise the shooting process to guarantee the coherence in the recorded actions.

One of the main rules that must be obeyed when shooting the actors is that all actions must be performed in-place, which means that the actor must stay always on the same position while acting. During the dramatization, the system will be in charge of automatically positioning and controlling the movement of the actors, so it is important that all actions be recorded without significant changes in the position of the actors. For most part of the actions this is not a problem,

however, some actions naturally require movements of the actor on the stage. A very common example is the basic action of walking, which is present in most part of the narratives and requires the explicit movement of the actors. In order to record this type of action, we propose the use of a treadmill positioned at the center of the filming set (Figure 5.12). The treadmill allows the actors to walk or run while staying in the same place. In our experiments, we used a simple exercises treadmill, which produced good results and allowed us to film the actors walking and running in-place. However, for more professional results it is recommended the use of a more personalized treadmill that could be embedded in the structure of the film set. In addition, the color of the treadmill must be the same as the color of the background, so it can be easily removed from the video during the post-production phase.

Figure 5.12: In-place walking action being performed over a treadmill.

Another important aspect to consider when filming the actors is whether the videos of the actions need to be in loop or not. A looping video consist of sequence frames that can be repeated continuously without jumps. Simple and direct actions, such as talking, looking and grabbing, do not require looping videos, because the end of the video indicates the end of the action. However, more dynamic actions, where the end of the video not necessarily indicates the end of the action, require videos ready to be played in loop by the system. For

example, it is not possible to predict the precisely length for the videos representing a walk action because it depends on the distance traveled by the virtual actor, which may vary from scene to scene. Another example is the idle action, which is used to represent supporting characters that are participating of the scene without performing any specific action. Again, the length of the video depends on the length of the scene. In these cases, the videos must be prepared to be played in loop, so the system can represent the actor walking any distance or staying in the idle position for an indefinite time.

Although the actual process of creating the looping videos is done in the post-production phase, it is important that the recorded material allows the creation of looping sequences. The recommendation is to film long sequences of the actions that require looping sequences being repeated by the actors, so the editors will have sufficient material to work with during the post-production phase. In our experiments, the looping actions were recorded for 5 minutes, which provided enough material to create the looping videos.

5.2.3. Shooting Locations

The locations are composed of a set of video or image layers representing the environments where the story events can occur. Usually, outdoor locations are represented through videos, which are able to reflect the natural dynamism of the environment (e.g. leaves being moved by the wind, people walking in the distance, birds flying around), and indoor locations that do not include dynamic elements are represented by static pictures. The process of shooting the locations of the narrative is easier than shooting the actors. It does not require the construction of a filming set and the actors do not have to go with the filming crew to the physical location.

The shooting script generated by the system provides a basic description of the locations that must be filmed based on the logical context of the story. Similarly to the process of filming the actors, each location of the narrative also must be recorded from eight angles with intervals of 45°, forming a circle around the stage (Figure 5.13). The radius of the circle depends on the focal length of the camera. The longer the focal length of the cameras, the larger the radius of the

circle must be to cover the entire stage on the frame of all the cameras. In cases of small locations (e.g. corridors, small rooms) or to reduce the production work, some angles can be omitted. In these cases the dramatization system will automatically avoid showing the scenes from the missing angles.

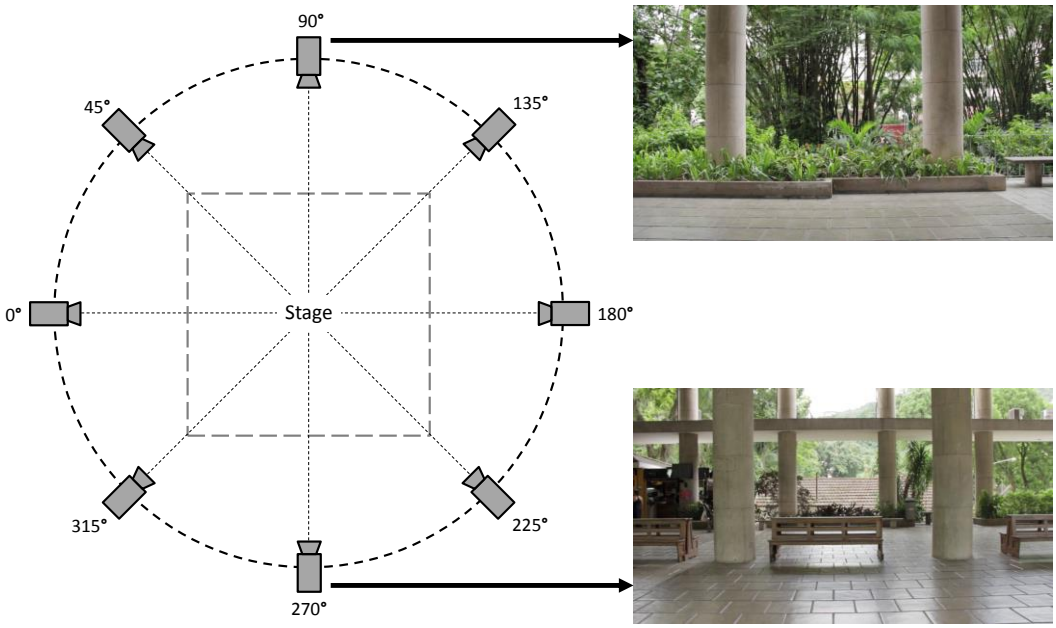


Figure 5.13: Camera placement for filming locations.

The locations can comprise one or more layers depending on the existence of objects that belong to the location and must overlap the characters acting in the scene (e.g. tables, pillars, doors). Although the layers are defined and created during the post-production phase, the filming crew must always imagine the objects that would be part of additional layers in order to film the location appropriately. Figure 5.14 shows an example of location composed of two layers (L_1 and L_2), where L_2 represents a table that overlaps the background L_1 and the characters acting in the scene.

Although it is possible to create layers with dynamic moving objects using videos, it is recommended to use photos with only static objects in order to simplify the work in the post-production phase. The main problem is the difficulty of separating the foreground elements from the background. While shooting the actors, the green screen was used to simplify this task. However, in this case there is no green screen so the editors have to separate the foreground layers manually frame by frame. An alternative to film locations where dynamic foreground layers

are really necessary is the placement of a green screen in the back of the foreground elements. In this case, two videos must be recorded, one without the green screen and other with the green background, keeping exactly the same camera position and angle in both videos.

Figure 5.14: Location with 2 layers (L_1 and L_2).

5.2.4. Shooting Static Scenes

Scenes that include complex interactions between characters or other objects that may not be adequately composed by the system in real-time can be represented as static scenes. This type of scene consists of a prerecorded video of the entire scene, including characters, film set, camera movements and different shots of the actions as a traditional film. When an event that is represented by a static scene is generated by the planning algorithms, the dramatization system will exhibit the prerecorded video instead of compositing the event in real-time.

The process of shooting static scenes is based on the master scene filming method. This method involves the filming of the entire scene with the master shot (a shot that includes the whole setting) along with coverage (shots that reveal different aspects of the action and use only view angles that are different from the master shot). In this way, the dramatization system always has two shots of each scene. If the agent detects a problem in the coverage shots, a new shot of the same action can be extracted from the master scene (Figure 5.15). The main motivation under the use of the master scene method is to make sure that there are no mismatches of continuity and no gaps between the actions. Furthermore, this

method gives to the editor the freedom to creatively cut and alter the pacing, the emphasis, and even the point of view of the scenes. Filming a scene with the master scene method can be done using a single camera or multiple cameras (Mascelli 1965). According to Brown (2011), the master scene method is used in probably 95% of narrative films shots today.

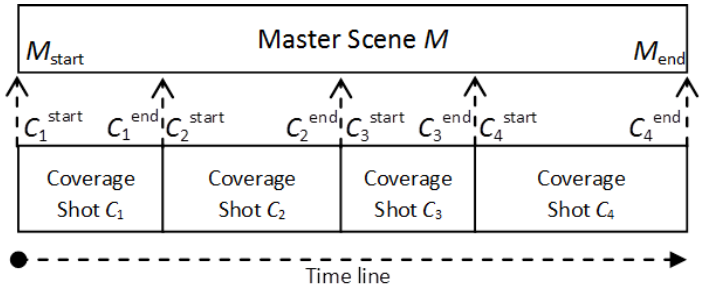


Figure 5.15: The master scene structure.

The choice of which scenes will be represented using static scenes is a decision of the film director. The proposed video-based dramatization system supports both dynamic composed scenes and static video sequences. An interactive film may be entirely represented using static scenes, however, as previously mentioned, it may face problems related to the lack of interactivity, story diversity and high production costs.

5.3. Post-Production

The post-production phase involves the process of editing the raw material captured during the production phase, including the process of removing the green screen background of the videos, separating the actors' actions and locations into individual video files, defining the structure of the virtual locations and associating the video files with the corresponding actors and locations. The next sub-sections describe these steps.

5.3.1. Editing Actions

The first step of the post-production phase comprises the process of editing the videos of the actors' actions filmed during the production phase. It involves

the task of removing the background of the videos to create alpha masks, the process of creating looping sequences when required, and the task of exporting the videos to be used by the system.

As a traditional video production, the first step of the editing process consists in extracting the content from the raw material, removing unnecessary parts and selecting the best shots when more than one shot of the same action was taken. This process can be done using any video editing software (e.g. Adobe Premiere, VirtualDub, Final Cut). In this step is also important to organize the video material, separating the videos of each actor and action in separated files and folders.

Once the material has been edited and organized, the next step of the process consists in removing the background of the videos to create alpha masks. The alpha mask is a video that encodes the clipping region that separates the actor from the background in the original video. Each frame of the alpha mask is a grey scale image in which black represents fully transparent pixels, white represents fully opaque pixels, and grey pixels represent a corresponding level of opacity. Figure 5.16 shows an example of alpha mask generated for its corresponding video source.

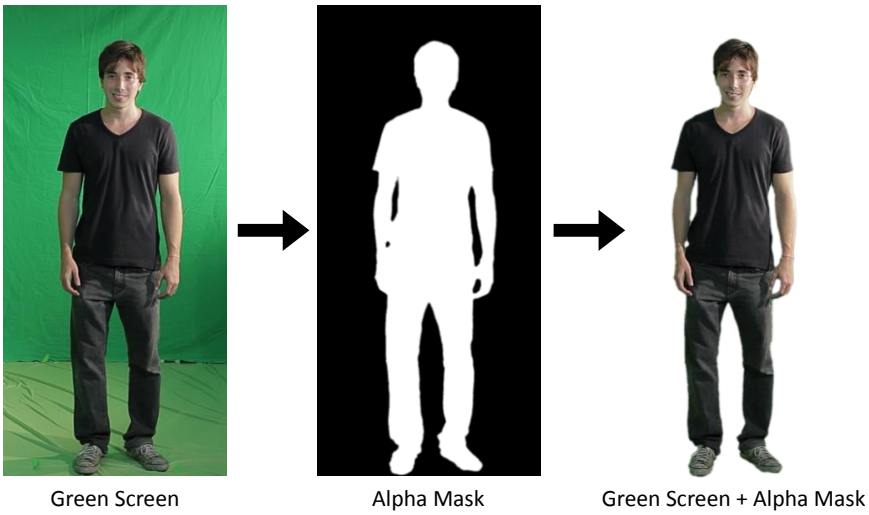


Figure 5.16: Example of alpha mask extracted from the green screen video.

The alpha masks are created using the chroma key matting technique. The most common video editing applications, like Adobe After Effects, Final Cut Pro and Pinnacle Studio include some tools that uses chroma key algorithms to

generate alpha masks. When the foreground object is quite solid with simple and sharp boundaries and is fully opaque, alpha mask can be easily extracted by the algorithms based on the estimation of the background color distribution. However, in some cases, objects may have intricate boundaries, such as hair strands and fluffy toys, or have semi-transparent parts, such as glasses and transparent clothes. In these cases, the extracted object may suffer from "color-spill", or some parts along the boundaries of the object might be cut out. In this way, manual adjustments are required in some situations to improve the results produced by the chroma key algorithms. In our experiments, the Adobe After Effects was used to create the alpha masks (Figure 5.17).

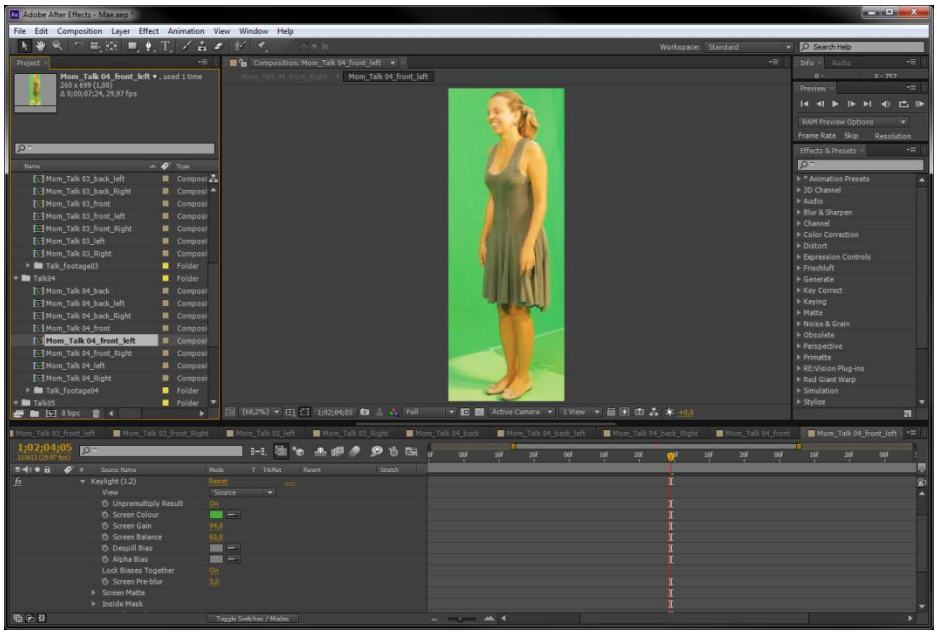


Figure 5.17: Adobe After Effects user interface.

Once the alpha masks have been created, the next step of the post-production phase involves the creation of looping sequences for the actions that may be played in loop during the dramatization of the narrative. This task could be manually done by observing and searching for similar video frames to march and connect the last frame of the video with the first frame. However, it would be a very time consuming task and, as it is not common task on the film industry, there is no available tools to assist the human editor. In order to simplify this process, an application was developed to automatically detect looping sequences (Figure 5.18).

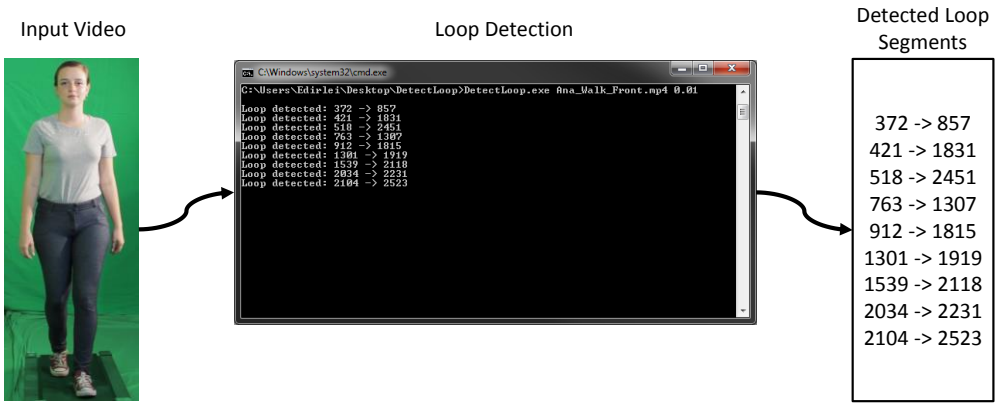


Figure 5.18: The loop detector tool.

The loop detector application receives as input the video segment to be processed and a similarity error factor α. The algorithm searches for loop segments by comparing all the frames of the video to determine the degree of similarity between them using the correlation coefficient metric:

$$F_{coor}(F_x, F_y) = \frac{\sum_i (F_x^h(i) - \overline{F_x^h})(F_y^h(i) - \overline{F_y^h})}{\sqrt{\sum_i (F_x^h(i) - \overline{F_x^h})^2 \sum_i (F_y^h(i) - \overline{F_y^h})^2}}$$

where $F_x^h(i)$ and $F_y^h(i)$ are the histogram values in the discrete interval i and $\overline{F_x^h}$ and $\overline{F_y^h}$ are the histogram bin averages. High values of correlation represent a good match between the frames, that is: $F_{coor}(F_x, F_y) = 1$ represent a perfect match and $F_{coor}(F_x, F_y) = -1$ a maximal mismatch. A frame interval is considered a loop segment only if $F_{coor}(F_x, F_y) > 1 - \alpha$.

The output of the loop detector application is a list of frame intervals of all loop segments detected in the input video. The human editor can then use this list to compare the segments and select the best loop sequence. Usually, the smaller sequences are the best options.

Once the alpha masks and looping sequences have been created, the final step is the process of exporting the videos. Only the frame region occupied by the actors in the videos must be exported. This is very important because the dramatization system uses the height of the video to estimate the real height of the actor (in pixels). The alpha masks and the videos must be exported to separated files and both must be encoded using the H.264/MPEG-4 Part 10 video compression format (ITU-T 2013), which is currently one of the most commonly used formats for the recording, compression, and distribution of video content.

Figure 5.19 illustrates the results of the action editing phase. For each video of an action filmed from a specific angle, two new video files are created: one containing the clipped region of the actor and the other with the corresponding alpha mask that will be used in the compositing process.

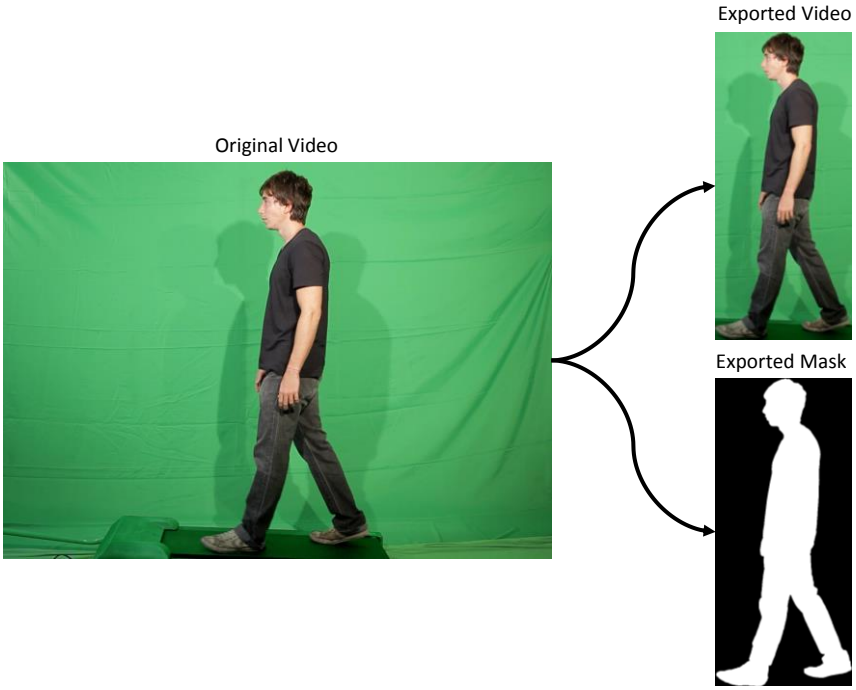


Figure 5.19: Example of results of the action editing phase.

Once all videos have been exported, they must be associated with their respective actors and actions. This information is defined in as XML file, which is presented in Section 5.3.3.

5.3.2. Editing Locations

The second step of the post-production phase involves the process of editing the videos or image layers of the locations where the events of the narrative can happen and creating the basic geometrical structure of the environment by defining the waypoints of each location.

Layers are only necessary when the location contains objects that belong to the environment and must overlap the characters acting in the scene (e.g. tables, pillars, doors). The task of creating layers is similar to the process of separating the characters from the background, but usually in this case there is no green screen background and the objects must be manually separated. However,

generally layers are only required in interior locations or they are composed of only static objects, which allows the layer to be composed of a single frame, simplifying the editing task. Once the layer elements have been separated from the background, an alpha mask must be created to define the clipping region of the layer. Similarly to the actors' actions, each location layer is composed of a video or image representing the visual elements of the layer and its respective alpha mask defining the clipping region (Figure 5.20). Locations composed of a single layer do not require alpha masks.

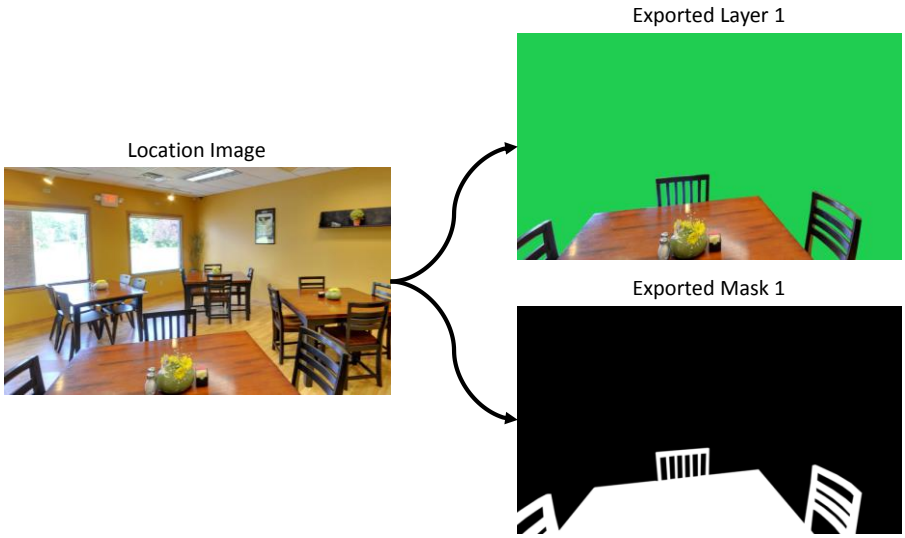


Figure 5.20: Location with a foreground layer defined by an image and its respective alpha mask.

Once the location layers have been defined, the next step consists in defining waypoints in the locations. The waypoints are used to define the basic geometrical structure of the locations, indicating where characters can be placed during the compositing process. This strategy is similar to the one used in games for navigation purposes, which simplifies the execution of path finding algorithms (Millington and Funge 2009).

In the proposed system, each location has a set of waypoints structured in the form of an undirected graph, where the vertices represent the waypoints and the edges represent the connections between the waypoints. There are three types of waypoints: (1) entrance/exit waypoints, which are used as a point of entrance/exit to characters entering/leaving the scene; (2) acting waypoints, which

are used as a point of reference to place characters that are performing some actions in the scene; and (3) connection waypoints, which are used to create paths between the other waypoints. Figure 5.21 shows an example of location that contains a graph connecting five waypoints (W_1, W_2, W_3, W_4 and W_5), of which W_4 and W_5 are entrance waypoints, and W_1, W_2 and W_3 are acting waypoints.

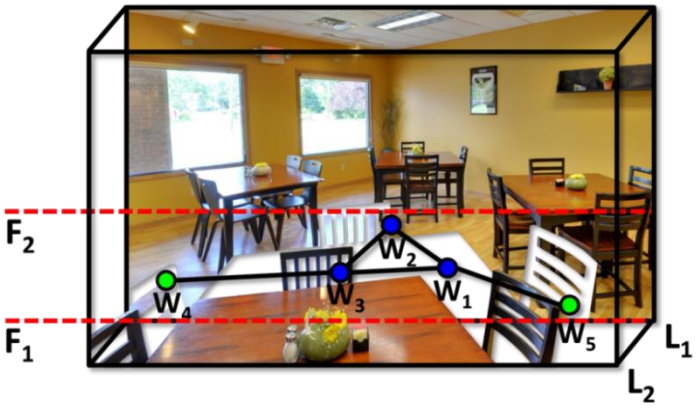


Figure 5.21: Location with 5 waypoints (W_1, W_2, W_3, W_4 and W_5). The front line F_1 and the far line F_2 delimit the region for waypoints.

Each waypoint gives information about its specific position in the location and the orientation that a character occupying that position must assume. The locations also contains the definition of a front line and a far line (F_1 and F_2 on Figure 5.21), which delimits the region where characters can be placed during the compositing process. Both far and front lines include the definition of the relative size that characters must have when they are placed over the lines. This information is used by compositing algorithm to estimate the size that characters must have when they are placed at any position of the scene.

The process of manually defining and annotating the positions of all waypoints may be a very time consuming task considering that all waypoints must be properly placed in the eight camera angles of each location. In order to simplify this process, we developed an interactive tool for waypoint placement (Figure 5.22). The application allows users to place and adjust waypoints in the eight camera angles simultaneously by displaying the location in a set of windows that reflect the circular structure of the available angles of the location. The tool also allows users to define the type, angle and connections between the waypoints. In

addition, the software also includes some facilities to assist users in defining the front and far lines for the locations.

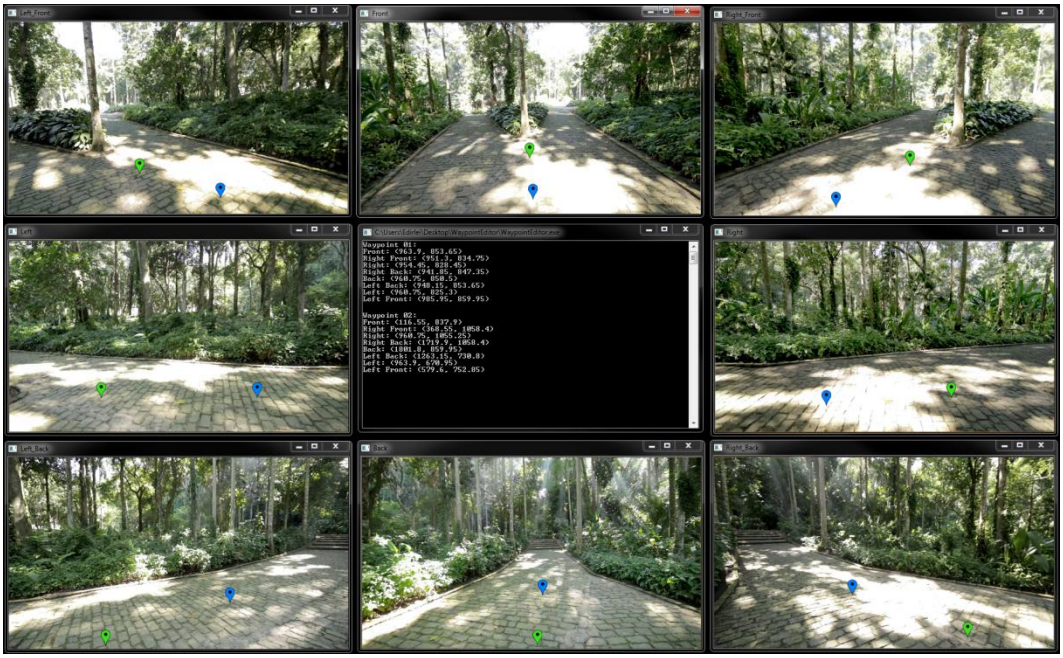


Figure 5.22: The interactive tool for waypoint placement.

Once the layers and waypoints have been established, they must be associated with their respective locations. This information is defined in an XML file, which is presented in Section 5.3.4.

5.3.3. Actor Definition

Actors are defined in an XML file, which associates the video files of actions with the logical definition of an actor entity. The XML file is composed of three main elements:

- **Actor**: represents an actor entity, which is identified by its attribute `name`. Each `Actor` contains a `Location`, defining the initial location of the actor, and a list of `Behaviors` representing the actions the actor can perform;
- **Behavior**: represents an action that an actor can perform and is identified by its attribute `name`. Each `Behavior` contains a set of videos representing the action from different camera angles;

- **Video**: represents a video of an action filmed from a specific camera angle identified by its attribute `type`. Each `Video` contains a `VideoFile` indicating the video of the action and a `MaskFile` indicating the respective alpha mask of the action.

The structure of the XML file that describes the actors of the narrative is illustrated in Figure 5.23. Each character of the story is defined by an `Actor` node, which is identified by the name of the character and contains a definition of the initial location of the character (`Location`) and the actions that can be performed by the actor (`Behaviors`). Each behavior is defined by a name and it is composed of a set of `Video` nodes that represent the action from different angles. Each video is identified by the name of the angle and contains two child nodes that indicate the path to video file of the behavior (`VideoFile`) and the respective path to the alpha mask video file (`MaskFile`). The behaviors can also include some additional parameters, such as the speed of a walk behavior or external audio files to be executed while the behavior is being performed by the actor.

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<ActorDatabase name="Video-Based Interactive Narrative">
  <Actor name="ActorName">
    <Location>ActorLocation</Location>
    <Behaviors>
      <Behavior name="BehaviorName">
        <Video type="AngleName">
          <VideoFile>Data\Behavior_Video.mp4</VideoFile>
          <MaskFile>Data\Behavior_Video_Mask.mp4</MaskFile>
        </Video>
        .
        .
        .
        <Parameter1>0.2</Parameter1>
        .
        .
        .
      </Behavior>
      .
      .
      .
    </Behaviors>
  </Actor>
  .
  .
  .
</ActorDatabase>
```

Figure 5.23: Structure of the XML file that describes the actors of the narrative.

An example of an XML file defining the characters of an interactive narrative can be found in (Lima and Feijó 2014).

5.3.4. Location Definition

Locations are also defined in an XML file, which associates the image or video layers of locations with the logical definition of a location. The XML file is composed of three main elements:

- **Location**: represents a location, which is identified by its attribute `name`. Each `Location` contains a set of videos or image layers representing the location from different camera angles;
- **Video**: represents the location filmed from a specific camera angle identified by its attribute `type`. Each video contains a list of `Layers` representing the layers of the location, a list of `Waypoints` indicating the available waypoints, and a definition of a `HorizonLine` and a `FrontLine`, which indicate the far and front lines of the location;
- **Layer**: represents an image or video layer of the location ordered according to its `zIndex`. Each `Layer` contains a `LayerFile` indicating the image or video of the layer and a `LayerMask` indicating the respective alpha mask of the layer;
- **Waypoint**: represents the structure of a waypoint identified by its attribute `name`. Each `Waypoint` is defined by a `type`, position (`x` and `y`), `angle`, `zIndex` and contains a set of `Connections` indicating connected waypoints.

The structure of the XML file that describes the locations of the narrative is illustrated in Figure 5.24. Each location of the story is defined by a `Location` node, which is identified by the name of the location and contains a set of `Video` nodes that represent the physical location from different angles. Each `Video` is identified by the name of the angle and contains a set of `Layer` nodes representing the image or video layers that compose the location. Each `Layer` has two child nodes that indicate the path to video or image file of the layer (`LayerFile`) and the respective path to the alpha mask (`LayerMask`). The `Video` nodes also contain the

definition of the far and front lines delimiting the region where characters can act during the dramatization (`HorizonLine` and `FrontLine`), and a set of `Waypoints` indicating the exact positions where they can be placed. Each `Waypoint` is defined by a `name`, a `type`, its position (`x` and `y`), the `angle` that characters must assume when occupying its position, and an index of its rendering order (`zIndex`). Each `Waypoint` also contains a set of child nodes indicating the waypoints that are connected to the current waypoint (`Connection`).

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<LocationDatabase name="Video-Based Storytelling">
  <Location name="LocationName">
    <Video type="AngleName">
      <Layers>
        <Layer zIndex="0">
          <LayerFile>Data\Layer_Video.mp4</LayerFile>
          <LayerMask>Data\Layer_Video_Mask.mp4</LayerMask>
        </Layer>
        .
        .
        .
      </Layers>
      <HorizonLine position="0" size="0"/>
      <FrontLine position="0" size="0"/>
      <Waypoints>
        <Waypoint name="WaypointName" type="WaypointType" x="0"
                                      y="0" zIndex="0" angle="0">
          <Connection>ConnectedWaypointName</Connection>
          .
          .
          .
        </Waypoint>
        .
        .
        .
      </Waypoints>
    </Video>
    .
    .
    .
  </Location>
  .
  .
  .
</LocationDatabase>
```

Figure 5.24: Structure of the XML file that describes the locations of the narrative.

An example of an XML file defining the locations of an interactive narrative can be found in (Lima and Feijó 2014).

5.3.5. Static Scenes Definition

Static videos that represent prerecorded scenes ready for presentation are also defined in an XML file, which associates the static coverage video of the action and master scene video with its respective logical story event.

The structure of the XML file that describes the static scenes of the narrative is illustrated in Figure 5.25. Each static scene is defined by a `Scene` node, which is identified by the logical sentence that describes the story event (`event`) and a boolean property identifying loop scenes (`loop`). Each `Scene` has two child nodes that indicate the path to video file of the coverage video (`VideoFile`) and the respective path to the master scene video (`MasterSceneFile`).

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<StaticSceneDatabase name="Video-Based Storytelling">
  <Scene event="EventDescription" loop="false">
    <VideoFile>Data\Static_Video.mp4"</VideoFile>
    <MasterSceneFile>Data\Static_Master.mp4"</MasterSceneFile>
  </Scene>
  .
  .
  .
</StaticSceneDatabase>
```

Figure 5.25: Structure of the XML file that describes the static scenes of the narrative.

An example of an XML file defining the static scenes of an interactive narrative can be found in (Lima and Feijó 2014).

5.3.6. Narrative Resource Pack

Video-based interactive narratives are composed of several video files, which are indexed and logically associated with characters, locations and events of the story through XML configuration files. Both video and XML files are stored in a narrative resource pack, which consists of a single compressed file that contains all the resources used by the dramatization system to represent the interactive narrative.

In order to create a narrative resource pack, all the resources of the narrative must be compressed using standard .ZIP file format. This file must contain the XML files identifying actors, locations and static scenes of the narrative. These files must be named according to the following conventions:

- `Actors.xml` – XML file that contains the description of the actors of the narrative;
- `Locations.xml` - XML file with the definition of the locations of the narrative;
- `Static.xml` – XML file that contains the description of the static scenes of the narrative.

Video files can be organized in any structure of directories inside of the resource pack as long as their file paths were properly defined in the XML files.

The narrative resource pack file is placed at the story server and it is automatically accessed by story dramatization clients. If a client does not have the resource pack of the requested narrative, the pack will be automatically downloaded and extracted by the client and will be used for the dramatization of the narrative.

5.4. Conclusion

This chapter presented the proposed process for the production of video-based interactive narratives, describing how to write and film an interactive story. In addition, some computational tools to assist the author during this process were described.

The specification of the story context undoubtedly requires some knowledge of programming and planning, which would limit the process of writing new stories to programmers. However, we believe that story specification should be a cooperative work involving both programmers and traditional story writers. While the author should be responsible for the creative process of having ideas for the story, the programmers should be responsible for codifying these ideas as a logical story context. In addition, we believe that the cooperation between story writers, programmers and story generation algorithms can positively affect the creative

process of the author by instigating a co-creation process, where the authors may embrace the output of the story generator algorithms as a contribution to the space of possible stories, changing their initial authorial intent and accepting it as a fundamental part in shaping and constraining the story space.

The production and post-production phases involve several filmmaking professionals, such as producers, actors, directors, camera operators and editors, which are in charge of constructing the film set, acting, shooting and editing the necessary actions, locations, and static scenes. Undoubtedly, these tasks require a considerable amount of work, especially for the people involved in the post-production phase, where all videos have to be edited and prepared to be used during the dramatization. Actors also have to adapt themselves to the production process, which require them to always act alone in front of a green screen performing generic actions without a predefined context, which differs from the way they are used to performing in traditional film productions.

6
Video-Based Dramatization System

This chapter describes the technical details about the implementation of the proposed video-based dramatization system.

6.1.
Methods and Libraries

The video-based dramatization system was implemented in C++ with some algorithms running on GPU (Graphics Processing Unit). The system is based on several video processing and artificial intelligence algorithms that were implemented using some open source libraries. The next sub-sections describe the methods and libraries used in the implementation of the proposed video-based dramatization system.

6.1.1.
Image and Video Processing

The main task of the video-based dramatization system is to compose and generate video sequences representing story events in real-time. The process of compositing a scene using video segments from different sources requires fast and optimized video and image processing algorithms capable of assembling multiple visual elements into a single piece of motion picture in real-time. In order to implement such algorithms, we adopted the OpenCV (Open Source Computer Vision Library),[5] which is an open source computer vision and machine learning software library written in C/C++ and designed for computational efficiency with a strong focus on real-time applications (Bradski and Kaehler 2008).

The OpenCV has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms (OpenCV, 2014). It contains a mix of low-level

[5] OpenCV - http://www.opencv.org

image-processing functions and high-level algorithms such as face detection, pedestrian detection, feature matching, and tracking. One of the OpenCV's goals is to provide simple-to-use computer vision interfaces and allow researchers to easily build sophisticated applications. The OpenCV can be used for several computer vision tasks, such as detecting and recognizing faces and other objects, processing video frames, tracking camera movements, establishing markers for augmented reality, among several other simple and complex computer vision tasks.

The OpenCV library also includes a GPU module written in CUDA (Compute Unified Device Architecture) that provides GPU acceleration for the computer vision and image processing algorithms, allowing programmers to benefit from GPU acceleration without requiring background in GPU programming (Pulli et al. 2012). The GPU module covers a significant part of the library's functionality and is still in active development. The module is designed as host API extension, which provides the user an explicit control on how data are moved between CPU and GPU memory.

The core of the proposed video-based dramatization system was built using the OpenCV functions to handle video files, process and compose video frames, and present the resulting motion picture. In order to improve the performance, the system uses the GPU module of the library whenever possible.

6.1.2. Artificial Neural Networks

During the compositing process, the dramatization system must follow the principles defined by cinematography theory and make intelligent decisions to create attractive and engaging visual presentations for the stories. In order to represent the cinematography knowledge and to make the system capable of performing intelligent decisions, we use several artificial neural networks trained to automatically select the best camera shots, visual effects and sound tracks for the scenes in real-time.

The artificial neural networks were implemented in the proposed system using the library FANN (Fast Artificial Neural Network Library),[6] which is an

[6] FANN - http://leenissen.dk/fann/

open source library that implements multilayer artificial neural networks in C with support for both fully connected and sparsely connected networks (Nissen 2003).

A complete description of the theory about artificial neural networks is presented by Hassoun (1995), Mitchell (1997) and Russell and Norvig (2010).

6.1.3. Emotions and Relations Network

Expressing and evoking emotions is a key factor to engage the audience in a narrative. The cinematography theory describes several ways to emphasize the emotions of characters by using specific camera shots, camera movements, light and music. In this way, the emotional states of characters participating in the action are essential to apply most of the cinematography concepts in the dramatization of stories. However, the current version of Logtell does not provide enough emotional information during the generation of stories. In order to overcome this limitation and simulate the emotions of characters during dramatization, we included in our system a dynamic multi-character network (Figure 6.1), where nodes represent the emotional state of characters and bidirectional arcs define affection relationships in the social environment of the story.

Emotional Attributes

Name: Peter	Name: Anne	Name: Carol
Anger: 2	Anger: 3	Anger: -8
Disgust: -4	Disgust: 7	Disgust: 2
Fear: 3	Fear: -8	Fear: -4
Sadness: 7	Sadness: -2	Sadness: 0
Sorrow: -3	Sorrow: -9	Sorrow: -3
Surprise: -1	Surprise: 3	Surprise: -4

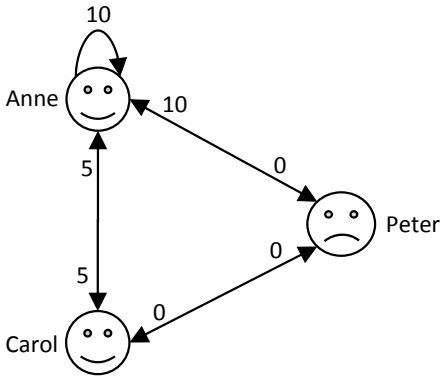


Figure 6.1: Emotions and Relations Network.

The emotional model adopted in the emotions and relations network is based on the six basic emotions proposed by Ekman and Friesen (1971), but we consider them lying on six emotion axes with negative and positive sides that represent opposite emotions:

- [calmness, anger];
- [liking, disgust];
- [confidence, fear];
- [joy, sadness];
- [cheeriness, sorrow];
- [anticipation, surprise].

The values in each axis are numbers within the interval [-10, +10]. The emotional state of a character *i* is given by intensity levels of its basic emotions $e_i^k(t) \in [-10, +10]$ and its affective relations $affection_{i,j}(t) \in [-10, +10]$ with the other characters. The sign (- or +) does not mean destructive or constructive emotions, but a connotation of drama impact and opposite states. The relations are directed and are not necessarily symmetric:

$$\exists\, i, j\ relation_{i,j}(t) \neq relation_{j,i}(t)$$

In this model, emotions can be combined to form a new emotion, for instance: love = joy + liking + confidence. Also, we can refer to extreme values on an axis as being special emotions, e.g.: grief = very high levels of sadness and ecstasy = very high levels of joy (that is, very low levels of sadness).

The network has fixed topology that is defined in an XML file by the author of the story. This definition includes the initial emotional state of all characters and their initial social relations. During dramatization, the network is updated when some event occurs. The update function can be described as:

$$\varphi(action, executor, victim)$$

where *action* indicates the action that occurred, *executor* represents the characters that performed the action, and *victim* indicates that characters that suffered the action.

Each action affects the emotions and relations in different ways. For example, considering the occurrence of a *kill* event, where a character CH_1 kills another character CH_2. In this case, the update function classifies the event into

one of three types of kill according to the current state of the characters and then updates the emotional states according to the selected type of action:

- **Intentional Kill:** occurs when the killer hates the victim $\left(affection_{CH_1,CH_2}(t) < -3\right)$. In this case, the emotional state of the killer will be updated by increasing his level of joy by +3 and confidence by +2.
- **Indifferent Kill:** occurs when the killer has a relation of indifference with the victim $\left(affection_{CH_1,CH_2}(t) > -3 \text{ and } affection_{CH_1,CH_2}(t) < +3\right)$. If the killer is a villain, his emotional state will be updated by increasing his level of joy by +1 and confidence by +1; otherwise, if the killer is a good person, his emotional state will be updated by decreasing his level of joy by -3, confidence by -2 and cheeriness by - 2.
- **Non-intentional Kill:** occurs when the killer likes $\left(affection_{CH_1,CH_2}(t) > +3\right)$ or loves $\left(affection_{CH_1,CH_2}(t) > +6\right)$ the victim. In this case, the emotional state of the killer will be updated by decreasing his level of joy by -4, confidence by -3, calmness by -3, liking by -2 and cheeriness by -2.

The constant values used to update the emotions and relations for each type action must be defined by the author of the story according to his/her authorial intent.

The emotions and relations network is capable of providing the basic emotional information required by the proposed video-based dramatization system, and its ability of simulating emotions and relations was already tested and validated in some of our previous works (Lima et al. 2010; Lima et al. 2011A).

6.2. Cinematography Agents

The proposed video-based dramatization system is composed of a set of cinematography-based autonomous agents that perform the same roles played by the corresponding filmmaking professionals. The process to create video-based interactive narratives is performed by the agents and is divided into two phases:

(1) scene definition, where the logical description of the scene is defined; and (2) scene compositing, where the video frames representing the scene are generated by the system. Figure 6.2 shows an overview of the video compositing process.

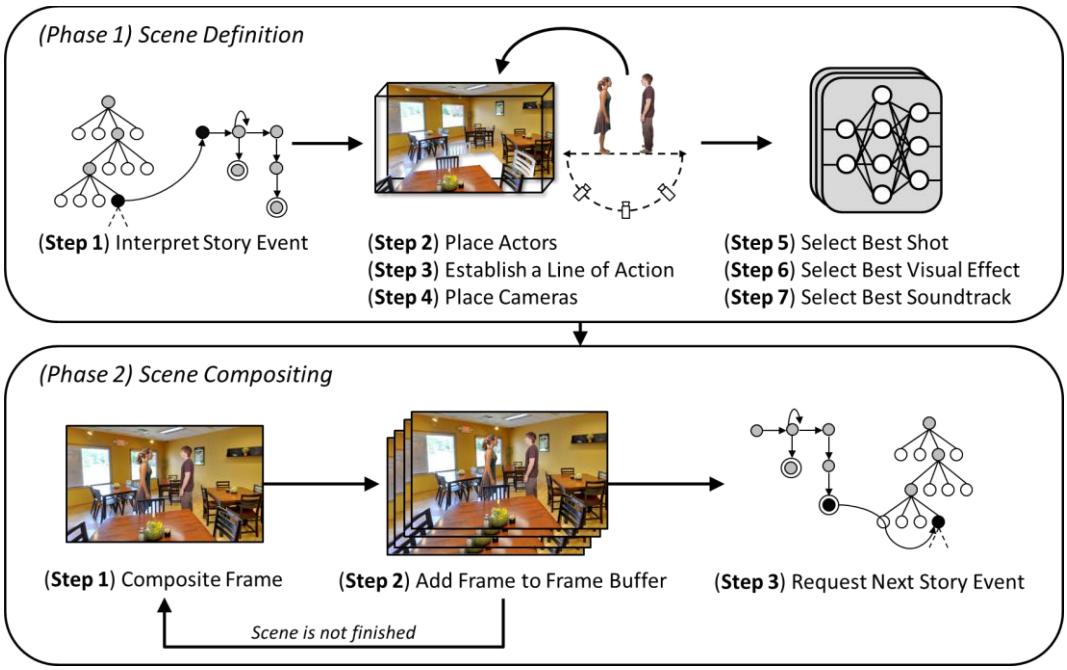


Figure 6.2: An overview of the video compositing process.

The tasks performed by the agents are divided in three processes: film directing, film compositing and film scoring. The next sub-sections describe these processes and the implementation of each cinematography-based agent.

6.2.1. Film Directing

In filmmaking, the director is responsible for creatively translating the written script into a visual form. He/she visualizes the script by giving to abstract concepts a concrete form, which helps him/her to determine the general structure of each scene of the narrative, including the position of actors and cameras. The director is responsible for the dramatic structure and the directional flow of the film (Mascelli 1965).

In the proposed video-based dramatization system, the agents Scriptwriter and Director share the responsibility of directing the dramatization of the interactive narratives. The directing process is divided into different steps, as

illustrated in the flowchart of Figure 6.3. The process starts when the Scriptwriter agent receives a new nondeterministic automaton containing the logical descriptions of a story event for dramatization. The first step of the directing process consists of parsing and interpreting the received automaton. Once the automaton has been parsed, the Director agent starts the dramatization of the first basic action described on the automaton. If execution of the basic action led to a final state in the automaton, a new automaton is requested to the story server; otherwise, if it led to a branching state, the system check's the users' suggestions to decide which path to follow in the automaton. Once the path has been chosen or when there was no branching point, the Director agent starts the dramatization of the next basic action.

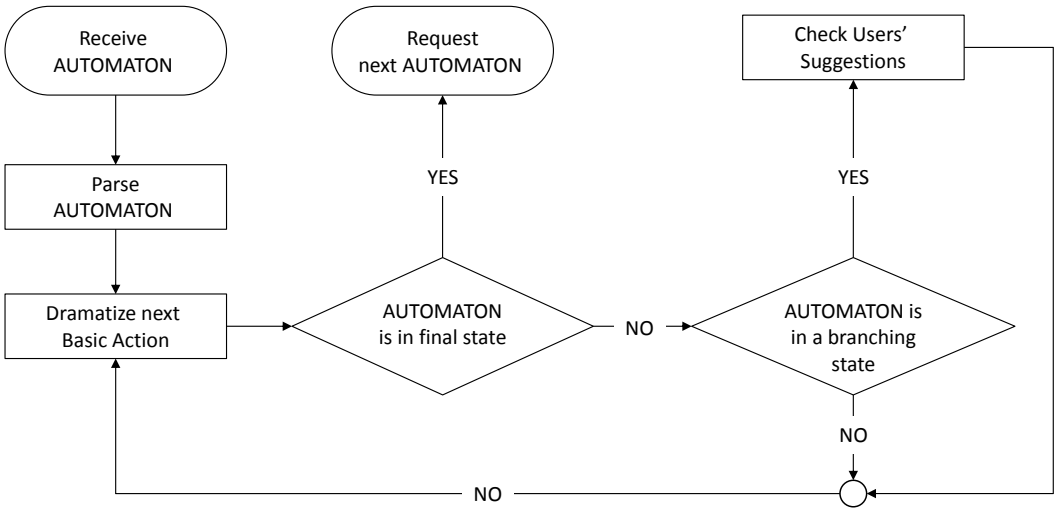


Figure 6.3: Flowchart of the directing process.

The next sub-sections describe in more details the implementation of all the steps of the directing process.

6.2.1.1. Scriptwriter

In filmmaking, the scriptwriter (or screenwriter) is responsible for creating a compelling and coherent story for the script of the film. Similarly, the Scriptwriter agent of the proposed video-based dramatization module is responsible for interpreting and managing the interactive story plots generated by the planning algorithms of the story generator module. The agent receives and interprets the

nondeterministic automata of the story events, requesting the next automaton after the successful execution of all actions present in the current automaton.

The automata interpreted by the agent are represented by adjacency list structures. They follow the same formalism introduced in Logtell by Doria et al. (2008), where states are described by invariants and transitions between states correspond to the basic actions that can be performed by the actors. States that have more than one adjacency are used as local decision points, where users can decide which action the actors should take. An example of automaton is described and illustrated in Section 5.1.1.4.

The communication between the Scriptwriter agent and the story generator module is done through a TCP/IP connection. The story generator module acts as a server and is always waiting for incoming connections on port 2563. The story dramatization module acts as a client that must connect to the IP address of the story generator module to request the automata of the story events. The communication protocol is based on the following rules:

- To request the first or the next automaton of an ongoing story, the client sends a message to the server in the format: `next#STATE_ID`, where `STATE_ID` indicates the ID of the final state reached during the execution of the current automaton. In the case of a new story, the state ID is -1.
- The server responds the client requests by sending a network message containing the next automaton of the ongoing story. The automaton is encoded in the following format:

```
plan#[AUTOMATON_ID, [STATE1_ID,
                       [STATE1_FACT1, ..., STATE1_FACTn],
                       [[BASIC_ACTION1, NEXT_STATE_ID], ...,
                       [BASIC_ACTIONn, NEXT_STATE_ID]]
                     ], ... ,
                     [STATEn_ID,
                       [STATEn_FACT1, ..., STATEn_FACTn],
                       [[BASIC_ACTION1, NEXT_STATE_ID], ...,
                       [BASIC_ACTIONn, NEXT_STATE_ID]]
                     ]]
```

which contains a list of states identified by an ID (STATE$_1$_ID, ..., STATE$_n$_ID). Each state is composed of a set of facts in the form of first order logic sentences describing the state (STATE$_n$_FACT$_1$, ..., STATE$_n$_FACT$_n$) and a set of basic actions ([[BASIC_ACTION$_1$, NEXT_STATE_ID], ..., [BASIC_ACTION$_n$, NEXT_STATE_ID]]) that also indicate the next state of the automaton (NEXT_STATE_ID) after the execution of the action.

The basic actions that compose the automata and can be performed by the actors are described in the form of ground first order logic sentences. For example, the action where a character *X* looks at another character or object *Y* in a location *W* is expressed by the sentence *LookAt*([*X*], [*Y*], [*W*]). The predicate indicates the action and the variable symbols indicate the characters, objects and locations related to the action. The symbols are expressed as lists of variables, which allows more than one character or object to be involved in the same action. For example, it is possible to express an event where two characters (X_1 and X_2) look at the same time to another character *Y* in a location *W* (*LookAt*([X_1, X_2], [*Y*], [*W*])). This is also very useful to represent dialog events, where a character can be speaking to several other characters. For example, the sentence *Tell*([*X*], [*Z*], [Y_1, Y_2, Y_3], [*W*]) indicates that the character *X* is speaking the utterance *Z* to the characters Y_1, Y_2 and Y_3 in the location *W*.

Once the Scriptwriter agent has received the network message containing the description of the automaton, the agent parses the message and creates an adjacency list structure representing the nondeterministic automaton. The structure is then sent to be executed by the Director agent.

6.2.1.2. Director

In filmmaking, the director is responsible for translating the script into a visual presentation. He/she controls the overall aspects of the film, including the content and flow of the narrative events, the performance of the actors, and the organization and selection of the locations in which the film will be shot. Similarly, the Director agent controls the overall flow of the dramatization by

interpreting and controlling the execution of the nondeterministic automaton of the story events, deciding which path to follow in the automata according to the users' choices. The agent is also responsible for assigning roles to the actors involved in the action and selecting the locations where the scenes will be shot.

The first task of the Director agent is to load the resources (actors and locations) used for dramatization. During the initialization of the system, the Director agent reads all information about actors, locations and static scenes from the XML files `Actors.xml`, `Locations.xml`, and `Static.xml`, which are included in the narrative resource package (whose the format were presented in Sections 5.3.3, 5.3.4, 5.3.5 and 5.3.6). The actors are instantiated and their respective behaviors and associated videos are properly loaded and stored in a hash table using the actor name as key. Similarly, the locations are also instantiated and their data (video/image layers and waypoints) are loaded and stored in another hash table using the name of the location as key. Static scenes are stored in another hash table using the static predicate as key. Storing the actors, locations and static scenes in hash tables provides an optimized and efficient way of accessing the resources during the compositing process, allowing a direct mapping between the variable symbols present in the first order logic sentences of the basic actions and the key/value pairs of the hash table.

The second task of the Director agent starts when a new automaton is received for dramatization. The execution of the nondeterministic automaton begins in the initial state and ends in a final state. In each transition between states, the corresponding basic action is dramatized. When a local decision point is reached (a state with more than one adjacency), the agent consults the users' choices to decide which action the actors should take (more details about local user interactions will be presented in Chapter 7).

In order to initiate the dramatization of a basic action, the Director agent creates a structure of a scene. This structure comprises a list of scene elements that compose the scene representing the basic action. There are three types of scene elements:

1. **Location**: represents the place where the action is happening and includes the video or image layers of 8 angles of the location together with their respective waypoints and encoded information;

2. **Main characters**: represent the characters participating in the action and include the videos and alpha masks of the actors performing their current actions;
3. **Supporting characters**: represent the characters that are not directly participating in the action, but are in the same place where the action is happening.

The process of creating the structure of a scene includes a simple verification to check whether the basic action is a static scene or it needs to be dynamically composed by the system. This is accomplished by consulting the hash table of static scenes using the predicate of the basic action as key. If the basic action is a static scene, the reference to the video of the scene is added to structure as a location and, during the composition process, its frames will be directly added to the frame buffer without additional processing costs; otherwise, if the scene needs to be dynamically composed, the scene elements are included in the structure according to the description of the basic action. For example, considering the action *LookAt*([X_1, X_2], [Y], [W]), the location W and the characters X_1, X_2 and Y will be included in the list of scene elements, and the current behavior of X_1 and X_2 will be set to the action *LookAt*. In addition, other characters that are not directly participating in the action, but are located in W are also included in the scene structure as supporting characters.

Once the scene structure has been created, it is sent to the Scene Composer agent, who will be in charge of compositing a video sequence to represent the basic action.

6.2.1.3. Actors

Actors are entities that represent the characters of the stories. They are composed of a set of behaviors representing the actions they can perform during dramatization. Although their actions are selected by the Director agent based on the scene that has to be dramatized, they have the freedom to represent the specified action autonomously.

The behaviors are the core of the Actors. Each behavior implements a specific action and is composed of a set of videos representing the actor performing the action from different angles. The behaviors can be simple, like an *Idle*, *Talk* or *ReadBook*, where the video of the action is simply played on the same position of the scene; or more complex, like a *Walk*, where the actor has to physically move across the scene while playing the pre-recorded video of the action.

The proposed system includes a comprehensive set of behaviors that can be used to create video-based interactive narratives. However, different story contexts may require the implementation of new behaviors for the actors. In order to simplify such implementations, the behaviors are coded in the system using a hierarchy of classes, where the base class provides the basic functionalities needed for the implementation of new behaviors without accessing the other components of the system.

Each behavior is implemented in a separate class that is inherited from the base class `BehaviorBase`, which contains the basic information and functions of a behavior and provides access to the video data representing the action. Each behavior implements a virtual method called `ProcessBehavior`, which is automatically executed in loop and can be used to implement the logic of the behavior. Figure 6.4 shows a simple template of a behavior class inherited from the class `BehaviorBase`.

```
class SimpleBehavior : public BehaviorBase
{
     public:
       SimpleBehavior(void);
       ~SimpleBehavior(void);

       void ProcessBehavior()
       {
             //Behavior logic loop
       }
};
```

Figure 6.4: Template of a behavior class inherited from the class BehaviorBase.

Simple actions, such as an *Idle* behavior (where the actor stands in the same position), do not require the implementation of a logic loop; however, more

dynamic behaviors, like a *Walk* action (where the actor has to move between waypoints), require the implementation of a logic loop to create the movement of the actor between waypoints.

The process of moving the actors between waypoints would be simple considering a scene from only one angle; however, it become more complex when we consider a dynamic angle that can change while the actors are performing the movement. If the actor is halfway between two waypoints when observed from one angle, he must be halfway when viewed from any other angle. However, the positions and even the distances between the waypoints are not the same when they observed from different angles in a 2D plane (Figure 6.5). In order to solve this problem, the position of the actors when moving between waypoints is calculated using a linear interpolation between the two waypoints. The position *x* and *y* an actor *A* when moving between two waypoints (WP_1 and WP_2) is given by:

$$A_x(WP_1^x, WP_2^x, p) = WP_1^x * (1 - p) + WP_2^x * p$$
$$A_y\left(WP_1^y, WP_2^y, p\right) = WP_1^y * (1 - p) + WP_2^y * p$$

where $p \in [0, 1]$ represents the relative position of the actor between the waypoints (when $p = 0$, the actor is at WP_1; when $p = 1$, the actor is at WP_2). The value of p is gradually increased according to a constant π, which defines the speed of the actor and can be adjusted to match the speed of the video of the actor walking with the speed of the physical movement.

(*a*) 90° (*b*) 0°

Figure 6.5: Scene of a character walking from W_2 to W_1. Image (*a*) shows the scene viewed from 90° and image (*b*) shows the same instant (p = 0.5) viewed from a virtual camera placed at 0°.

The structure of waypoints that establishes the locations where characters can be placed in the scene is represented as a graph. Sometimes the actors may have to walk through several waypoints to reach their destination. Consequently, the logical loop of the walk behavior also implements an A* search algorithm (Russell and Norvig 2010) in order to find the best paths the actors have to follow to reach their destination.

6.2.2. Film Compositing

The compositing process consists in assembling the visual elements that compose the scenes into a single piece of video. The goal is to create the illusion that all elements always existed in the same location. In real filmmaking, this process usually is manually done by several special effects professionals that work for days in short video segments to create realistic scenes. In video-based interactive storytelling, however, this process must be done in real-time and without human intervention.

As a traditional film, a video-based interactive narrative must have a cinematic look and be composed of a variety of different shots, camera movements and lighting effects. In order to create such cinematic interactive narratives, actors and settings are both shot from 8 different angles with intervals of 45° during the production process. In this way, the system has the freedom to compose scenes from different angles, simulate camera movements and create more dynamic video sequences that cover all the important aspects of the cinematography theory. However, handling such tasks without human intervention requires the development of fast and intelligent algorithms to apply the cinematography techniques to create attractive and engaging visual representations for the story events in real-time.

In the proposed video-based dramatization system, the agents Editor, Cameraman, Scene Composer and Director of Photography share the responsibility of compositing the scenes according to the information provided by the Director agent. The compositing process is divided into several steps, as illustrated in the flowchart of Figure 6.6.

The compositing process starts when the Scene Composer agent receives a new scene structure for dramatization. The first step of the compositing process consists in defining the basic setup for the scene by placing the actors that are participating in the action on the available waypoints of the location. Then, the Cameraman agent establishes the line of action and places the possible cameras to shot the scene according to the spatial information of the scene setup. Based on the available cameras, the Editor agent selects the best angle and type of shot to film the scene. Then, the agent verifies the occurrence of jump cuts. If a jump cut is detected, a new camera angle is selected; otherwise, the agent continues the process and selects the most adequate shot transition (cut, dissolve, wipe or fade). Before starting compositing the frames, the Director of Photography agent enters in the process and selects the best lighting and color effects to emphasize the emotional content of the scene. Finally, after defining all the visual aspects of the scene, starts the actual process of generating video frames representing the scene, which is the most time-consuming task. Once a frame has been generated, it is added to the frame buffer to be shown to viewers. After all frames have been successfully composed, the Scene Composer agent requests the next scene structure to be dramatized.

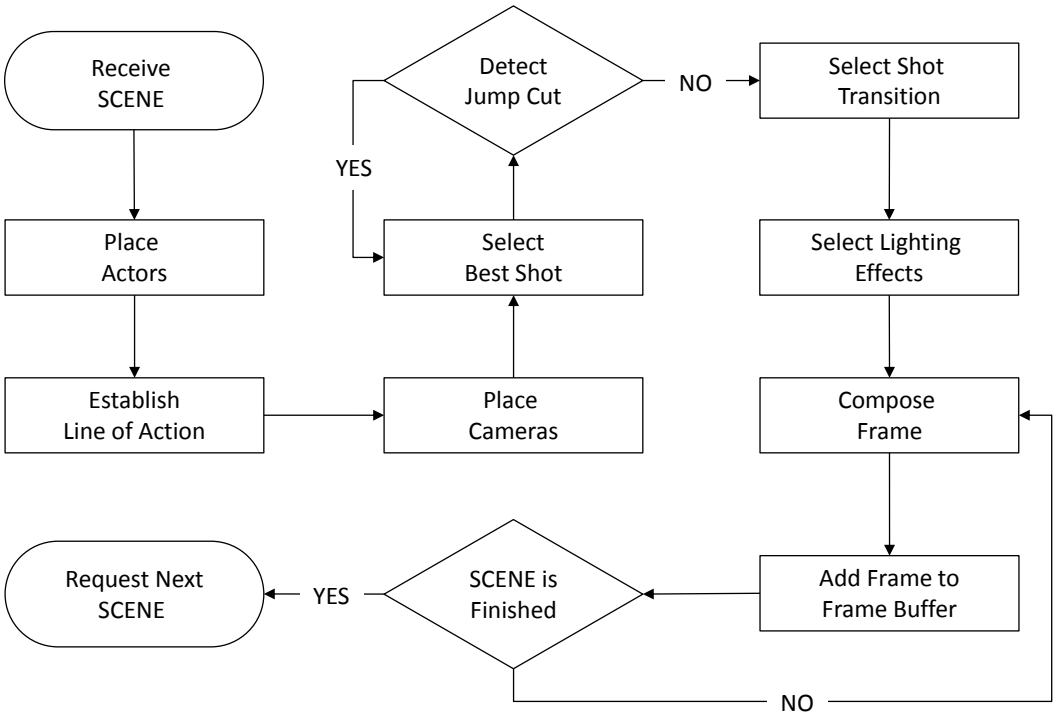


Figure 6.6: Flowchart of the compositing process.

The next sub-sections present more details about the implementation of all the steps of the compositing process.

6.2.2.1. Placing Actors and Establishing the Line of Action

In the first step of the compositing process, the basic configuration of the scene is logically defined by the Scene Composer agent. First, actors who are participating in the scene are placed on the available waypoints of the location where the scene is happening.

As previously mentioned in Section 5.3.5, there are three types of waypoints:

- **Entrance/Exit Waypoints**: used as starting or ending points to place characters when they are performing *GoIn* or *GoOut* actions, that is, when they are entering or leaving the scene location;
- **Acting Waypoints**: used to place characters when they are performing other actions in the scene;
- **Connection Waypoints**: used to connect and create paths between the other waypoints of the location. Characters only walk through these waypoints when they need to go to other waypoints.

The actors are placed on the scene according to the actions they are performing. If an actor is executing a *GoIn* action, he will be placed on the first entrance waypoint that is not occupied by another actor. If the actor is performing a *Talk* action, he will be placed in the first acting waypoint available. The position and angle of the actors are defined according to the information provided by the waypoints. However, the size of the actors must be automatically calculated by system.

As in the real world, the closer the actor is to the camera, the larger it must appear to be in relation to the rest of the scene. Accordingly, in our method the width A_w and height A_h of an actor A in a location L are given by:

$$A_w(A, L) = A_v^w \left(\frac{\alpha(A, L)}{100}\right)$$

$$A_h(A,L) = A_v^h\left(\frac{\alpha(A,L)}{100}\right)$$

where A_v^w and A_v^h represent the original size (width and height) of the video of actor *A*, and the function $\alpha(A, L)$ computes the relative size of the actor through a linear interpolation between the front line L_{F2}^{pos} and far line L_{F1}^{pos} according to his current position A_y and his relative size on the front and far lines (L_{F2}^{size} and L_{F1}^{size}):

$$\alpha(A,L) = L_{F1}^{size}\big(1-\gamma(A,L)\big) + L_{F2}^{size}\big(\gamma(A,L)\big)$$

where $\gamma(A, L)$ is a function that normalizes the current position A_y of the actor *A* in the interval of [0,1]:

$$\gamma(A,L) = \frac{A_y - L_{F1}^{pos}}{L_{F2}^{pos} - L_{F1}^{pos}}$$

Figure 6.7 shows an example of scene containing an actor placed in two different waypoints with his size calculated using the proposed method.

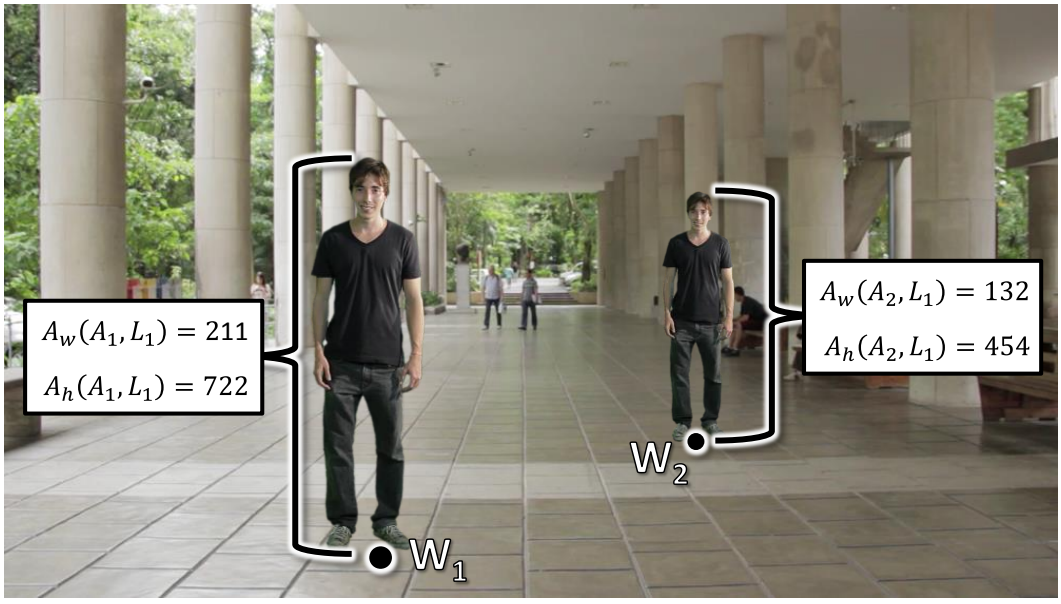


Figure 6.7: Example of scene using the proposed method to calculate the size of the actor.

Although the actors are initially placed over the waypoints, their position may change during the dramatization of the action. Thus, every time the position of an actor is modified, his relative size is recalculated and updated.

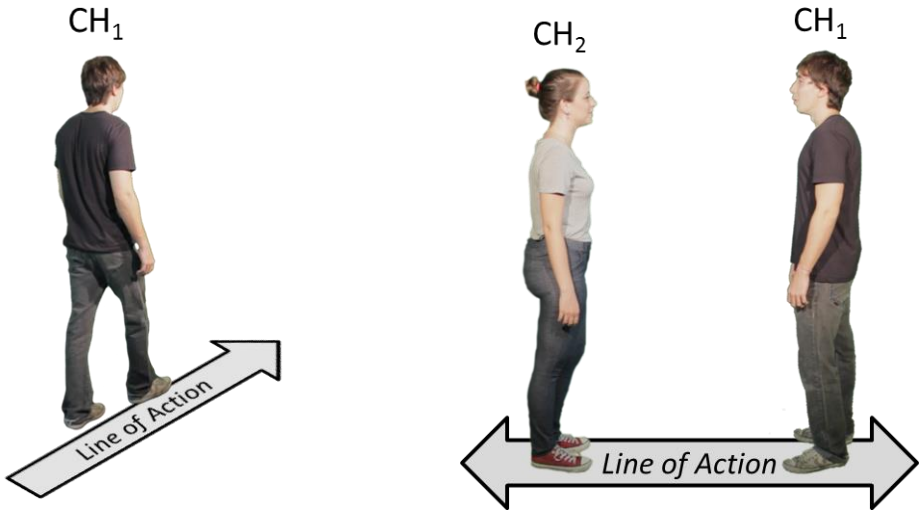
Once the position, angle and size of all actors involved in the action are properly defined, the next step of the process comprises the definition of a "line of action", which is used to maintain the spatial continuity of the scenes. As presented in Chapter 3, the line of action (or action axis) consists of an imaginary line connecting the most important elements or directing the focus of the action in a scene. When shooting a scene, the cameras must be placed only at one side of this line (180 degree rule). The placement of the camera in different positions and angles in the same scene must occur only within the 180 degree arc. When shooting two consecutive shots of the same subject from inside of the 180 degree arc, the camera angle for the new shot must be at least 30 degrees from the angle of the previous shot (30 degree rule). In this way, the two shots can be considered different enough to avoid jump cuts (Mascelli 1965), which is an undesirable effect that causes visual jumps in either space or time of the film. These rules help to maintain the visual continuity of consecutive shots, and keep the narrative moving forward logically and smoothly, without disruptions in space or time (Brown 2011).

In order to establish the virtual line of action in the scene, the Cameraman agent adopts some common guidelines presented by Hawkins (2005) and Thompson and Bowen (2009), which state that in a scene with a single character, the line of action usually is given by the initial direction of the character. In scenes involving more characters, it is established by a line connecting the two most important characters in the scene. In this way, the virtual line of action is defined based on the position and orientation of the characters participating in the action as illustrated in Figure 6.8.

6.2.2.2. Camera Placement and Definition

After defining the position and orientation of all actors and establishing the line of action, the next step of the compositing process comprises the definition of the virtual cameras that can be used to film the scene. According to Brown (2011),

camera placement is a key decision in storytelling. It determines what the audience sees and from what perspective they see it. Each shot requires placing the camera in the best position for viewing characters, setting and action at that particular moment in the narrative (Mascelli 1965). The approach employed to accomplish this task in the proposed system is based on the use of some standard arrangements for camera placement.



(*a*) (*b*)

Figure 6.8: Examples of line of action. Image (*a*) shows a line of action established for a scene of a character (CH_1) walking in a direction *d*. Image (*b*) shows the line of action for scene of a dialog between two characters CH_1 and CH_2.

The cinematography theory defines standard patterns for camera placement depending on the type of scene (Arijon 1976; Katz 1991; Kenworthy 2009). These patterns act as a guide for possible choices of initial camera placement, with the final configuration depending on the constraints of the scene (Hawkins, 2005). In a scene of a dialog between two characters, for example, it is common to use the pattern known as the triangle system, whereby all possible shots for any subject are taken from three points forming a triangle within the currently chosen side of the line of action (Figure 6.9).

The four configurations of the triangle system illustrated in Figure 6.9 can be combined to multiply the camera options. The complete triangle system offers 7 camera viewpoints contained within a triangular formation (Figure 6.10).

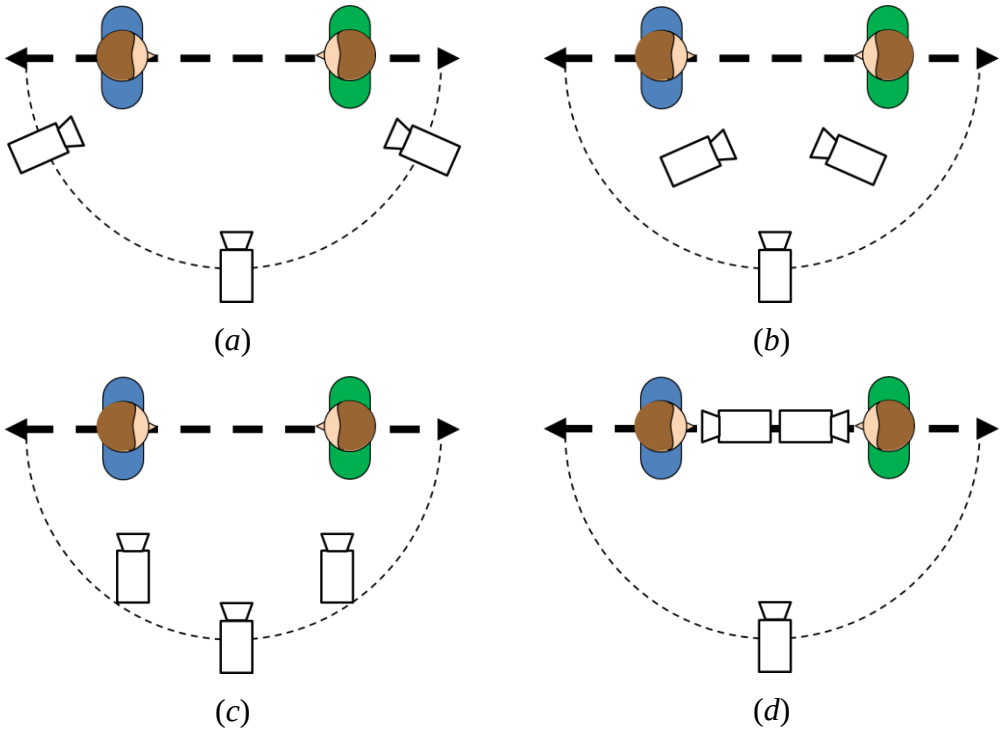


Figure 6.9: Triangle Systems. Image (*a*) shows the over-the-shoulder triangle system; image (*b*) shows the 45° triangle system; image (*c*) shows the profile triangle system; and image (*d*) shows the close-up triangle system.

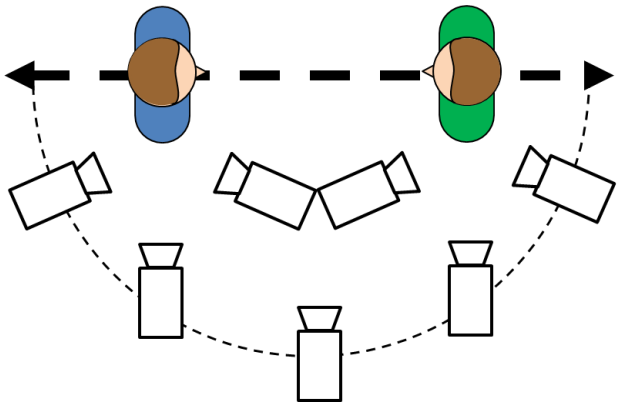


Figure 6.10: Full Triangle System.

Although the camera placement patterns provide the basic guidelines on how to place the cameras, not all angles can be produced using the pre-recorded videos of actors and locations filmed during the production process. However, the wide variety of angles and the high-definition video resolution of the material provide to the system the ability of simulating most of the possible cameras by manipulating the angle, type of shot and subjects. Figure 6.11 illustrates some of

the shots that can be simulated using the video material available in a scene of a dialog between two characters.

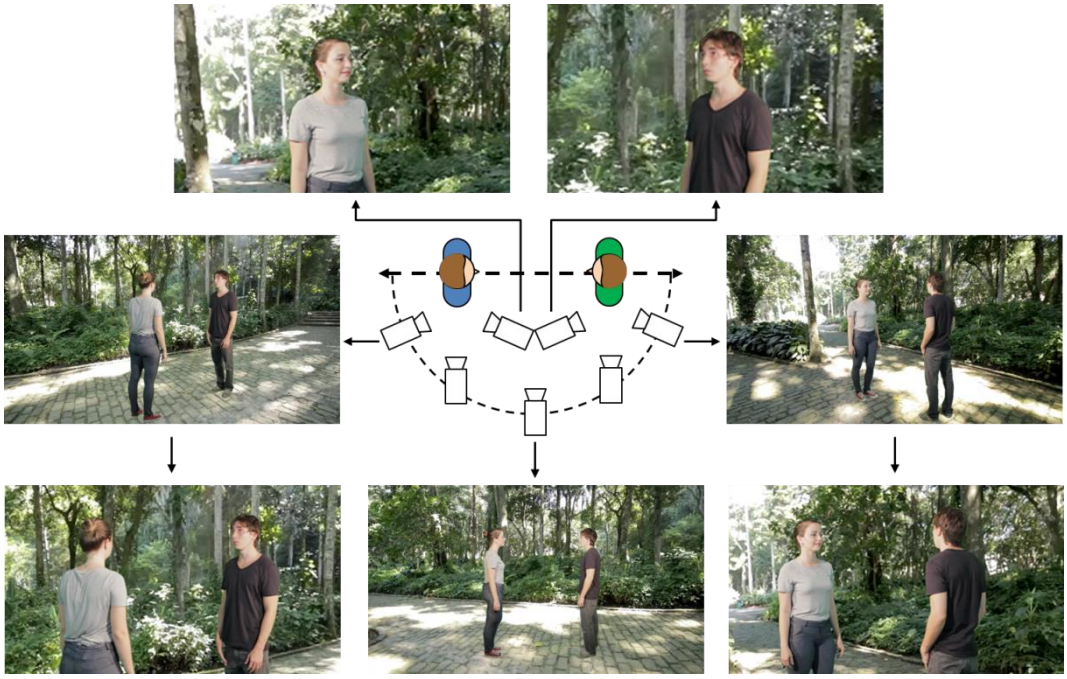


Figure 6.11: Examples of shots that can be simulated using the video material available in a scene of a dialog between two characters.

Each camera is defined by 4 parameters:

1. **Angle**: defines the angle and the position of the camera. There are 5 possible angles at one site of the line of action;
2. **Target Subjects**: defines the target subjects of the camera. The targets will be centralized in the shot;
3. **Shot Type**: determines the type of shot used by the camera. There are 5 possible types of shot: long shot, medium long shot, medium shot, medium close-up and close-up;
4. **Movement Type**: defines the type of movement executed by the camera. There are 4 types of camera movements:
    a. **Static**: no camera movements are executed. The camera remains static in the initial configuration;
    b. **Follow**: the camera follows the subjects and keeps them on the frame;

c. **ZoomIn**: the camera performs a zoom in movement, starting from the initial configuration and ending with the type of shot defined by a parameter;

d. **ZoomOut**: the camera performs a zoom out movement, starting from the initial configuration and ending with the type of shot defined by a parameter.

In order to simulate different types of shots using the pre-recorded videos, the system relies on the high-definition resolution of the video material. As explained in Chapter 5, during the production process the videos of actors and locations are both recorded using high-definition video resolution. Actors are filmed in a long shot (which shows their whole body), and locations are filmed in a very long shot (which includes the whole environment). The high quality of the videos permits the system to zoom in and zoom out during the compositing process, which allows the system to generate different types of shots and camera movements. As a consequence, the output video produced by the system is in a standard-definition resolution, which avoids the degradation of the video quality. If a higher output resolution is required, the video material has to be recorded in an even higher resolution. In the experiments conducted during the development of this thesis, the video material was recorded in full HD resolution (1080p), and output produced by the system is in SD resolution (480p).

The process of simulating the virtual camera during the compositing process is based on the concepts of "*World*", "*World Window*" and "*Viewport*", commonly used in computer graphics to specify coordinates systems (Hughes et al. 2013). As illustrated in Figure 6.12, the *Scene World* of the video-based dramatization is composed by the entire scene, including the video or image layers of the location and all the other elements that composed the scene. The *Camera Window* specifies the rectangular region of the video to be filmed, and is defined by the position ($shot_x$, $shot_y$) and size ($shot_w$, $shot_h$) of the shot, which are calculated by the system based on the camera parameters. The *Viewport* represents the rectangular region used to project and display the video filmed through the *Camera Window*.

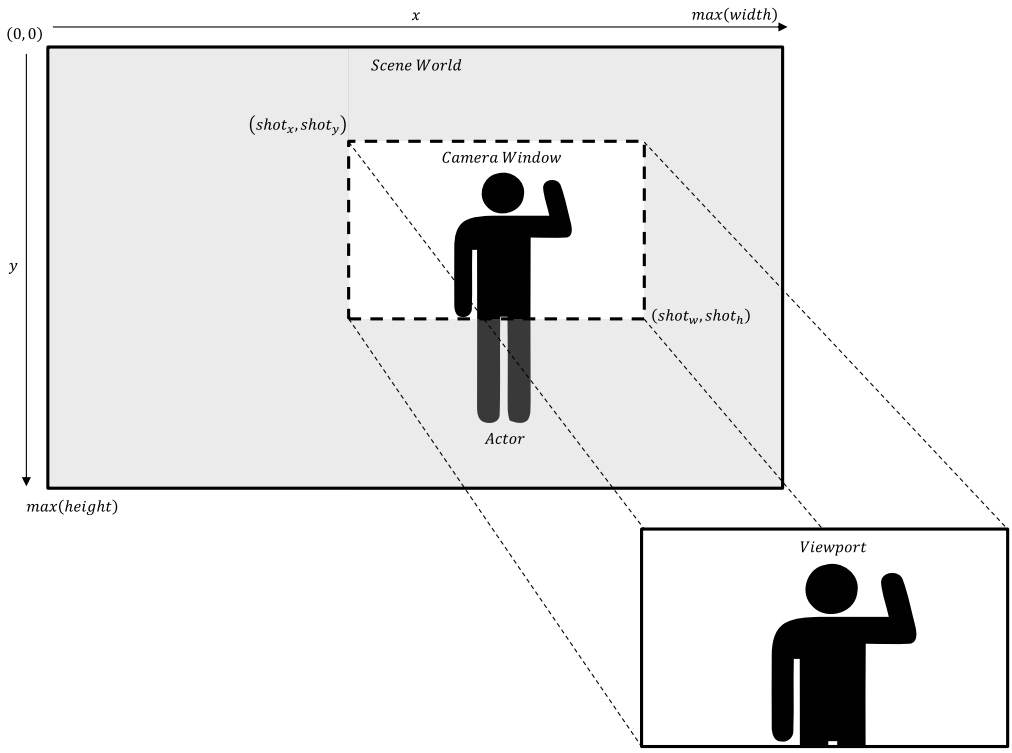


Figure 6.12: Window system.

The position ($shot_x$, $shot_y$) and size ($shot_w$, $shot_h$) of the *Camera Window* are calculated according to the type of shot and the target subjects (E) of the selected camera. The very long shot, which is the widest shot and include the whole scene, is given by:

$$shot_x(E) = 0$$
$$shot_y(E) = 0$$
$$shot_w(E) = max(width)$$
$$shot_h(E) = max(height)$$

The other types of shot are given by:

$$shot_x(E) = \varphi(E) - \frac{shot_w(E)}{2}$$
$$shot_y(E) = min(E_y) + \omega(E, \beta)$$
$$shot_w(E) = shot_h(E) * \frac{max(width)}{max(height)}$$
$$shot_h(E) = \Delta(max(E_y),\ min(E_y)) - \omega(E, \alpha)$$

where $\varphi(E)$ determines the central position of the subject elements E in a scene, and $\omega(E, c)$ is a functions that uses the constants α and β to determine the level of zoom and the height of the camera:

$$\varphi(E) = \left(\frac{\Delta(max(E_x),\ min(E_x))}{2}\right) + min(E_x)$$

$$\omega(E, c) = \Delta\big(max\big(E_y\big),\ min\big(E_y\big)\big) * c$$

The following values are used in the constants α and β to compute the respective types of shot:

- **Long shot:** $\alpha = -0.21,\ \beta = -0.14$
- **Medium long shot:** $\alpha = 0.15,\ \beta = -0.1$
- **Medium shot:** $\alpha = 0.42,\ \beta = -0.08$
- **Medium close-up shot:** $\alpha = 0.58,\ \beta = -0.06$
- **Close-up shot:** $\alpha = 0.74,\ \beta = -0.03$

The equations used to calculate the position and size of the *Camera Window* make the process of performing camera movements during the compositing process easier. In a static camera, the position and size of the *Camera Window* are only calculated once at the beginning of the scene. In order to implement a camera that follows the subjects, the position of the *Camera Window* just has to be recalculated in each frame of the scene. Zoom in and zoom out movements also can be easily achieved by performing a linear interpolation between the initial values of α and β and the final values in the target type of shot.

6.2.2.3. Video Editing

In filmmaking, the video editing process occurs during the post-production phase, where the editor selects the best shots from the raw footage, and combines them into sequences to create a finished motion picture. However, in video-based interactive storytelling, there is no post-production phase. All the editing tasks must be done in real-time during compositing process. It is similar to a live TV

show or a live sport transmission, where the editor has to switch between different cameras without knowing beforehand the actions taken by all the actors.

When it comes to narrative storytelling, camera angle selection is a crucial editing decision. According to Brown (2011), a carefully-chosen camera angle can heighten the dramatic content of the story, while a carelessly picked camera angle may distract or confuse the audience by depicting the scene so that its meaning becomes difficult to be comprehended. Therefore, the selection of camera angles is one of the most important factors in constructing a picture of continued interest (Mascelli 1965). However, cinematography does not define strict rules on how to select the best shots. Usually, each director and editor has his own style and he/she defines the shots according to his/her own knowledge and preferences.

In the proposed system, the real-time video editing task is performed by the Editor agent, which uses cinematography knowledge of video editing to select the best cameras to film the scenes and to keep the temporal and spatial continuity of the film by avoiding jump cuts and selecting the most adequate shot transition for the scenes.

The first step of the video editing phase comprises the process of selecting the best camera configuration to film the action. The proposed approach to solve this problem consists of encoding the knowledge of a real film editor into our system. This knowledge is represented by means of several artificial neural networks trained to solve cinematography problems involving camera shot selection.

The proposed model to represent the knowledge of a real film director is illustrated in Figure 6.13. For each type of scene (e.g. dialog scene, chasing scene, fighting scene), there are two neural networks: the first one is trained to classify the best camera angle for the shot based on geometric information extracted from the scene; and the second is trained to select the best type of shot based on the camera angle selected by the first neural network and emotional information extracted from the characters participating in the scene. Distinct neural networks are used for each type of scene because the number of input and output variables available depends on the type of scene and the number of involved actors. Moreover, the choices of camera angle and shot type may change substantially in different types of scenes.

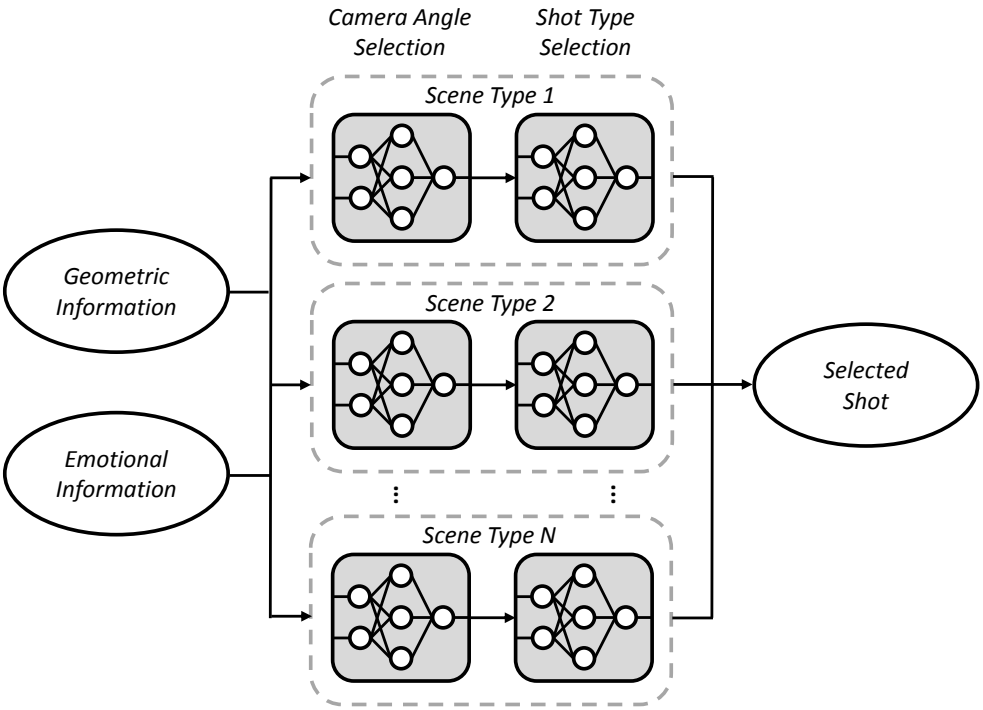


Figure 6.13: Neural network system.

The proposed method uses single hidden layer neural networks trained by a standard back-propagation learning algorithm using a sigmoidal activation function (Rumelhart et al. 1986). The structure of this type of neural network is defined in terms of input, output, and hidden layers. The input of the neural network used for selecting the best camera angle comprises a set of geometric features extracted from the scene setup. It includes the angle and position of the characters participating in the action (X, Y and Z-index, relative to the center of the scene and arranged considering the order of importance of the characters in the scene), and the id of the action performed by the main character. The number of input values depends on the type of scene and the number of characters involved in the action. For example, a dialog scene between two characters includes 9 input values, and consequently, 9 nodes in the input layer of the neural network for this type of scene. The output of this neural network comprises the possible camera angles proposed by the Cameraman agent during the camera placement phase. For example, in a scene of a dialog between two characters, there are 3 possible camera angles, and consequently, 3 nodes in the output layer of this neural network (Figure 6.14). When the output is calculated, the activated neuron in the output layer indicates the selected camera angle to be used in the shot.

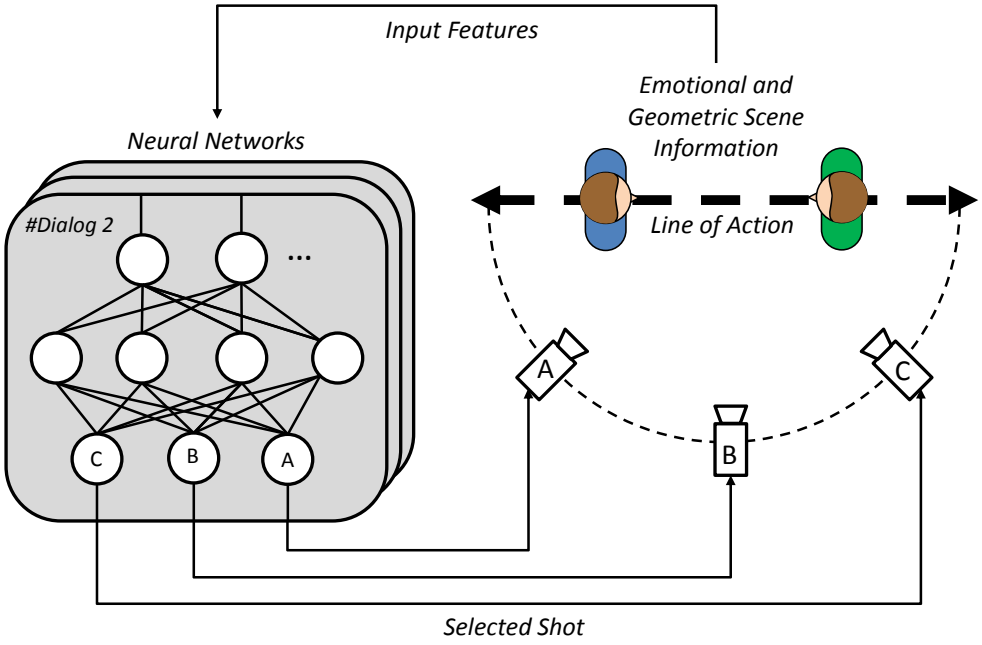


Figure 6.14: Structure of the camera selection neural network for a scene of a dialog between two characters.

With the camera angle selected by the first neural network, the next step is the selection of the type of shot. Usually, the decision of the best type of shot depends on the emotional content of the scenes (Mascelli 1965). More intimate shots, like close-ups, are often employed when there is a substantial change in the emotions of characters, highlighting the facial expressions of the subjects (Bowen and Thompson 2009). The input of the neural network used for selecting the best type of shot comprises a set of emotional features extracted from the Emotions and Relations Network (presented in Section 6.1.3). It includes the variation (relative to the previous shot) of the six emotions and the relations of characters participating in the action, together with the id of the camera angle selected by the first neural network. The number of input values depends on the number of characters involved in the action. For example, a dialog scene between two characters includes 15 input values (15 nodes in the input layer). The output of this neural network is composed of 5 nodes, which represent the five most common types of shot (close-up, medium close-up, medium shot, medium long shot and long shot). When the output is calculated, the activated neuron in the output layer indicates the selected type of shot.

Artificial neural networks are not intelligent by themselves – they need to be trained with a collection of training samples to create a classification function capable of recognizing similar situations in the future. In order to collect samples to be used as training data for the neural networks, we simulated 50 scenes and, for each one, the best shot (angle and shot type) was selected according to the vision of a real film editor. Each decision generates one training sample, which includes all the features used as input for the neural networks, together with the selected camera angle and shot type for the simulated scene. Once the neural networks were trained, they can be used in real-time to select the best cameras to film the scenes.

After selecting the best camera configuration to film the action, the next step of the video editing phase comprises the verification of the occurrence of jump cuts and the selection of the most adequate shot transition.

Every time a new camera angle is selected to film the scene, a transition between shots occurs. An important principle of cinematography is that such transitions should be unobtrusive and sustain the audience's attention on the narrative (Mascelli 1965). With an effective editor, the audience will not notice how shots of various frame sizes and angles are spliced together to tell the story. One way of complying with this principle is to avoid jump cuts. As detailed in Chapter 3, a jump cut is often regarded as a mistake in classical editing (Butler 2002). It usually occurs when two very similar shots of the same subject are joined together by a cut, producing the impression that the subject "jumps" into a new pose, causing a disorientation effect in the audience. There should be a definite change in image size and viewing angle from shot to shot. Another important cinematography principle used by conventional editors to join and maintain the continuity between shots is the use of adequate scene transitions. As introduced in Chapter 3, there are four basic ways to transit from one shot to another: cut, dissolve, wipe and fade. Each type of transition has its own applications and meanings.

Both detection of jump cuts and selection of shot transitions are based on visual similarities between shots. However, at this phase of the compositing process, the image frames of the scene have not yet been generated yet. In order to perform a visual comparison of the shots, the Editor agent requests to the Scene Composer agent the generation of a simulated first frame of the current scene

based on the information that have been defined during the previous steps of the compositing process. Moreover, the agent also keeps a copy of the last frame of the last scene that has been composed by the system. In this way, it is possible to analyze the visual similarities between the shots and make the most adequate editing decision.

The proposed approach to create a computer program that is able to automatically edit video segments consists in translating cinematography principles and practices directly into logical rules. We propose the use of a similarity scale to classify the transition between two consecutive shots and to detect possible jump cuts. Firstly, given two consecutive shots C_x and C_y, we calculate the histogram C_x^h of the last frame of C_x and the histogram C_y^h of the first frame of C_y. Then we use a metric such as a correlation coefficient to determine the degree of similarity between shots:

$$F_{coor}(C_x, C_y) = \frac{\sum_i (C_x^h(i) - \overline{C_x^h})(C_y^h(i) - \overline{C_y^h})}{\sqrt{\sum_i (C_x^h(i) - \overline{C_x^h})^2 \sum_i (C_y^h(i) - \overline{C_y^h})^2}}$$

where $C_x^h(i)$ and $C_y^h(i)$ are the histogram values in the discrete interval i and $\overline{C_x^h}$ and $\overline{C_y^h}$ are the histogram bin averages. High values of correlation represent a good match between the frames, that is: $F_{coor}(C_x, C_y) = 1$ represent a perfect match and $F_{coor}(C_x, C_y) = -1$ a maximal mismatch.

Using the histogram correlation coefficient, we define three classes of similarity $\{S_1, S_2, S_3\}$. These classes can be expressed by the following rules:

- If $F_{coor}(C_x, C_y) \in (\beta,\ 1]$ then similarity class is S_1.
- If $F_{coor}(C_x, C_y) \in [\alpha,\ \beta]$ then similarity class is S_2.
- If $F_{coor}(C_x, C_y) \in [-1,\ \alpha)$ then similarity class is S_3.

The similarity scale and the classes of similarity are illustrated in Figure 6.15. The similarity class S_1 represents a class of high similarity between frames and no editing procedure is required. Transitions of videos with similarity S_1 are

not noticed by the audience. The similarity class S_2 represents a condition that causes jump cuts. In this case, no shot transition can be applied and the Editor agent should select a new shot for the next scene. The similarity class S_3 represents a class of low similarity between frames. In this case, the transition between videos can be done using a cut, dissolve, wipe or fade, without causing jump cuts.

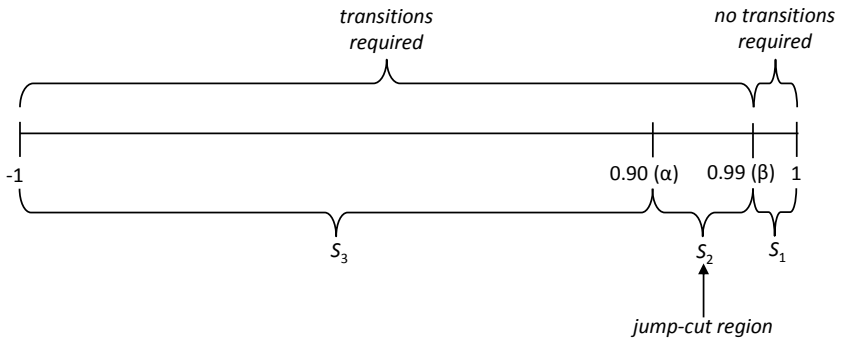


Figure 6.15: Similarity Scale (the values of α and β are experimental).

When a transition between two shots is classified as similarity class S_2, the video sequence must be changed to avoid the jump cut. If the next scene is a static one, the Editor agent uses the extra video source provided by the master scene video of the static scene to show the same action from a different camera angle, which guarantees an S_3 situation without breaking the film continuity and avoiding the jump cut; otherwise, if the scene is being dynamically composed by the system, the Editor agent returns to the previous step and selects a new camera angle and shot type based on the second most activated output neuron of the neural networks, which guarantees a different camera angle and S_3 situation in the similarity scale.

The transitions between shots classified in the similarity class S_3 lead us to the next step of the editing process: the selection of adequate transitions. To identify the most adequate transition to join two consecutive shots, we formulate a set of rules based on the cinematography literature to classify the shots into the four basic classes of transitions. Considering the functions $L_T(x)$ and $L_S(x)$ that return the temporal and spatial locations of a video segment x based on its plot action chain, we define the following transition rules:

$$\forall C_x \forall C_y \left(L_T(C_x) = L_T(C_y)\right) \wedge \left(L_S(C_x) = L_S(C_y)\right) \Rightarrow T_{cut}$$

$$\forall C_x \forall C_y \left(\left(L_T(C_x) \neq L_T(C_y)\right) \vee \left(L_S(C_x) \neq L_S(C_y)\right)\right) \wedge \left(F_{coor}(C_x, C_y) \geq 0.75\right) \Rightarrow T_{dissolve}$$

$$\forall C_x \forall C_y \left(\left(L_T(C_x) = L_T(C_y)\right) \wedge \left(L_S(C_x) \neq L_S(C_y)\right)\right) \wedge \left(F_{coor}(C_x, C_y) \leq 0.75\right) \Rightarrow T_{wipe}$$

$$\forall C_x \forall C_y (\neg C_x \vee \neg C_y) \Rightarrow T_{fade}$$

Video segments classified as class T_{cut} are ready to be processed and the direct cut can be executed. The video segments classified as class $T_{dissolve}$, T_{wipe} or T_{fade} must pass through another analyzer to determine the duration of the transition. In a conventional editing process, the editor usually uses the duration of a transition to represent the temporal variation that occurs during the transition. Based on this idea and considering t the exhibition time (usually in minutes or seconds) and T the story time (usually hours or years), the duration of the transition t_d is given by:

$$t_d(C_x, C_y) = t_{min} + \left(\frac{(\Delta L_t(C_x, C_y) - T_{max})(t_{max} - t_{min})}{T_{max} - T_{min}}\right)$$

where T_{max} and T_{min} represent respectively the maximal and minimal temporal variation in the story time, and the variables $t_{\max}$ and $t_{\min}$ represent the maximal and minimal duration of the transition. Usually $t_{min} = 0.5$ and $t_{max} = 2.0$ seconds (the minimal and maximal time for a transition, using dissolve transition as a reference).

Figure 6.16 illustrates the process of computing the transition between two consecutive shots using the proposed editing method.

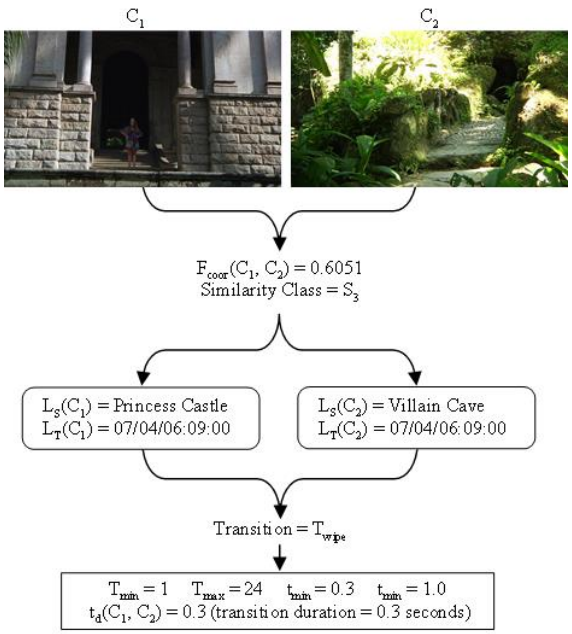


Figure 6.16: Example of a transition computation.

6.2.2.4. Color and Lighting Effects

In films, emotions can be expressed not only through the dramatization of actors, but also through colors, lighting and other visual effects. In filmmaking, the director of photography is responsible for the quality of the photography and the cinematic look of the film. Using his/her knowledge of lighting, lenses, cameras, and film emulsions, the director of photography creates the appropriate mood, atmosphere, and visual style of each shot to evoke the emotions required for each scene (LoBrutto 2002).

In the proposed video-based dramatization system, the Director of Photography is the agent responsible for defining the visual aspects of the narrative, manipulating the illumination and applying lens filters to improve and create the emotional atmosphere of scenes. The proposed approach to create an autonomous agent capable of performing this task in real-time consists of encoding the knowledge of a real director of photography into our system. This knowledge is represented by means of an artificial neural network trained to solve cinematography problems involving the selection of the emotions of scenes.

The neural network used to represent the knowledge of the Director of Photography follows the same formalism of the one used by the Editor agent. Its input comprises a set of 6 emotional features, the relations of the characters

participating in the action, and another feature describing the mood of the location where the scene is happening. All features are extracted from the Emotions and Relations Network and their values are calculated in a way to reflect the overall mood of the whole scene described by a single automaton according to the intensity of the characters' emotions and their importance to the narrative.

The emotion ε of an actor A in a scene S is given by:

$$A_\varepsilon(S) = max([A_\varepsilon^1, A_\varepsilon^2 \dots A_\varepsilon^n] \in S) * A_i$$

where *max* is a function that returns the maximum intensity value of the emotion ε of the actor A along the scene, and $A_i \in [0 \dots 1]$ is the importance factor of the actor in the narrative.

Considering a scene composed of a set of actors S_α, the overall emotion of the scene to be used as input to the neural network can be calculated by:

$$S_\varepsilon(S_\alpha) = max\big(A_\varepsilon(S_\alpha)\big)$$

The scene affectivity can be calculated based on the affectivity average of the actors participating in the scene. Considering $A_{\beta(x,y)}$ the set values of affectivity of an actor x to an actor y along a scene S, the actors' affectivity can be described as:

$$A_\beta(S) = \frac{1}{n}\sum_{i=0}^{n-1} A_{\beta(x,y)}^i \in S$$

Similarly, the affectivity of a scene composed of a set of actors S_α can be calculated by:

$$S_\beta(S_\alpha) = \frac{1}{n_a}\sum_{i=0}^{n_a-1} A_\beta\big(S_\alpha^i\big)$$

The output of the neural network comprises a set of emotional profiles, which describe specifics moods and the visual effects that must be simulated in

order to produce their respective emotions in the scene. Table 6.1 shows examples of emotional profiles used in the implementation of the proposed video-based dramatization system.

Profile	Visual Effect
Sad Scene	Soft sepia lens filter.
Fear Scene	Low brightness and soft film grain effect.
Anger Scene	Warm lens filter.
Happy Scene	High brightness.
Tension Scene	Soft warm lens filter.

Table 6.1: List Emotional profiles used by the Director of Photography agent.

In order to collect samples to be used as training data for the neural network, we simulated 50 scenes and, for each of them, the best emotional profile was selected according to the vision of a real director of photography. Each decision generates one training sample, which includes all the features used as input for the neural network, together with the selected emotional profile for the simulated scene. Once the neural network is trained, it can be used in real-time to select the best visual effects to represent the scenes' emotions.

6.2.2.5. Frame Compositing

After defining the whole structure of a scene, starts the actual process of compositing video frames representing the scene based on the information defined in the previous steps of the compositing process. In the proposed architecture, this task is performed by the Scene Composer agent, who employs parallel and optimized image processing algorithms to assemble the multiple visual elements that compose the scene into a single piece of motion picture in real-time.

The process of compositing video frames is the most time-consuming task and must be performed in real-time. The system must be able to generate at least 30 frames per second. In order to accomplish this task, it is proposed a parallel architecture capable of managing and compositing multiple video frames at the same time (Figure 6.17). In this architecture, the Scene Compositing Control

manages the compositing process and the execution of several threads that are responsible for compositing the frames. To each thread, a specific frame of the scene is assigned and, when a thread finishes compositing a frame, it is added to the Frame Buffer, which is an ordered list of frames that are ready for exibition.

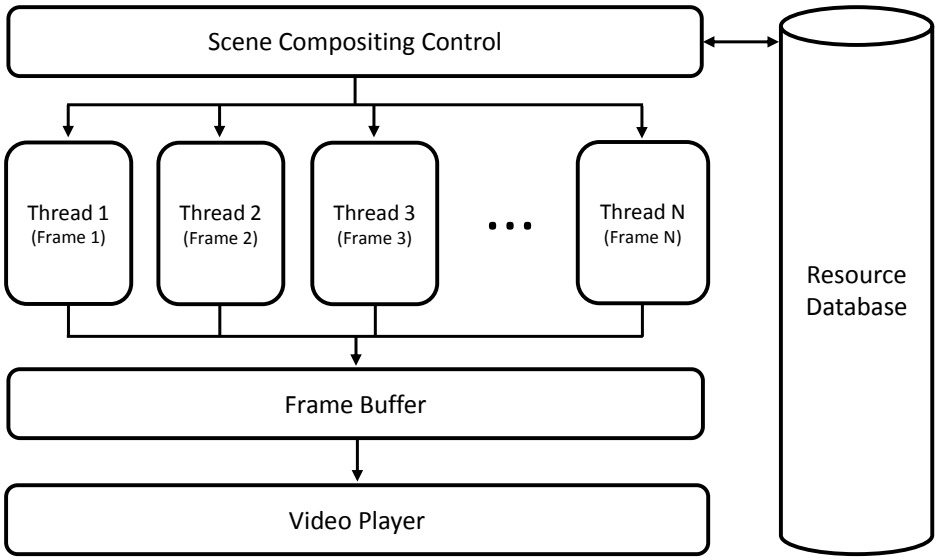


Figure 6.17: Parallel video compositing architecture.

Each compositing thread generates its assigned frame according to the information provided by the scene structure. The pseudocode of compositing algorithm is illustrated in Figure 6.18. The algorithm receives the id of the frame that has to be generated (`frame_id`) and a reference to the current scene structure (`scene_structure`). The compositing process starts by retrieving the background frame (`bg_frame`) of the current scene location defined in the scene structure according to the angle chosen by the Editor agent to film the scene. It is important to notice that the frame is retrieved based on the id of the frame that has been assigned to the thread (`frame_id`). Then, for each scene element present in the scene structure (actors and location layers), the algorithm retrieves its frame (`element_frame`) and alpha mask frame (`mask_frame`) based on the `frame_id` and orientation of the element in the scene. Next, the `element_frame` and `mask_frame` are combined to create an RGBA image (`alpha_frame`) that uses the `mask_frame` as the alpha channel of the image. Then, the `alpha_frame` is resized according to the width and height of the scene element that were defined in the previous steps

of the compositing process. The next step consists of a clipping operation, which is performed over the `alpha_frame` in order to eliminate parts of the element that are not inside of the frame region defined by the angle and type of shot selected by the Editor agent. Before blending the `alpha_frame` with `bg_frame`, a color correction operation is performed to adjust the color of `alpha_frame` according to the color of its area in `bg_frame`. Then, an alpha GPU compositing operation is performed to blend the resulting `alpha_frame` with the `bg_frame`, which completes the compositing process of the scene element. Once all scene elements have been composed, the emotional profile selected by the Director of Photography is simulated by applying some color filters and lighting effects in `bg_frame`. After applying the visual effects, the algorithm returns the composed frame (`bg_frame`).

```
1. function compose_frame(frame_id, scene_structure)

2.   get bg_frame of frame_id from the location defined in
                                                    scene_structure

3.   foreach scene_element in scene_structure do
4.     get element_frame and mask_frame of frame_id from
                                                      scene_element
5.     combine element_frame and mask_frame to create a alpha_frame
6.     resize alpha_frame according to the size of scene_element
7.     perform clipping operation in alpha_frame
8.     correct the color of alpha_frame based on bg_frame
9.     perform an alpha GPU compositing operation blending
                                          alpha_frame with bg_frame
10.  end

11.  apply color and lighting effects in bg_frame to simulate
                                    the selected  emotional profile

12.  return bg_frame
13.end
```

Figure 6.18: Pseudocode of the compositing algorithm.

The process of creating the alpha frame consists in adding the alpha mask to the RGB video frame of the scene element as an alpha channel, which creates an RGBA image. This image contains, besides the RGB color information, an extra alpha channel that retains the matte information. Combining the frames into a single image simplifies the resize operation that is performed in the next step of

the compositing algorithm. The scaling algorithm used to resize the frames is based on the nearest-neighbor interpolation method, which is the fastest image scaling algorithm implemented by the OpenCV library (Bradski and Kaehler 2008).

The clipping operation is used to eliminate parts of the scene elements that are not inside of the rectangular region of the frame, which is defined by the configuration of the virtual camera (position, angle and type of shot). The OpenCV library offers an optimized way of performing clipping operations by allowing the definition of a region of interest (ROI) in images. Image processing algorithms only operate inside of the region defined by the ROI. In this way, the clipping operation consists of calculating the region of the scene element that is inside of the rectangular region of the scene frame based on its position and size, and then adjusting the ROI of the scene element to match the scene frame.

The color correction algorithm used in the frame compositing process is based on the exposure compensation method proposed by Brown and Lowe (2007) to correct color differences in panorama image stitching, which is the process of combining multiple images with overlapping fields of view to produce a segmented panorama or high-resolution image. The approach proposed by Brown and Lowe (2007) adjusts the intensity gain level of the images to compensate for appearance differences caused by different exposure levels.

The exposure compensation method is used in the frame compositing process to adjust the exposure levels of the scene elements based on the background frame. The intensity gain of the frame I_i of a scene element is given by the error function defined by the sum of normalized intensity gain errors for all overlapping pixels in the background frame I_j:

$$e = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} N_{ij}\left(\left(g_i\bar{I}_{ij} - g_j\bar{I}_{ji}\right)^2/\sigma_N^2 + (1-g_i)^2/\sigma_g^2\right.$$

where N_{ij} is the number of pixels in frame i that overlap frame j, and $\bar{I}_{ij}$ is the mean value of frame I_i in the overlapping region of frame I_i and I_j. The parameters σ_N and σ_g represent the standard deviation of the intensity errors and the gain standard deviation respectively, which have been empirically set to

$\sigma_N = 10.0, I \in \{0..255\}$ and $\sigma_g = 0.1$ (Brown and Lowe 2007). The criterion employed to determine the intensity gain level aims at the minimization of the error function with respect to the gain g.

The alpha compositing operation is applied to blend the alpha frame of the scene element with the background frame. The algorithm uses an "over" operator to blend the color and alpha values of the images together on a pixel-by-pixel basis. Considering r_A, g_A and b_A the RGB components that define the color of a pixel in element A with alpha value α_A, and r_B, g_B and b_B the RGB components that define the color of a pixel in the background element B with alpha value α_B, the resultant RGB color components of the pixel r_C, g_C and b_C are given by:

$$r_C = \alpha_A r_A + \alpha_B r_B (1 - \alpha_A)$$
$$g_C = \alpha_A g_A + \alpha_B g_B (1 - \alpha_A)$$
$$b_C = \alpha_A b_A + \alpha_B b_B (1 - \alpha_A)$$

The alpha compositing algorithm runs on GPU and takes advantage of its parallel architecture to compute the color of several pixels simultaneously, which improves the performance of the process and allows the system to compose the interactive scenes in real-time.

The process of applying the emotional profile selected by the Director of Photography in the generated frames consists of simulating lens filters and modifying the brightness and contrast of final frame according to the selected profile. Lens filters are simulated by overlaying a colored image layer over the final frame. The opacity of the image layer controls the intensity of the simulated filter. The brightness and contrast are automatically adjusted to match the description of the emotional profile. Both operations are performed on GPU to improve the performance of the compositing process.

6.2.3. Film Scoring

In filmmaking, scoring consists in the process of writing and compositing the soundtracks of the film. Music is an integral part of the film, as it helps to connect the emotional content with the events on the screen. During the post-

production phase, the music director is the person in charge of creating the mood and the emotional atmosphere of the film.

As a traditional film, a video-based interactive narrative must emphasize the dramatic content of the story and use music as a tool to express the emotions of the narrative. In the proposed video-based dramatization system, the agent Music Director is responsible for selecting the most adequate soundtracks and manipulating the audio of the narrative to create the emotional atmosphere of each scene according to the information provided by the Director agent.

6.2.3.1. The Music Director Agent

The proposed approach to create an autonomous agent capable of selecting the most adequate soundtracks to create the emotional atmosphere of the narratives in real-time consists of encoding the knowledge of a real music director into our system. This knowledge is represented by means of an artificial neural network trained to solve cinematography problems involving the selection of the best soundtracks for the narrative scenes.

The neural network used to represent the knowledge of the Music Director agent is very similar to the one used in the Director of Photography agent. Its input comprises the same set of emotional features extracted from the Emotions and Relations Network and its output consist of a set of emotional profiles describing specifics moods and the soundtracks that can be used to produce their respective emotions in the narrative. Table 6.2 shows examples of emotional profiles used in the implementation of the proposed video-based dramatization system.

In order to collect samples to be used as training data for the neural network of the Music Director agent, we simulated 50 scenes and, for each of them, the best emotional profile was selected according to the vision of a real music director. Each decision generates one training sample, which includes all the features used as input for the neural network, together with the selected emotional profile for the simulated scene. Once the neural network is trained, it can be used in real-time to select the best soundtracks for the narrative.

Scene	Audio
Happy Scene	Soundtrack with major keys; rapid tempos; high pitched with large variations.
Sad Scene	Soundtrack with minor keys; slow tempos; narrower range melodies.
Fear Scene	Soundtrack with rapid tempos; dissonance; small pitch variations.
Anger Scene	Soundtrack with minor keys; fast tempos; high pitched.
Tension Scene	Soundtrack with minor keys; ascending melodies; Dissonant harmonies.

Table 6.2: List Emotional profiles used by the Music Director agent.

6.3. Conclusion

This chapter described the technical details of the implementation of all cinematography-based agents that compose the proposed dramatization system, including the algorithms for real-time video compositing and editing.

The system was inspired by cinematography theory and designed to apply and respect the main cinematography principles and rules in the video compositing process. The proposed technique to represent the knowledge of some cinematography-based agents using a collection of artificial neural networks trained by their corresponding filmmaking professionals allows the system to learn the personal style of the human professionals and replicate it during the video compositing process, giving to the system the ability to apply the cinematography rules and principles while keeping the signature of the human artist in the computer generated content.

The next chapter will present the technical details about the implementation of the user interaction module and the multi-user interaction mechanisms of the proposed system.

7
User Interaction

As introduced in the previous chapters, the proposed video-based interactive storytelling system supports two types of user interactions: global and local. Global interactions enable users to suggest events to next story chapters and local interactions allow users to interfere on the execution of the nondeterministic automata. The Suggestion Manager is the main module of the user interaction system and is responsible for managing both forms of interaction, including the process of interpreting and extracting meaning from the suggestions given by users in natural language. While global suggestions are continuously collected by the system, local user interventions occur only at specific points of the narrative in parallel with the global user interaction.

Figure 7.1 illustrates a flowchart of the global and local interaction processes executed by the Suggestion Manager. The global interaction process starts when the module receives from the story generator a set of valid global suggestions represented as simple first-order logic sentences. Once the suggestions have been received, they are then sent to all connected interaction mechanisms to be handled by their respective user interfaces. Then, the Suggestion Manager starts to collect suggestions from the interaction mechanisms. When a new user suggestion is received, it is processed and interpreted through a natural language processing algorithm. The extracted content is then used to update the votes for the global suggestions. This process is executed continuously and, every time the story generator requests a suggestion to be incorporated into the story, the most voted suggestion is sent and removed from the current list of suggestions. The list is maintained and future votes are calculated together with the previous suggestions.

The local interaction process is executed in parallel with the global interaction. It starts when a new set of local interaction options are received from the story dramatization module. Then, similarly to the global interaction process, the interaction options are sent to all interaction mechanisms, suggestions are

collected, interpreted and votes are calculated. However, different from the global interaction, the local interaction process can be interrupted by the dramatization module at any moment. When it happens, the most voted option is selected and the others are discarded. Usually, users have around 15 seconds to decide and vote on desired options.

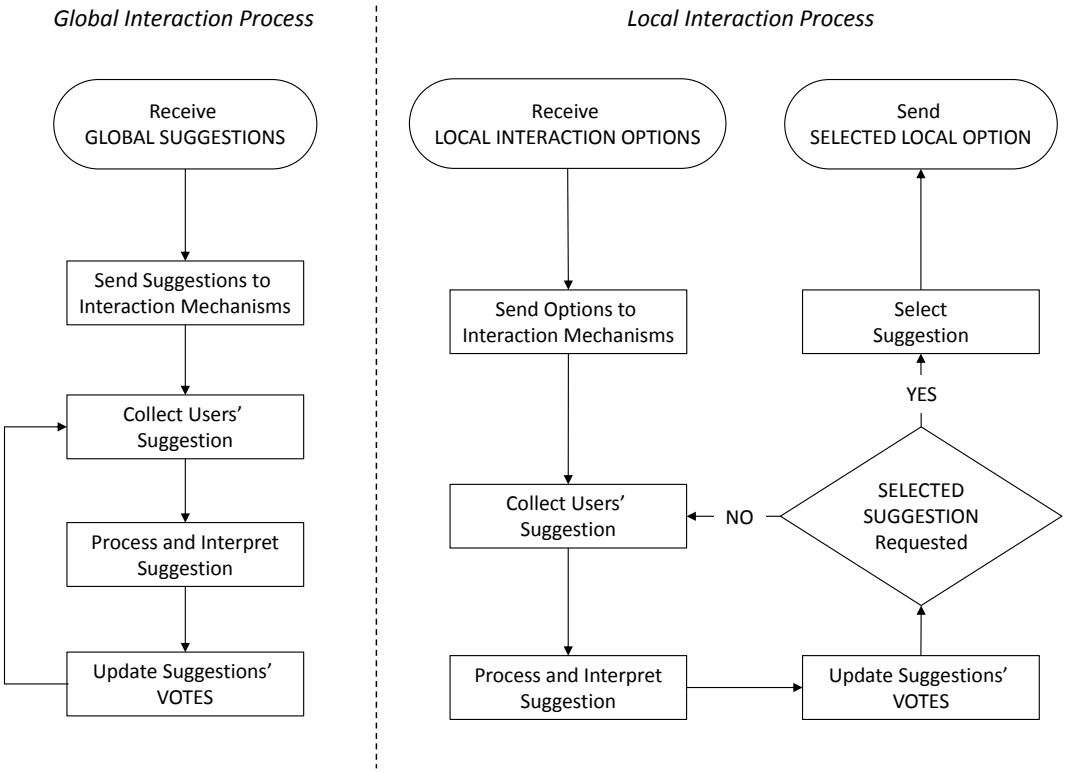


Figure 7.1: Flowchart of the global and local interaction processes executed by the Suggestion Manager module.

The most complex task performed by the Suggestion Manager consists in the process of interpreting and extracting meaning from the users' suggestions, which involves natural language processing. The next section will describe this process and Section 7.2 will present more details about the interaction mechanisms.

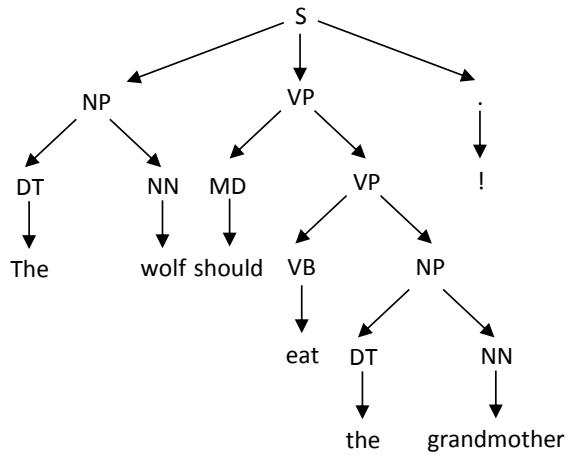
7.1. Natural Language Processing

A traditional natural language processing task consists of two main phases (Jurafsky and Martin 2000): (1) syntax parsing, where the syntax tree and the

grammatical relations between the parts of the sentence are extracted; and (2) semantic analysis, which is the extraction of the meaning of words or phrases.

In order to interpret the users' suggestions, we adopted the Stanford Parser to perform the syntax parsing of the sentences. The Stanford Parser (Stanford 2014) is a probabilistic parser that represents all sentence relationships as typed dependency relations instead of using phrase structure representations. However, it also produces phrase structure trees.

The Stanford Parser is capable of producing 55 different typed dependencies (Marneffe and Manning 2008). These dependencies reflect the grammatical relationships between the words. Such grammatical relations provide an abstraction layer to the pure syntax tree and provide information about the syntactic role of all elements. Figure 7.2 (*a*) shows a phrase structure tree generated by the Stanford Parser for the sentence "*The wolf should eat the grandmother!*". The corresponding typed dependencies are listed in Figure 7.2 (*b*). Typed dependencies facilitate the analysis of semantic relationships between words based on both their grammatical relationships and overall sentence syntactical structure.



(a)

det(wolf-2, The-1)
nsubj(eat-4, wolf-2)
aux(eat-4, should-3)
root(ROOT-0, eat-4)
det(grandmother-6, the-5)
dobj(eat-4, grandmother-6)

(b)

Figure 7.2: Phrase structure tree (a) and the typed dependencies (b) of "*The wolf should eat the grandmother!*".

The typed dependencies are all binary relations, where a grammatical relation holds between a "*governor*" and a "*dependent*". In the above example, the

relation nsubj (nominal subject) relates the noun "*wolf*" with the corresponding verb "*eat*", whereas the relation dobj (direct object) relates this verb with the object "*grandmother*". In this way, the sentence elements are extracted and the sentence structure can be translated into simple first-order logic sentences. In the above example, the following sentence is extracted:

```
eat(wolf, grandmother)
```

which means that the "w*olf*" must perform the action "*eat*" and the victim is the "*grandmother*".

In the present work we generate simple logic sentences composed by a conjunction of predicates, e.g. from "*The wolf should eat Anne and his grandmother!*" is generated the sentence:

```
eat(wolf, anne) ∧ eat(wolf, grandmother)
```

With this dependency chain, the system is able to extract "subject – direct object" relationships from sentences. However, for this pattern to be valid, four conditions must be met: (1) a nominal subject (nsubj) dependency must exist; (2) the dependent of the nsubj dependency must be a family member (in the phrase structure tree); (3) the governor of this dependency must be a verb, which means that a family member is the head noun of the subject of a clause which is predicated by the verb; and (4) a direct object (dobj) dependency must exist and the governor of this dependency must match the index of the governor of the nsubj dependency – then we assume that the dependency of the dobj relation is paired with the family member found initially.

In the example above, the extracted logical sentence already contains the semantic meaning needed by our interaction system to infer a valid suggestion to the story. However, there are some cases where the subjects are not directly referenced. For example, in the sentence "John saves the grandmother and marries her.", the pronoun "her" refers to "grandmother". However, when we compute the typed dependencies for this sentence (Figure 7.3), we see in the relation "dobj(marries-5, her-6)" that the pronoun "her" was not resolved and, in some cases, it's not possible to solve it using only the phrase structure tree. The process

of resolving what pronoun or a noun phrase refers to is called anaphora resolution. To solve this problem, we used another tool from the Stanford Natural Language Processing Group, the Stanford Deterministic Coreference Resolution System (Raghunathan et al. 2010), which is able to indicate precisely the correct reference of any unknown pronoun.

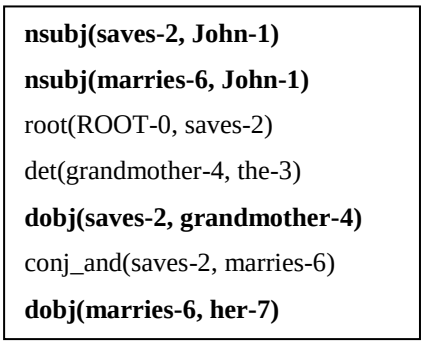


Figure 7.3: Example of anaphora problem in the sentence "*John saves the grandmother and marries her*".

The parser also verifies the occurrence of negations. For example, in the sentence "*The wolf should not eat Anne!*", the adverb "*not*" completely changes the meaning of the sentence. To identify negations, the parser analyses the occurrence of negation modifiers ("*neg*") in the typed dependency list. Figure 7.4 illustrates the typed dependency for the example above and the occurrence of the negation modifier.

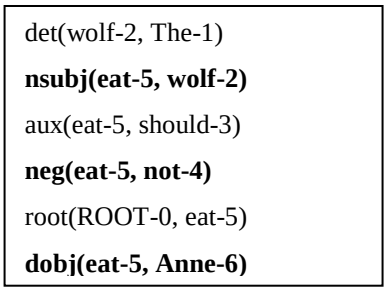


Figure 7.4: Example of negation in the sentence "*The wolf should not eat Anne!*".

After translating the "subject – direct object" relations into first-order logic sentences, the parser also needs to validate the sentences. For example, the predicate "$eat(CH_1, CH_2)$" requires a nominal subject CH_1 that is a valid character

and a direct object CH_2 that also is a valid character in the story context. Moreover, the verb "*eat*" also must be a valid action. To perform this validation, the parser has access to a list of valid actions, characters and places. In this way, the parser is able to identify the elements that the words represent. However, almost all words have synonyms and to deal with this, the parser also incorporates a dictionary of synonyms associated with each action, character and place. So, it is able to parse sentences such as "*John should annihilate the villain!*", where the verb "*annihilate*" is a synonym of the action "*kill*", and the object "villain" the role of the character "Wolf".

Ideally, the parser expects sentences that contain at least one verb, one nominal subject and a direct object. However, this does not always happen; in some cases the subject, the direct object, or both are omitted. For example, the sentence "*Kill the wolf!*" does not express directly who should perform the action "*kill*", but indicates the direct object "*wolf*" (Figure 7.5). In this case, the parser is still able to generate a partial logic sentence to represent it:

```
kill(*, wolf)
```

which means that someone "*" must perform the action "*kill*" and the victim is "wolf". The operator "*" can be replaced by any valid character to complete the logical sentence.

root(ROOT-0, Kill-1)
det(wolf-3, the-2)
dobj(Kill-1, wolf-3)

Figure 7.5: Example of omitted subject in the sentence "*Kill the wolf!*".

The entire process of extracting valid first-order logic sentences from text phrases is illustrated in Figure 7.6. In the Syntax Parsing step, the Stanford Parser receives a text phrase S_x as input and generates a Dependency Tree and the Typed Dependencies for the sentence. Using this information, in the Semantic Analysis phase, the parser performs the Anaphora Resolution process to resolve the pronouns of the sentence and find valid synonyms using the Synonym Dictionary.

Finally, the parser checks the integrity of the sentences using some Logic Rules and returns a list of valid first-order logic sentences (P_x^n).

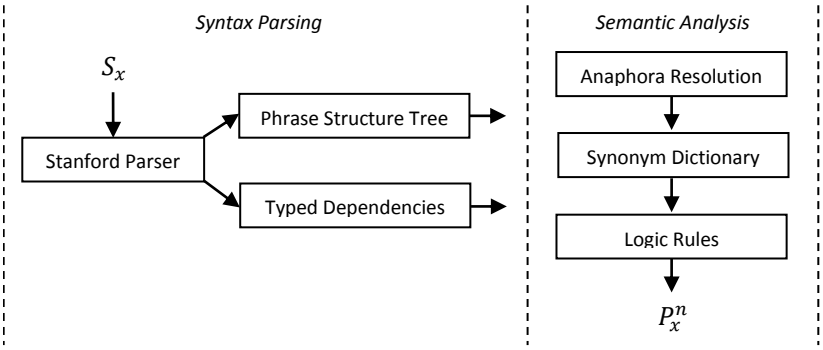


Figure 7.6: The process of extracting valid first-order logic sentences. S_x is the input text phrase and P_x^n is the output list of predicates.

Another way of identifying and extracting meaning from natural language sentences is through an analysis of the user satisfaction with some target concept. The process of extracting user's satisfaction also involves natural language processing, more specifically the area of Sentiment Analysis (Liu 2010). However, instead of using complex sentiment analysis techniques, we adopted a more simplistic approach to solve this problem. We describe the target concept as simple questions based on possible suggestions to the story (e.g. "*Would you like to see the Big Bad Wolf attacking Little Red Riding Hood in the next chapter?*"). Users usually respond to the questions positively or negatively, i.e. agreeing or disagreeing. In this way, the parser only needs to identify positive and negative answers in the users' comments.

The approach adopted by our parser to identify positive and negative answers uses a list of words, where each word W_i is associated with a numerical score $W_i^s \in [-1.0, +1.0]$. High negative scores represent very negative words and high positive scores represent very positive words. Considering C_x a user commentary, the sentiment $St(C_x)$ is given by:

$$St(C_x) = \frac{1}{n}\sum_{i=1}^{n} W_i^s \quad if \quad (W_i \in C_x)$$

where $St(C_x) \in [-1.0, +1.0]$ indicates if C_x is a positive commentary ($C_x > \beta$) or a negative commentary ($C_x < \alpha$), in which α and β defines a precision threshold where uncertain commentaries are ignored (classified as neutral).

To illustrate this process, let's consider $= -0.3$ $\beta = +0.3$, and the following user commentaries for the suggestion $S_1 =$ "*Would you like to see the princess Marian dying in the next chapter?*":

1. "*Yes!!* :)"
2. "*I would love to see it happening!!!* ;)"
3. "*No!! I love her…* :("
4. "*This story is boring…* :("

For case (1), the word "*Yes*" and the emoticon ":)" have both the score +1.0; giving the sentiment $St(C_1) = +1.0$ and classifying it as a positive commentary. In case (2), the word "*love*", "*see*" and the emoticon ";)" have the scores +0.8, +0.5 and +0.9 respectively; giving the sentiment $St(C_2) = +0.73$ and classifying the sentence as a positive commentary. In case (3), the word "*No*", "*love*" and the emoticon ":(" have the scores -1.0, +0.8 and -1.0 respectively; giving the sentiment $St(C_3) = -0.4$ and classifying the sentence as a negative commentary. Finally, in case (4) the word "*boring*" and the emoticon ":(" have the scores -0.7 and -1.0 respectively; giving the sentiment $St(C_4) = -0.85$ and classifying the sentence as a negative commentary.

7.2. Interaction Mechanisms

The user interaction module of the video-based interactive storytelling system works as a multimodal and multi-user interaction server that supports the integration of several interaction mechanisms based on suggestions. Each interaction mechanism acts as a multi-user server that has its own client interface, allowing several users to be connected in the same interaction network.

Two interaction mechanisms were integrated with the user interaction module: social networks and mobile devices. The first method is based on the idea of using social networks (such as Facebook, Twitter and Google+) as a user

interface, allowing users to collaborate with the development of the stories in a social environment. The second interaction mechanism combines the use of mobile devices (such as smartphones and tablets) with natural language to allow users to freely interact with virtual characters by text or speech. The next sub-sections will present more details about these interactions mechanisms.

7.2.1. Social Interaction

The social interaction interface is implemented through the application programming interface (API) provided by the social networks. The module is constantly sending messages through the social networks to induce facts to users or to provoke them, which we denominate "induction messages". When the system starts or when a new chapter is beginning, the induction message is an introduction to the story or chapter. Users receive this message as an update in the social network and are able to comment on the message (Facebook and Google+) or use hashtags (Twitter) to indicate suggestions. The introduction message describes the story characters, places, gives some tips about what could happen in the story and incentive the users to comment what they would like to see happening in the story. Figure 7.7 shows an example of introduction message used for fairy tale interactive story.

"*Once upon a time there was a little girl, named Anne, but mostly known as Little Red Riding Hood. She lived in a certain village with her mother, who was excessively fond of her; but always concerned about the health of Anne's grandmother, who lived in distant village. Not far away in a sinister forest, lurched the Big Bad Wolf, the evil wolf ready to eat anything that fits in his mouth. But there was also a brave woodcutter John, ready to save everyone. Uncountable stories can be told in this world of fantasy. Will Little Red Riding Hood be attacked by the Big Bad Wolf? Or killed by the monster? Will the grandmother be eaten by the wolf? Will the woodcutter save everyone?*

A new interactive story is about to begin. Comment here what you would like to see happen with the characters of this story."

Figure 7.7: Example of an introduction message.

During the dramatization of a chapter, the system keeps sending induction messages to the social networks using the global suggestions for the next chapters received from the Suggestions Manager. In this case, the induction messages have the form of simple questions (e.g. "*Would you like to see Little Red Riding Hood hitting the Big Bad Wolf in the next chapter?*"). When the social interaction module receives local interaction options, a special induction message is created in the form of an invitation for a poll (e.g. "*Little Red Riding Hood should trust the strange wolf? Yes or No?*"). In this case, a poll is created in the social networks and users are able to select and vote on the desired outcome by clicking on the poll option. Figure 7.8 illustrates the dynamic behavior of the social interaction system through an activity diagram.

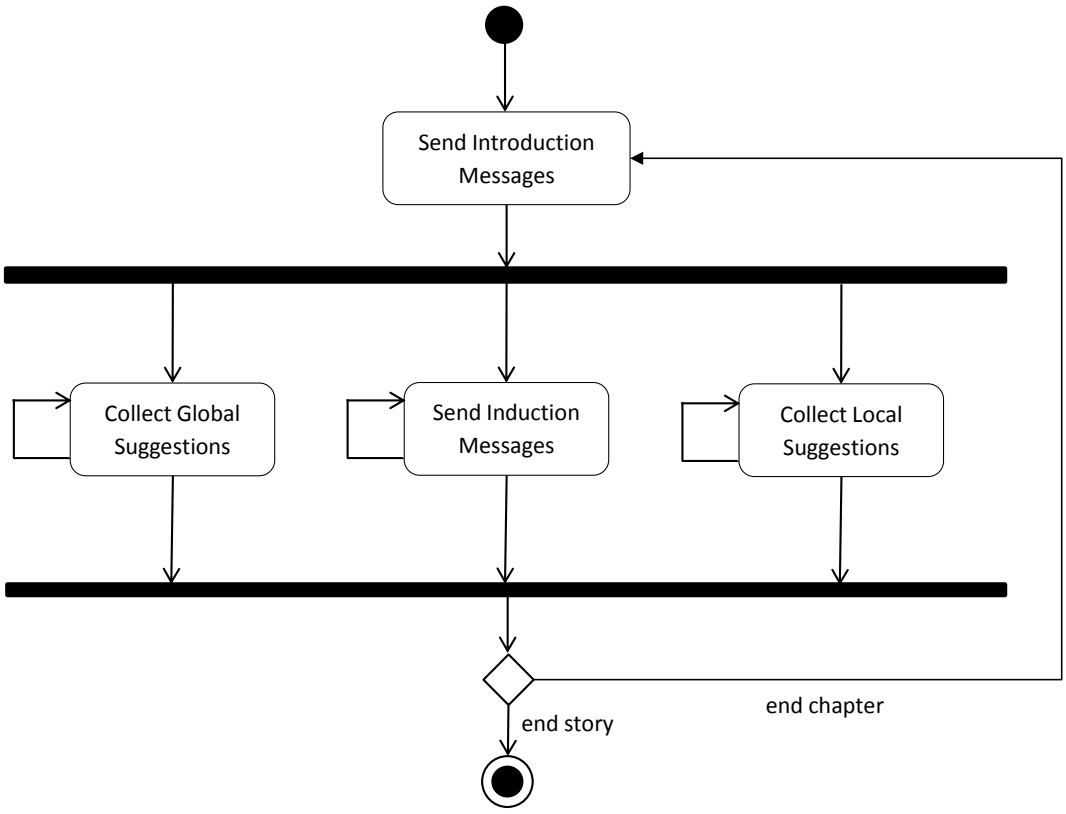


Figure 7.8: Activity diagram of the social interaction module.

There are three basic ways users can interact with the stories through social networks: (1) interaction by comments – where they explicitly express their desires through comments in natural language; (2) interaction by preferences – where they express satisfaction or state preferences; and (3) interaction by poll –

where a poll is created and users vote in what they want. The next sub-sections present more details about these interaction methods.

7.2.1.1. Interaction by Comments

The interaction by comments allows users to explicitly express their desires through comments on the social networks (Figure 7.9). Every time the system detects a new user comment in an active introduction or induction message, the content of the comment is extracted and sent to the Suggestion Manager to be interpreted and counted as a vote to the expressed suggestion.

Figure 7.9: Example of user comment expressing a suggestion on Facebook.

In the case of Facebook and Google+, besides writing comments, users are also able to “like” or “+1” a comment of another user, which indicates that they liked what the comment says. In this way, the interaction system considers the number of users that directly wrote that something should happen and the number of users that liked the respective comments.

7.2.1.2. Interaction by Preferences

The interaction by preferences allows users to express their satisfaction with the story suggestions through social networks. Instead of directly writing a

comment expressing a desire, users are able to "like" (Facebook) or "+1" (Google+) a suggestion generated by the interaction system (Figure 7.10). Users can also write comments on the generated suggestions expressing their satisfaction with the proposed events.

Figure 7.10: Example of user "liking" a system generated suggestion on Facebook.

Every time the system detects a new user comment in an active induction message that expresses a global suggestion, the content of the comment is extracted and sent to the Suggestion Manager to be interpreted by the sentiment analysis algorithm, which will classify the comment as positive or negative. Positive comments count as positive votes to the suggestion described in the post and the negative comments count as negative votes. The number of users that "like" (Facebook) or "+1" (Google+) the suggestion also count as positive votes.

7.2.1.3. Interaction by Poll

The interaction by poll allows users to choose what they want through polls in the social network. Instead of directly writing a comment or waiting for the desired suggestion to appear (posted by the interaction by preferences), they are able to see all available options and vote on the suggestion of their choice (Figure 7.11).

The interaction system is constantly checking the results of the active polls, and informing the Suggestion Manager about the most voted suggestions. The process of extracting users' choices from a poll does not require any complex algorithm. However, the importance of this method should not be underestimated, because it provides an easy way of interaction where users who do not like to

write or do not know exactly what they want are able to interact just by clicking on a poll option.

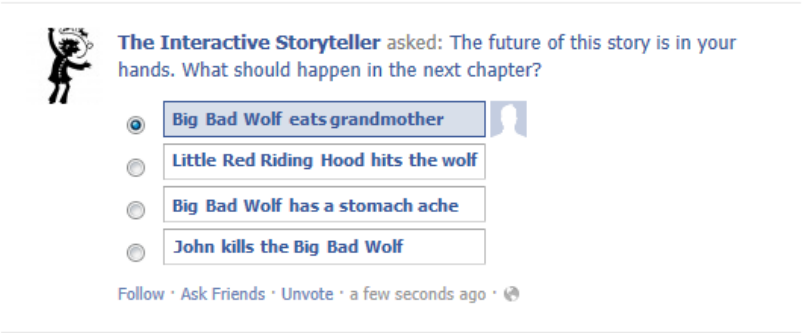


Figure 7.11: Example of poll with story suggestions generated by the system on Facebook.

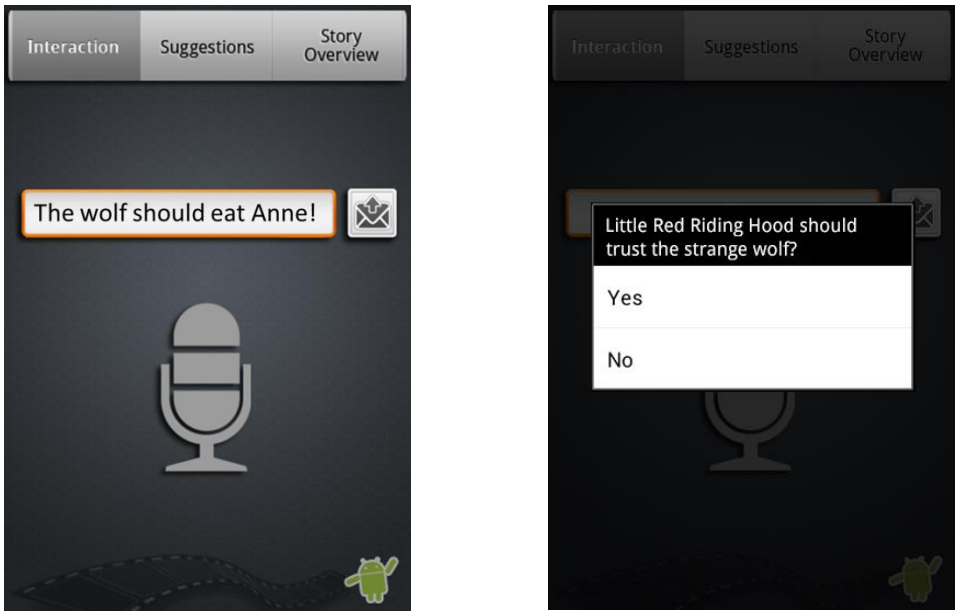
7.2.2. Mobile Interaction

The multi-user mobile interaction interface was designed to support both global and local user interactions. It incorporates spoken and written natural language in a simple mobile application, where users can write or speak what they want to happen in the story or easily select the desired outcome for local decision points. With this interaction method it is easy to imagine the possibility of watching a movie while advices are continuously being sent to the characters.

The mobile interface consists of a small application developed for Android mobile devices (such as smartphones and tablets), where users can interact with the ongoing stories by writing or speaking a suggestion/advice to the virtual characters using natural language. The user interface of this application is shown in Figure 7.12.

Users can interact with the story by typing the suggestions and advices in the text box shown on the mobile interface or by pressing the microphone icon and then speaking out the intended suggestion. The Android Speech Recognition API is used to recognize the user speech and to convert it into text. In this way, the system only needs to handle and understand written text. After reading the user input (text or speech), the mobile application sends the user suggestions through a TCP/IP connection to the Suggestion Manager, which is responsible for interpreting and managing all user suggestions.

The mobile application is also in charge of handling local user interactions. When the application receives local interaction options it displays the list of choices on the screen and alerts the user by vibrating the mobile device. Users are able to touch the desired option on the screen or speak out the intended choice (Figure 7.12 - b). The selected option is then sent to the Suggestion Manager to be counted as a vote to its respective outcome of the local decision point.



(*a*) (*b*)

Figure 7.12: Mobile user interface. Image (*a*) shows the main screen of the mobile application; and image (*b*) shows the interface during a local interaction.

7.3. Conclusion

This chapter described the technical details about the implementation of the user interaction module of the video-based interactive storytelling system and the multi-user interaction mechanisms using mobile devices and social networks.

The mobile user interface allows users to freely interfere in the narratives using their personal mobile devices, while the social interaction allows users to do the same in the multi-user environment of social networks. It is important to notice that the social interaction interface is dependent of the social network services, which are constantly changing the way external applications can interact

with their services. Consequently, the social interaction module of our system requires constant updates to adapt to these changes. In addition, it faces the risk of being unable to perform its tasks if an update in the social network API blocks its access to the required resources.

The next chapter describes the interactive narratives produced to validate our system and the technical tests performed to evaluate the algorithms used in the dramatization and in the user interaction modules of the proposed system.

8
Application and Evaluation

Two prototype interactive narratives were produced to validate the proposed approach to video-based interactive storytelling: "*The Game of Love*" and "*Modern Little Red Riding Hood*".

The Game of Love pertains to a romantic drama genre and tells the story of a young boy named Peter, who falls in love with an unknown girl and tries to do anything to get closer to her. The main characters of the narrative are: the young lover, Peter; the unknown girl, Anne; Anne's best friend, Carol; and two imaginary creatures, a little angel and a little devil. The story takes place in six main locations: a university, Peter's house, Anne's house, a party, a beach, and the city square. In the main storyline, Peter falls in love with Anne at the university and tries to know more about her by hacking her Facebook page. After getting some information, Peter manages to go out with Anne on a date, but she find out that he invaded her social network account. Users are able to influence the decisions made by the main characters and change the future of the young couple. Figure 8.1 shows some scene from "*The Game of Love*".

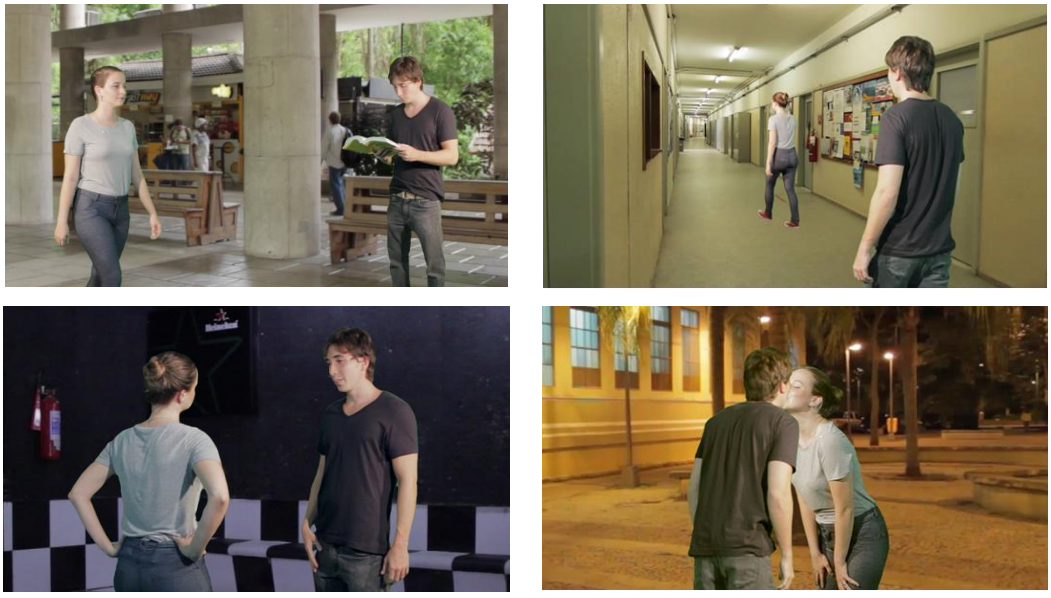


Figure 8.1: Scenes from "The Game of Love".

The second prototype video-based interactive narrative developed, "*Modern Little Red Riding Hood*", is an adaptation of the famous Little Red Riding Hood fairy tale. It tells a modern and comic version of the original story, with funny and unexpected outcomes. The main characters of the narrative are: the girl called Little Red Riding Hood, her mother, her grandmother, the Big Bad Wolf, and the woodcutter. The story takes place in three main locations: the Little Red Riding Hood house, the forest, and the grandmother's house. The prototype is able to generate a considerable number of diversified stories to comply with the users desires. In the more conventional stories, the narrative evolves following the traditional fairy tale plot with the Big Bad Wolf tricking the Little Red Riding Hood and getting to her grandmother's house first, eating the grandmother and attacking Little Red Riding Hood when she finds out what happened. In stories with a more unconventional outcome, Little Red Riding Hood celebrates the death of her grandmother, and then shares her basket of goodies with the Big Bad Wolf. In stories with a more comic outcome, the Big Bad Wolf eats both Little Red Riding Hood and her grandmother, and then gets a stomach ache. Figure 8.2 shows some scene from "*Modern Little Red Riding Hood*".

Figure 8.2: Scenes from "*Modern Little Red Riding Hood*".

In order to evaluate the proposed methods for video-based interactive storytelling from a technical point of view, we performed two tests: a performance and accuracy test to validate the methods of dramatization and user interaction,

and a visual evaluation test to compare the compositing results automatically produced by the proposed system with the results manually produced by human filmmaking professionals. The following sections describe these tests.

8.1. Technical Evaluation

The technical evaluation concerns the accuracy and the real-time performance of the video compositing and user interaction methods used in the video-based interactive storytelling system. The evaluation was mainly focused on the methods that are based on image processing and machine learning algorithms, which are the most time-consuming processes and require a validation of accuracy. Each method was evaluated individually and the results are presented in the next sub-sections. The computer used to run the experiments was an Intel Xeon E5620, 2.40 GHZ CPU and 24 GB of RAM.

8.1.1. Video Editing

The video-based interactive storytelling system implements two video editing techniques that are used in real-time to automatically select the best shots to compose the scenes and the most adequate scene transitions to join two different shots.

8.1.1.1. Shot Selection

In order to validate the shot selection method proposed in this thesis, we performed two tests: (1) a recognition rate test to check the accuracy of the predicted shots; and (2) a performance test to check the necessary time to select a new shot.

As presented in Chapter 6, the shot selection method uses two neural networks for each type of scene. The first one is trained to classify the best camera angle for the shot, and the second is trained to select the best type of shot for the selected camera angle. In order to evaluate the accuracy of the shot selection method, for each type of scene implemented in the prototype application (total of

8 types), we created 5 training sets with a different number of samples and, for each one, a test set with half the size of the corresponding training set. The samples were collected through a simulation process, where we created several scenes varying the type of scene, number of actors, emotional states and actions, and then, for each scene, we asked to a human editor to make the selection of the best shot (angle and shot type) to film the scene. Each decision generates one sample, which includes all the features used as input for the neural networks, together with the selected camera angle and shot type for the simulated scene. The training sets were used to train the neural networks and the samples of the current test set were then predicted. Table 8.1 and Figure 8.3 show the computed results of this test with the training set size ranging from 10 to 50 samples. The presented percentages of accuracy correspond to the average of the results obtained for the neural networks used in the different types of scenes.

Number of Samples	**10**	**20**	**30**	**40**	**50**
Camera Angle Accuracy	82.2%	89.6%	92.5%	95.2%	98.4%
Shot Type Accuracy	76.5%	85.3%	89.4%	93.3%	97.5%
Shot Selection Accuracy	72.3%	81.7%	87.6%	92.2%	96.2%

Table 8.1: Recognition rate of the shot selection method with training sets ranging from 10 to 50 samples.

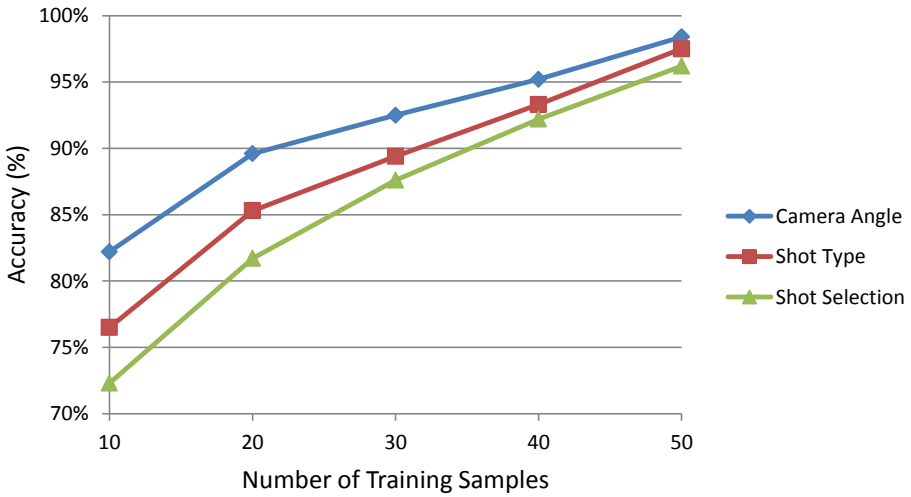


Figure 8.3: Recognition rate of the shot selection method with training sets ranging from 10 to 50 samples.

In order to evaluate the performance of the proposed solution, we used our shot selection method to predict the shots for a sequence of 5 scenes, with a total of 30 different shots. For each shot, we calculated the time necessary to extract the features used as input for the neural networks and to perform the classification process to select the best shot. As a result we got the average time of 18.3 milliseconds (standard deviation of 6.2 milliseconds), which indicates the capacity of the proposed method to selected the best shots in real-time.

The results of the recognition rate test indicate the capacity of the proposed method to learn and replicate the editing style of a human editor. The good recognition rates achieved with small training sets indicate that a human editor can train the neural networks without having to choose the best shots for too many scenes. It is important to notice that we used training and testing samples generated by the same human editor. If we test the neural networks with samples generated by another human editor that has a different editing style, the recognition rates will probably be lower. This, however, was expected because every editor has its own style and preferences. The proposed technique is capable of learning this personal editing style and replicating it during the video compositing process, which keeps the signature of the human artist in the computer generated content.

8.1.1.2. Transition Selection

The capacity of the proposed method to select the most adequate transitions for video segments was evaluated by comparing the results of the proposed method with the decisions made by human editors of well-known movies.

Firstly, we analyzed the initial scene of the movie The Lord of the Rings: The Return of the King (New Line Cinema 2003). The scene starts with *Déagol* falling into the river *Anduin* and finding the ring, and ends with *Frodo* and *Sam* following *Gollum* through the *Vale of Morgul*. The test sequence had approximately 8 minutes and a total of 94 shots manually separated into individual video files (where the frames that contain transition effects were eliminated by hand). This sequence was chosen because the scene starts in the past and gradually progresses to the present story time, enabling the evaluation of

different temporal transitions. Given the ordered sequence of shots of the movie, we computed (for each consecutive shot) the transition between the shots (see Figure 8.4) and then compared the result with the original transition used in the movie. In this case, there is no need of a story planner or video compositing algorithms, because the movie is linear and all scenes are pre-recorded. In the video-based interactive storytelling system, the temporal and spatial information used by the algorithm is automatically calculated, but in this test we manually annotated this information in each shot. The result of the test is shown in Table 8.2.

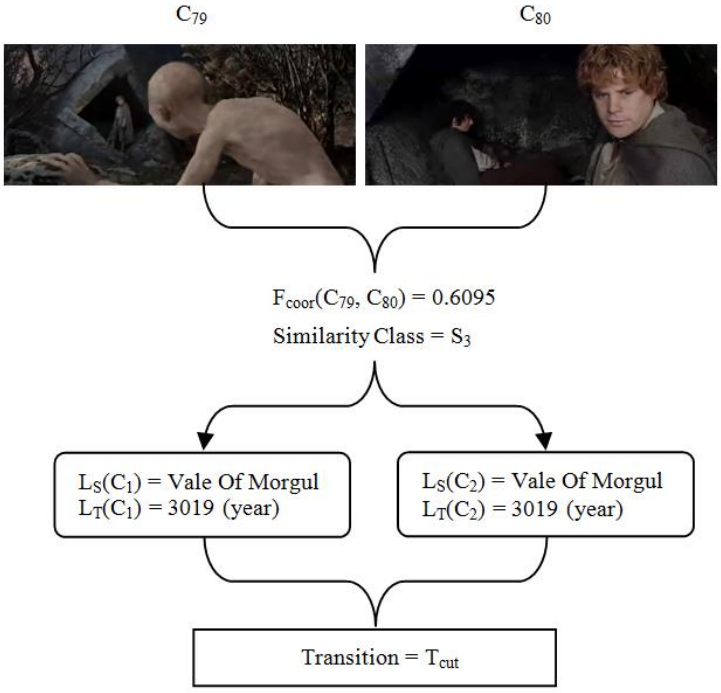


Figure 8.4: Example of a transition computation between two shots (C_{79}, C_{80}) of The Lord of the Rings: The Return of the King. Copyrighted images reproduced under “fair use” policy.

Transitions	Cut	Dissolve	Wipe	Fade
Original	86	6	0	2
Our Method	84	7	0	2
Hits Rate	97.6%	85.7%	100%	100%

Table 8.2: Comparison between the original transitions in the Lord of the Rings: The Return of the King with the transitions selected by our method.

We found two transitions that do not match the original transitions. The first one is a cut classified by the proposed method as a dissolve. Analyzing the video segments it is difficult to justify why the actual editor chose a cut, because it is clear that the shots occur in different times and the cut causes some disorientation in the audience. This disorientation does not occur using a dissolve transition. In the other mismatch, our method classified the transition as a jump cut. Indeed, visually analyzing the film we see that there is a jump cut in a very short fighting scene. Actually we cannot affirm that the human editor is wrong in these cases, because each editor has his/her own style.

As a second evaluation test, we selected a movie by another director in a different film genre: the classic Psycho, directed by Alfred Hitchcock (Universal Studios 1960). We analyzed the last scene of the film. The scene starts with *Lila* going to investigate *Mrs. Bates'* house and finishes in the end of the film, when *Mary's* car is pulled out of the swamp. The test sequence had approximately 14 minutes and a total of 153 shots manually separated in individual video files. The results of this test are shown in Table 8.3.

Transitions	**Cut**	**Dissolve**	**Wipe**	**Fade**
Original	150	2	0	1
Our Method	150	2	0	1
Hits Rate	100%	100%	100%	100%

Table 8.3: Comparison between the original transitions in Psycho with the transitions selected by our method.

The automated transition selection method does not involve complex computing tasks; however, its computing complexity grows according to the resolution of the analyzed video segments. In order to check the performance of the proposed method, we tested its execution in the most common video resolutions. For each resolution, we calculated the necessary time to compute the histograms, calculate the histogram correlation and classify the transition. This process was executed in a sequence of 40 video segments and the average time was computed for each resolution. The results of the performance tests are shown in Table 8.4.

Resolution	704x480	1280x720	1920x1080
Time(*ms*)	8.43	16.81	22.74

Table 8.4: Performance results of the transition selection method with different video resolutions.

8.1.2. Photography and Music

The validation of the methods used to select the best visual effects and soundtracks for the narratives was based on two tests: (1) a recognition rate test to check the accuracy of the predicted visual effects and soundtracks; and (2) a performance test to check the necessary time to perform the prediction process.

In order to evaluate the accuracy of the neural networks used to select the best visual effects and soundtracks for the narrative we simulated several scenes varying the type of scene, number of actors, emotional states and actions, and then, for each scene, we asked to a human director of photography and a human music director to make the selection of the visual and audio profile that best represent the scene emotion. Each decision generates one sample for both neural networks, which includes all the features used as input for the neural network, together with the identification of the selected visual effect and soundtrack for the simulated scene. Based on the samples collected, we created 5 training sets with a different number of samples and, for each one, a testing set with half the size of the corresponding training set. The training sets were used to train the neural networks and the samples of the current test set were then predicted. Table 8.5 and Figure 8.5 show the computed results of this test with the training set size ranging from 10 to 50 samples.

Number of Samples	10	20	30	40	50
Visual Effects Accuracy	89.7%	92.4%	95.9%	98.1%	98.8%
Music Accuracy	90.2%	93.1%	94.8%	97.5%	98.4%

Table 8.5: Recognition rate of the visual effects and music selection method with training sets ranging from 10 to 50 samples.

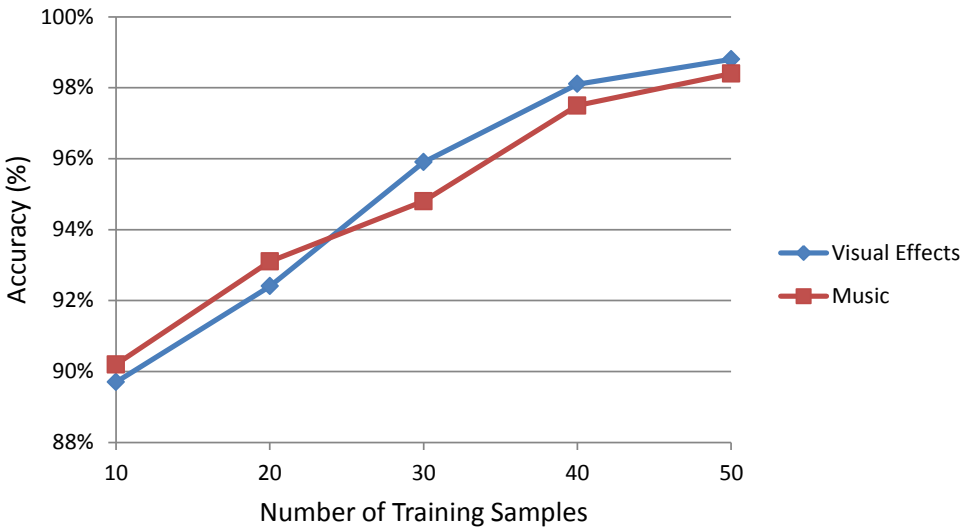


Figure 8.5: Recognition rate of the visual effects and music selection method with training sets ranging from 10 to 50 samples.

In order to evaluate the performance of the proposed solution, we used our method to select the visual effects and soundtracks for a sequence of 20 scenes and, for each one, we calculated the time necessary to extract the features used as input for the neural networks and to perform the classification process. As result we get the average time of 8.4 milliseconds (standard deviation of 4.6 milliseconds) to select the best visual effects and an average time of 7.9 milliseconds (standard deviation of 5.3 milliseconds) to select the soundtracks to the scenes, which indicates the capacity of the proposed method to perform its task in real-time.

The results of the recognition rate test are similar to the ones obtained by the neural networks trained to select the best shots for the scenes and also indicate the capacity of the proposed method to learn and replicate the decisions made by a human director of photography and music director using small training sets. This approach allows the system to learn the personal style of human filmmakers and replicate it during the dramatization of a video-based interactive narrative.

8.1.3. Frame Compositing

The frame compositing process is the most time-consuming task and must be performed in real-time to allow the system to generate video frames while the

narrative is being exhibited. The algorithm comprises several image processing methods and its complexity grows according to the number of scene elements that have to be composed in the frame. In order to evaluate the performance of the parallel architecture of the proposed frame compositing process, we conducted a performance test to check the average frame rate of the proposed system with the number of threads ranging from 1 to 8. Five sequences of 4 basic actions with an increasing number of actors were simulated and dramatized by the system, generating a total average of 600 frames per sequence. The results of the performance tests of the parallel composing architecture are shown in Figure 8.6.

The results of the performance experiments show that the process of compositing a frame becomes more expensive as more scene elements are added to the frame. However, the parallel architecture of the system can compensate the cost of the frame compositing task by dividing the work among multiple CPU cores.

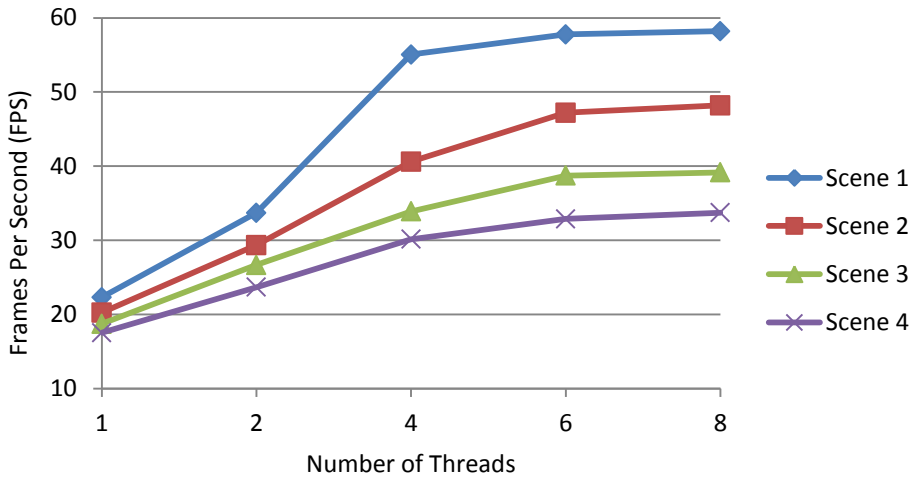


Figure 8.6: Performance results of the parallel composing architecture with the number of actors in the frame ranging from 1 to 4 and with the number of compositing threads ranging from 1 to 8.

8.1.4. Natural Language Interface

The user interaction methods adopted in the video-based interactive storytelling system are mostly based on natural language. In order to evaluate the natural language processing algorithms implemented in the user interaction

module, we performed two experiments: (1) the recognition rate test, to check the accuracy of the predicted suggestions; and (2) the performance test, to check the time needed to process the input suggestions and recognize them as first-order logic sentences. For both tests, we used a set of 107 text suggestions collected from users that were testing the interaction system. After a manual analysis of the suggestions, we found 81 suggestions that were manually classified as valid suggestions.

For the recognition rate test, we used our method to extract valid story suggestions from the text suggestions and then compared the results with the results obtained by the manual classification. As a result we got a recognition rate of 90.6%, with only 10 valid suggestions being incorrectly classified as invalid suggestions. The main reason for the incorrect classifications was the occurrence of spelling mistakes.

To evaluate the performance of our method, we again utilized the 107 suggestions collected from users, and calculated the average time necessary to recognize them as first-order logic sentences. As a result we got the average time of 2.7 milliseconds to process an input suggestion and recognize them as first-order logic sentences (standard deviation of 1.3 milliseconds).

Similarly, we evaluated the method utilized to recognize user satisfaction. During the tests of the system, we collected a set of 43 text comments expressing user satisfaction. Then, we used our simplistic method of sentiment analysis to classify the comments as positive and negative comments then compared the results with the results obtained through a manual classification. As result we get a recognition rate of 97.6%, with only 1 positive comment incorrectly classified as negative. The time consumed by the algorithm is almost insignificant (less than 0.001 milliseconds).

In the experiments, the multi-user natural language interface produced good results. However, natural language processing is not a trivial task. It is possible that our parser will not correctly recognize every possible valid sentence, but we believe that it will be able to recognize the sentences in the most part of the cases without the audience being aware of mistakes.

8.2. Visual Evaluation

The visual evaluation concerns the overall aspects of the scenes composed by the system. In order to perform this test, we conducted an experiment comparing the results automatically produced by the proposed system with the results manually produced by two teams of filmmaking professionals, where each team was composed of a film director and a video compositing professional. We selected a sequence of three basic actions and asked the human teams to compose the scene representing each of the basic actions. Then, we used our video-based dramatization system to generate the same sequence of basic actions. Both human and system had available the same video resources to compose the frames. Table 8.6 shows the selected basic actions, including the logical description used by the dramatization system and the natural language description that was given to the human subjects.

	Logical Description	**Natural Language Description**
Action 1	*GoIn*([*Anne*], [*University*])	"*Anne enters in the university where Peter is reading a book*."
Action 2	*Tell*([*Peter*], [*S17*], [*Anne*], [*Nightclub*])	"*In the nightclub, Peter asks Anne if she likes to go out to parties*."
Action 3	*Kiss*([*Peter*], [*Anne*], [*MainSquare*])	"*Peter kisses Anne in the Main Square*."

Table 8.6: Description of the selected basic actions used in the visual evaluation test.

In order to perform the task, the human subjects decided to use the Adobe After Effects CS6. The results of the visual evaluation test comparing the initial frames of the scenes composed by the human professionals and the initial frames automatically generated by the proposed video-based dramatization system for the three selected basic actions are shown in Table 8.7.

During the experiment, we also recorded the time both human and system spent to complete the tasks. Table 8.8 shows a comparison of the time spent by the subjects to complete the composition of a single frame of each basic action.

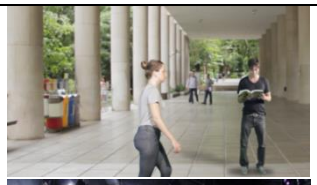
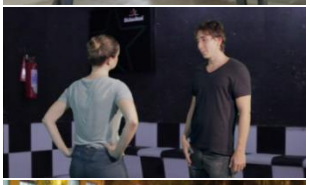
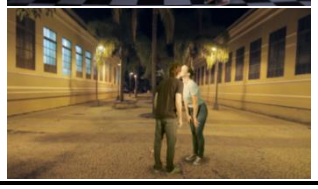
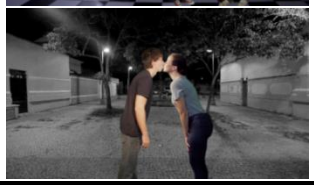
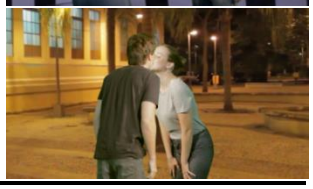
	Team 01	Team 02	System
Action 1			
Action 2			
Action 3			

Table 8.7: Visual comparison between the selected frames of the scenes composed by the human subjects and the corresponding frames automatically generated by the proposed video-based dramatization system for the three basic actions.

	Team 01	Team 02	System
Action 1	36.20 (*min*)	29.16 (*min*)	3.55 (*sec*)
Action 2	50.47 (*min*)	26.22 (*min*)	2.42 (*sec*)
Action 3	20.16 (*min*)	39.17 (*min*)	2.04 (*sec*)

Table 8.8: Comparison between the times spent by the human professionals and the system to compose the scenes representing the three basic actions.

Although the scenes automatically generated by the system have different angles and distances from those specified by the human teams, the results are similar in quality in a way that it would be difficult to a human to identify which ones was generated by a computer software. These similarities indicate the capacity of the proposed automatic video compositing methods to generate frames similarly as video compositing professionals do. In addition, the system is capable of generating multiple frames instantaneously, while the human professional takes several minutes to composite the frames.

9
Conclusion

This thesis proposed a new approach to video-based interactive narratives that uses video compositing techniques to dynamically create video sequences representing the story events – rather than using only prerecorded scenes. In our method, actors are filmed by several cameras in front of a green screen in a variety of emotional states and situations that are compatible with the logical structure of narratives. Afterwards, an automatic process controls the cinematography language, compositing scenarios and choosing view parameters (zoom, angle and camera movements). This approach allows the generation of more diversified stories, increases interactivity, and reduces production costs. This chapter presents the conclusions remarks, summarizes the contributions, points some limitation of our approach and suggests topics for future research work.

9.1.
Concluding Remarks

This thesis explored video-based interactive narratives from three different points of view: authors, developers and users. For authors, we presented a general guide on how to write and film interactive stories, and also developed some computational tools to help them in the production process. For developers, we proposed an architecture for video-based interactive storytelling systems and presented the technical details about the implementation of the real-time video compositing and editing algorithms. For users, we designed attractive and engaging interaction mechanisms.

In any form of storytelling, the author is the key component for a successful story. However, authoring for interactive storytelling is a difficult task. It involves the process of logically specifying the context of the story, thinking about possible events in terms of parameters, preconditions and effects, which are tasks that typical story writers are not familiar with. In addition, the production team (e.g.

artists, cinematographers, actors) must produce the visual content (e.g. 3D models, videos, 2D animations) to represent the narrative in accordance with the logical specification of the story and considering all possible storylines that may be created to comply with the user's desires. Our initial thought about the use of videos to dramatize interactive narratives was that it would simplify and reduce the production work in comparison with the artistic efforts necessary to produce 3D models and animations for a 3D dramatization. However, along the development of this work, we have realized that filming and editing the video material necessary for a video-based dramatization require as much authorial work as producing a 3D/2D interactive narrative. Obviously it involves different professionals – while a 3D interactive narrative requires designers and artists, a video-based dramatization requires filmmakers and actors.

From the technical point of view, developing a video-based dramatization system required completely new algorithms and techniques for video compositing and virtual cinematography. Traditional 3D/2D dramatization systems adopt many of the techniques used in games and simulations to create the story worlds and control the virtual characters, animations and cameras. On the other hand, video-based systems cannot make use of most of those techniques due to limitations imposed by the video resources (e.g. lack of freedom to show characters from any angle, non-parameterized actions and movements, immutable video sequences). In order to overcome some of those limitations, we proposed filming the actors and locations from multiple angles in front of a green screen, which allowed us to create specialized algorithms for positioning actors and cameras, selecting the best shots to film the scenes, simulating camera movements, and compositing all scene elements into a single piece of motion picture in real-time. The results of the technical evaluation tests demonstrated the efficiency and applicability of the proposed video compositing and editing algorithms for video-based interactive storytelling.

Since the beginning of this research, television and cinema were the main target mediums of the proposed video-based interactive storytelling system. The advance of interactive storytelling technology to these new mediums required new interaction mechanisms to support the multi-user characteristic of these platforms. The proposed user interaction interfaces (social networks and mobile devices) provided the basic multi-user setting required for both mediums. In addition, they

provided an engaging way for users to interfere in the narratives through natural language. Although few experiments had been conducted on applying these mechanisms on real environments of television and cinema, the small scale experiments confirmed their applicability and efficacy in providing an engaging user interaction interface that support multi-user interactions. We believe that video-based interactive storytelling may be the first step towards the emergence of real interactive films, which can expand the boundaries of traditional branching interactive films towards a new form of digital entertainment.

The next sections present more details about the contribution of this thesis, its limitations and some directions for future research.

9.2. Contributions

The main contributions of this thesis are:

- *Video-based interactive storytelling using video compositing techniques*. We proposed a new method to represent video-based interactive narratives using real-time video compositing and editing techniques. Previous works on video-based interactive storytelling are all based on static and immutable pre-recorded video sequences that are rearranged during presentation, which reduce interactivity, story diversity, and increase the productions costs. The proposed method dynamically generates video sequences representing the story events in real-time, which provides the system with the possibility of generating more diversified stories without increasing production costs.
- *Interactive video narratives based on cinematography principles and techniques*. The proposed techniques for the generation of video-based interactive narratives follow cinematography principles and rules to create attractive and engaging visual representations for the story events. The architecture of our system is composed of a set of cinematography-based autonomous agents that share the responsibility for creating dynamic video sequences respecting cinematography rules. Previous works on video-based interactive narratives focus mainly on the creation of stories by

ordering video segments, without taking into account cinematography concepts.

- *Video-based interactive storytelling and robust story generation algorithms*. The proposed video-based interactive storytelling system is integrated with a robust story generation system based on planning under nondeterminism and capable of generating complex and diversified interactive story plots. Most of the previous works on video-based interactive storytelling, especially the interactive films produced for TV and Cinema, are based on rudimentary branching narrative structures, which simplify the use of videos for the representation of the story (all possible events are predefined and can be pre-recorded), but compromise the diversity of stories and the user's sense of agency. By integrating real-time video-compositing techniques with robust automated story generation algorithms, these limitations can be overcome and video-based interactive narratives with real interactive and dynamic plots can be created.
- *Real-time video compositing algorithm*. We proposed a parallel frame compositing algorithm capable of managing and compositing multiple video frames simultaneously to guarantee real-time performance. The algorithm was evaluated through a performance test, which demonstrates that compositing a frame becomes more expensive as more scene elements are added to the frame. However, the parallel architecture of the proposed algorithm can compensate the expensiveness of the frame compositing task by dividing the work among multiple CPU cores. The algorithm can also be used in other applications that require some form of automated video compositing process.
- *Automated method for shot selection using expert cinematography knowledge*. We proposed a method to select the best camera shots to show the generated scenes of video-based interactive narratives. Our approach consists of representing the knowledge of a real film editor using several artificial neural networks trained to solve cinematography problems involving camera shot selection. The proposed technique is capable of learning the personal editing style of human editors and replicating it during the video compositing process, which keeps the signature of the human artist in the computer generated content. This method is based on

our previous work that used support vector machines (SVM) to select the camera angles in a 3D environment (Lima et al. 2010).

- *Automated method for the selection of scene transitions based on cinematography theory*. We proposed a method to guarantee the temporal and spatial continuity of video-based interactive narratives by avoiding jump cuts and selecting the most adequate shot transition for the narrative scenes. Our approach consists in translating cinematography principles and practices directly into logical rules. The method was evaluated by comparing the results of the proposed method with the decisions made by human editors of well-known movies. The results indicate that our method is capable of selecting scene transitions as professional human editors in most of the cases.
- *Multi-user natural language interface for interactive storytelling using mobile devices*. We proposed a new multi-user interface that allows users to freely interact with virtual characters by text or speech using mobile devices. The interaction mechanism was designed to support both global and local user interactions. By using the proposed method, users are able to write or speak what they want to happen in the story, or easily select the desired outcome for local decision points. Most previous works on interaction methods for interactive storytelling focus mainly on single-user interactions.
- *Multi-user interaction though social networks*. We explored the use of social networks as a multi-user interface. We presented and evaluated an interaction method that allows users to interact and change stories through social networks (such as Facebook, Twitter and Google+). This method allows users to collaborate with the development of interactive stories in a social environment through their own social network clients, using smartphones, tablets, or personal computers without having to install any additional software. The activity that results from the user interactions in the social network may attract more viewers to the broadcasting channel (increasing the audience). In addition, viewers can make new friends through the interaction in the social network. As far as we are aware, this is the first time this form of interaction is explored in an interactive narrative.

- *General guide and computational tools for the production of video-based interactive narratives*. We presented a general guide on how to write and film interactive narratives, and proposed some computational tools developed to assist the production of video-based interactive narratives. The author is one of the most important components in any form of storytelling, especially in video-based interactive storytelling, where he/she has to specify the logical context of the story and cinematographers have to film and edit the videos resources necessary for the dramatization of the interactive story. As far as we are aware, this is the first research work to explore the concepts of authoring in video-based interactive storytelling.

9.3. Publications and Awards

The results of the research on video-based interactive storytelling were published in leading conferences in the field of multimedia and interactive storytelling. The real-time video editing method that automatically generates the most adequate shot transitions, avoids jump cuts, and creates looping scenes, was published in the International Conference on Multimedia and Expo (Lima et al. 2012A). In addition, more papers on this matter are being prepared for submission to journals of multimedia and computer entertainment.

The research on user interaction methods also has originated some publications: the social interaction method for interactive storytelling was published in the International Conference on Entertainment Computing (Lima et al. 2012B); the multi-user natural language interface using mobile devices was published in the Brazilian Symposium on Computer Games and Digital Entertainment (Lima et al. 2012C); and another paper describing a study on multimodal, multi-user and adaptive interaction methods was also published in the Brazilian Symposium on Computer Games and Digital Entertainment (Lima et al. 2011B). The results of the research on story dramatization also have originated some publications: a paper exploring the use of an augmented reality visualization interface combined with a sketch-based interaction interface was published in the International Conference on Entertainment Computing (Lima et al. 2011A), and in

the journal of Entertainment Computing (Lima et al. 2014A). In addition, another paper presenting a system capable of generating dynamic interactive narratives in the format of comic books was published in the International Conference on Advances in Computer Entertainment Technology (Lima et al. 2013).

The research that led to this thesis also received two international awards from the International Telecommunication Union (ITU).[7] The first award is an honorable mention on "*Innovation*" in the "*1nd ITU IPTV Application Challenge*" competition (2011), with the video-based interactive narrative called "*The Princess Kidnapping*"; and the second award is an honorable mention on "*Interactivity*" in the "*2nd ITU IPTV Application Challenge*" competition (2012), with the comic-based interactive narrative called "*Little Gray Planet*". Both interactive narratives were designed for interactive TV. The ITU is the United Nations specialized agency for information and communication technologies.

9.4. Limitations and Directions for Future Research

Although the proposed approach to create video-based interactive narratives has achieved the primary objectives of this thesis, we also identified some limitations and directions for future research, which can be categorized into three main topics: image quality, authoring process, and evaluation experiments.

The image quality of the results produced by our system is still far from the excellent visual quality of feature films. Image quality depends on real-time techniques for realistic lighting, which relight actors with the proper illumination of the environment, and consider cast shadows and interreflection. Interactive real-time video rendering with complex illumination and materials is still an open issue even in the multimedia research area. A future work would be to explore the existent dynamic lighting techniques and verify the possibility of applying them in the real-time video compositing process for interactive storytelling. Examples of promising approaches include the use of techniques for capturing the actor's live-action performance illuminating him with a sequence of time-multiplexed basis lighting conditions (Wenger et al. 2005; Chabert et al. 2006), and the use of

[7] ITU - http://www.itu.int/

interactive ray tracing techniques in the compositing process (Pomi and Slusallek 2005).

The second main limitation of our approach is related with the amount of authorial work during the production and post-production phases. The proposed method to generate video-based interactive narratives is entirely based on the use of video resources filmed from different angles, which gives to the system the freedom to dramatize scenes applying the basic cinematography concepts during the dramatization of the narrative. However, filming the actors performing their actions from 8 different angles generates a huge number of video files, which grows according to the number of actions the characters can perform during the narrative. The process of editing and removing the background of all these videos using a traditional chroma key matting technique requires a huge amount of work in the post-production phase, which increases the production costs. An alternative to overcome this limitation could be the adoption of a more efficient and automated matting technique, such as a hardware-based solution (Joshi et al. 2006; Sun et al. 2006). These solutions may also improve the current visual quality of the compositing results, which suffers from color spills produced by the green screen background in the actors.

Another factor that increases the amount of work during the production and post-production phases is the existence of replaceable accessories or clothes in the characters, which will require the same actions to be filmed several times varying the accessories/clothes. A possible solution to this problem would be the inclusion of the dynamic and replaceable objects in the scenes during the compositing process using a tracking procedure to correctly sync the object movements with the actor movements. For example, if a character needs to hold different weapons during the narrative, he could be filmed holding a generic object with distinct tracking markers that would be tracked during the compositing process to identify the correct position to place any weapon in the character hands.

Another limitation of this thesis is the lack of large-scale user experiments to validate the usability of the proposed video-based system from a Human-Computer Interaction (HCI) perspective. An interesting experiment would be a comparative study between a video-based interactive narrative and a 3D/2D version of the same story. In this direction, the IRIS Evaluation Toolkit (Klimmt

et al. 2010; Roth et al. 2009) provides a good methodology to evaluate and compare the general users' experience provided by both dramatization modalities.

The visual quality of the video sequences produced by the proposed video compositing algorithms also has to be evaluated in more precise studies. An interesting experiment to complement the visual evaluation presented in this thesis would be a Turing Test applied to the generated video sequences in order to evaluate if human subjects are able to differentiate the scenes created by the compositing algorithms and the scenes created by the filmmaking professionals.

The interactive film production process described in this thesis and the authoring tasks for video-based interactive storytelling also need to be better evaluated. An interesting future work would be a deeper study about this process from the perspective of the people involved in the authoring tasks, which may provide a more detailed feedback about the problems and possible solutions to improve the process and the video-based interactive storytelling system in general. In addition, a more precise evaluation of the production costs is also necessary. In this direction, another interesting future work would be a comparative study of the costs for producing 3D interactive narratives, video-based interactive narratives using video compositing techniques, and video-based interactive narratives using only static video segments.

The present system was built based on the third version of the Logtell system, which incorporates the basic temporal modal logic of the first version (Ciarlini et al. 2005), the client/server architecture of the second version (Camanho et al. 2009), and planning under nondeterminism (Silva et al. 2010) combined with the use of nondeterministic automata to control the dramatization of events (Doria et al. 2008) that where introduced in the third version of the Logtell. Much work remains to be done towards the integration of the proposed dramatization system and user interaction interfaces with the recent advances in the Logtell Project, such as the stream-based architecture for delivering interactive narratives in multiple platforms (Camanho et al. 2013), the new non-deterministic planning model using dramatic properties of the story events (Gottin 2013; Ferreira 2013), and the incorporation of information-gathering events in the story plots (Silva et al. 2012).

References

Aamodt, A., Plaza, E., 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. Artificial Intelligence Communications, 7 (1), pp. 39-52.

Adams, E., 1999. The Designer's Notebook: Three Problems for Interactive Storytellers. Gamasutra. Available at: http://www.gamasutra.com/view/feature/131821/the_designers_notebook_three_.php [Accessed June 23, 2014].

ADIVI, 2014. Add digital information to video. Product homepage, InnoTeamS. Available at: http://www.adivi.net/ [Accessed June 23, 2014].

Ahanger, G., Little, T.D.C., 1997. A System for Customized News Delivery from Video Archives. In Proceedings of International Conference on Multimedia Computing and Systems, Ottawa, Canada, pp. 526-533.

Ahanger, G., Little, T.D.C., 1998. Automatic composition techniques for video production. IEEE Transactions on Knowledge and Data Engineering, 10 (6), pp. 967-987

Arens, Y., 1986. CLUSTER: An approach to Conceptual Language Understanding. Ph.D. Thesis, University of California at Berkeley, USA.

Arijon, D., 1976. Grammar of the Film Language. Silman-James Press, Los Angeles, USA.

Aronson, D., 2006. DV Filmmaking: From Start to Finish. O'Reilly Media, Sebastopol, USA.

ArtOfVFX, 2013. Game Of Thrones (Season 3): Doug Campbell – VFX Supervisor – Spin VFX. Available at: http://www.artofvfx.com/?p=4915 [Accessed June 23, 2014].

Aylett, R., Louchart, S., Dias, J., Paiva, A., 2005. FearNot! An Experiment in Emergent Narrative. In Proceedings of the 5th International Conference on Intelligent Virtual Agents, Springer Press, pp. 305-316.

Barbash, I., Taylor, L., 1997. Cross-cultural Filmmaking: A Handbook for Making Documentary and Ethnographic Films and Videos, University of California Press, New York, USA.

Barber, H., Kudenko, D., 2007. Dynamic Generation of Dilemma-based Interactive Narratives. In Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference. AAAI Press: Menlo Park, California, n. 6.

Bates, J., 1992. The Nature of Character in Interactive Worlds and The Oz Project. Technical Report, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA. Available at: https://www.cs.cmu.edu/afs/cs/project/oz/web/papers/loeffler.ps [Accessed June 23, 2014].

Bates, J., 1994. The role of emotion in believable agents. Communications of the ACM, 37(7), pp. 122-125.

Batini, C., Ceri, S., Navathe, S.B., 1992. Conceptual Database Design : an Entity Relationship Approach. Addison-Wesley, Boston, USA.

Bejan, B., 1992. I'm Your Man, Sony Pictures Entertainment/Loews Theatres.

Bocconi, S., 2009. Vox Populi: generating video documentaries from semantically annotated media repositories. Ph.D. Thesis, Technische Universiteit Eindhoven, Eindhoven, Netherlands.

Bowen, C.J., Thompson, R., 2009. Grammar of the Edit. Second Edition, Focal Press, Oxford, USA.

Bradski, G., Kaehler, A., 2008. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, Sebastopol, USA.

Bringsjord, S., Ferrucci, D.A., 1999. Artificial Intelligence and Literary Creativity. Inside the Mind of BRUTUS, a Storytelling Machine. Psychology Press, United Kingdom.

Brown, B., 2011. Cinematography: Theory and Practice: Image Making for Cinematographers and Directors. Focal Press, Waltham, USA.

Brown, M., Lowe, D., 2007. Automatic Panoramic Image Stitching Using

Invariant Features. International Journal of Computer Vision, 74 (1), pp. 59-77.

Butler, J.G., 2002. Television: critical methods and applications. Lawrence Erlbaum Associates Publishers, New Jersey, USA.

Cai, Y., Miao, C., Tan, A.H., Shen, Z., 2007. A Hybrid of Plot-Based and Character-Based Interactive Storytelling. In Proceedings of the 2nd International Conference of ELearning and Games. Hong Kong, p. 260-273.

Caldwell, T., 2011. The Film Analysis Handbook. Insight Publications, Cheltenham, Australia.

Camanho, M., Feijó, B., Furtado, A., Pozzer, C., Ciarlini, A., 2013. A Model for Stream-based Interactive Storytelling as a New Form of Massive Digital Entertainment. In XII Brazilian Symposium on Games and Digital Entertainment, São Paulo, Brazil, pp. 109-117.

Camanho, M.M., Ciarlini, A.E.M., Furtado, A.L., Pozzer, C.T., FEIJÓ, B., 2009. A Model for Interactive TV Storytelling. In VIII Brazilian Symposium on Digital Games and Entertainment, Rio de Janeiro, Brazil, pp. 197-206.

Cavazza, M., Charles, F., Mead, S., 2002. Character-based interactive storytelling. IEEE Intelligent systems, 17 (4), pp. 17-24.

Cavazza, M., Charles, F., Mead, S.J., Martin, O., Marichal, X., Nandi, A., 2004. Multimodal acting in mixed reality interactive storytelling. IEEE Multimedia, 11 (3), pp. 30-39.

Cavazza, M., Lugrin, J-L., Pizzi, D., Charles, F., 2007. Madame bovary on the holodeck: immersive interactive storytelling. In Proceedings of the 15th International Conference on Multimedia. pp. 651-660.

Cavazza, M., Pizzi, D., Charles, F., Vogt, T., André, E., 2009. Emotional Input for Character-based Interactive Storytelling. In Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, Budapest, Hungary, pp. 313-320.

Chabert, C.F., Einarsson, P., Jones, A., Lamond, B., Ma, W.C., Sylwan, S., Hawkins, T., Debevec, P., 2006. Relighting human locomotion with flowed reflectance fields. In ACM SIGGRAPH 2006 Sketches, 76, New York.

Charles, F., Cavazza, M., 2004. Exploring the Scalability of Character-based Storytelling. In Third ACM Joint Conference on Autonomous Agents and Multi-Agent Systems. New York, pp. 872-879.

Charles, F., Lugrin, J., Cavazza, M., Mead, S., 2002. Real-time camera control for interactive storytelling. In Proceedings of the Game-On, London, United Kingdom.

Cheshire, T., Burton, C., 2010. Transmedia: Entertainment reimagined, Wired UK. Available at: http://www.wired.co.uk/magazine/archive/2010/08/features/what-is-transmedia [Accessed June 23, 2014].

Chua T.S., Ruan, L.Q., 1995. A Video Retrieval and Sequencing System. ACM Trans. Information Systems, 13 (4), pp. 373-407.

Ciarlini, A.E.M., Furtado, A.L., 2002. Understanding and Simulating Narratives in the Context of Information Systems. In 21st International Conference on Conceptual Modeling Tampere, Finland, pp. 291-306.

Ciarlini, A.E.M., Pozzer, C.T., Furtado, A.L., Feijó, B., 2005. A logic-based tool for interactive generation and dramatization of stories. In Proceedings of the International Conference on Advances in Computer Entertainment Technology, Valencia, Spain, p. 133-140.

Činčera, R., Roháč, J., Svitáček, V., 1967. Kinoautomat: One Man and His House, Czechoslovakia.

Costikyan, G., 2000. Where Stories End and Games Begin. Game Developer, Sept., pp. 44-53.

Crawford, C., 2004. Chris Crawford on Interactive Storytelling, New Riders, New Jersey.

Davenport, G., Murtaugh, M., 1995. ConText Towards the Evolving Documentary. In Proceedings of the third ACM international conference on Multimedia, San Francisco, pp. 377-389.

Davis, R., 2010. Complete Guide to Film Scoring. Berklee Press Publications, Boston, USA.

Donikian, S., 2003. DraMachina: an authoring tool to write interactive fictions. In

First International Conference on Technologies for Interactive Digital Storytelling and Entertainment, Darmstadt, Germany, pp. 101-112.

Doria, T.R., Ciarlini, A.E.M., Andreatta, A.A., 2008. Nondeterministic Model for Controlling the Dramatization of Interactive Stories. In Proceedings of the 2nd ACM Workshop on Story Representation, Mechanism and Context. Vancouver, Canada, p. 21-26.

Dow, S., Mehta, M., Lausier, A., MacIntyre, B., Mateas, M., 2006. Initial Lessons from AR-Façade, An Interactive Augmented Reality Drama. In: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology. n. 28.

Dragon's Lair, 1983. Advanced Microcomputer Systems.

Ekman, P., Friesen, W.V., 1971. Constants across cultures in the face and emotion. Journal of Personality and Social Psychology, 17, pp. 124-129.

El-Nasr, M.S., 2007. Interaction, Narrative, and Drama Creating an Adaptive Interactive Narrative using Performance Arts Theories. Interaction Studies, 8 (2), pp. 209-240.

Ferreira, P.A., 2013. Modelo de planejamento temporal não determinístico considerando propriedades dramáticas em mudança contínua para storytelling interativo. M.Sc. Thesis, Departamento de Informática Aplicada, Universidade Federal do Estado do Rio de Janeiro, Brazil.

Ferreira, R.J.M., 2012. New Zealand Film Music in Focus: Music by New Zealand Composers for Feature Films. University of Auckland, New Zealand.

Fikes, R., Nilsson, N., 1971. STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence, 2, pp. 189-208.

Foster, J., 2010. The Green Screen Handbook: Real-World Production Techniques. Wiley: Sybex Press, New York, USA.

Furtado, A.L., Ciarlini, A.E.M., 2000. Generating Narratives from Plots using Schema Information. In 5th International Conference on Applications of Natural Language to Information Systems, Versalhes, France, pp. 17-29.

Gabrielsson, A., Lindström, E., 2001. The Influence of Musical Structure on

Emotional Expression. Oxford University Press, New York, USA.

Gastal, E.S.L., Oliveira, M.M., Shared sampling for real-time alpha matting. Computer Graphics Forum, 29 (2), pp. 575-584.

Ghallab, M., Nau, D., Traverso, P., 2004. Automated Planning: Theory and Practice. Morgan Kaufmann Publishers, San Francisco.

Gilroy, S., Porteous, J., Charles, F., and Cavazza, M., 2012. Exploring Passive User Interaction for Adaptive Narratives. In Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces. pp. 119-128.

Gottin, V.M., 2013. Verificação abstrata de propriedades dramáticas contínuas em eventos não determinístico. M.Sc. Thesis, Departamento de Informática Aplicada, Universidade Federal do Estado do Rio de Janeiro, Brazil.

Grasbon, D., Braun, N., 2011. A morphological approach to interactive storytelling. In Proceedings of CAST01 Living in Mixed Realities. Special issue of Netzspannung.org/journal, the Magazine for Media Production and Intermedia Research, Sankt Augustin, Germany, pp. 337-340.

Hales, C., 2005. Cinematic interaction: From kinoautomat to cause and effect. Digital Creativity 16 (1), pp. 54-64.

Hassoun, M.H., 1995. Fundamentals of Artificial Neural Networks. The MIT Press, Cambridge, USA.

Hawkins, B., 2005. Real-Time Cinematography for Games. Charles River Media, Hingham, USA.

Heavy Rain, 2010. Quantic Dream, Sony Computer Entertainment.

Hornung, A., 2003. Autonomous Real-Time Camera Agents in Interactive Narratives and Games. M.Sc. Thesis, Department of Computer Science, Aachen University of Technology, Germany.

Hughes, J.F., Dam, A.V., McGuire, M., Sklar, D.F., Foley, J.D., Feiner, S.K., Akeley, K., 2013. Computer Graphics: Principles and Practice. 3rd Edition, Addison-Wesley Professional, Boston, USA.

ITU-T, 2013. Advanced video coding for generic audiovisual services, Series H:

Audiovisual And Multimedia Systems Infrastructure of audiovisual services – Coding of moving video. Available at: http://www.itu.int/rec/T-REC-H.264 [Accessed June 23, 2014].

Jenkins, H., 2003. Game Design as Narrative Architecture. In First Person: New Media as Story, Performance, and Game. Harrigan P., Wardrip-Fruin, N. (eds.), MIT Press, Cambridge.

Jensen, J.F., 1988. Adventures in Computerville: Games, Inter-Action & High Tech Paranoia i Arkadia. Kultur & Klasse 63, Copenhagen: Medusa.

Joshi, N., Matusik, W., Avidan, S., 2006. Natural video matting using camera arrays. ACM Transactions on Graphics, 25, pp. 779-786.

Jull, J., 1998. A Clash between Game and Narrative. M.Sc. Thesis, University of Copenhagen. Available at: http://www.jesperjuul.dk/thesis [Accessed June 23, 2014].

Jung Von Matt/Spree, 2010. Last Call, Berlin. Available at: https://www.youtube.com/watch?v=qe9CiKnrS1w [Accessed June 23, 2014].

Jurafsky, D., Martin, J.H., 2000. Speech and language processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice-Hall, New Jersey, USA.

Karlsson, B.F.F., 2010. A model and an interactive system for plot composition and adaptation, based on plan recognition and plan generation. Ph.D. Thesis, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Brazil.

Katz, S., 1991. Film Directing Shot by Shot: Visualizing from Concept to Screen. Michael Wiese, Los Angeles, USA.

Kenworthy, C., 2009. Master Shots: 100 Advanced Camera Techniques to Get an Expensive Look on Your Low-Budget Movie. Michael Wiese Productions, Los Angeles, USA.

Klein, S., Aeschlimann, J.F., Balsiger, D.F., Coverse, S.L., Court, C., Forster, M., Lao, R., Oakley, J.D., Smith, J., 1973. Automatic Novel Writing: A Status Report. Technical Report 186, Computer Sciences Department, University of

Wisconsin. Available at: http://minds.wisconsin.edu/handle/1793/57816 [Accessed June 23, 2014].

Klimmt, C., Roth, C., Vermeulen, I., Vorderer, P., 2010. The Empirical Assessment of The User Experience In Interactive Storytelling: Construct Validation Of Candidate Evaluation Measures. Technical Report, Integrating Research in Interactive Storytelling - IRIS. Available at: http://ec.europa.eu/information_society/apps/projects/logos/4/231824/080/deliverables/001_IRISNoEWP7DeliverableD72.pdf [Accessed June 23, 2014].

Kneafsey, J., 2006. Virtual Cinematography for Computer Games. M.Sc. Thesis, Institute of Technology Blanchardstown, Dublin, Ireland.

Kodak, 2007. The Essential Reference Guide for Filmmakers. Eastman Kodak Company, Rochester, USA.

Kuka, D., Elias, O., Martins, R., Lindinger, C., Pramböck, A., Jalsovec, A. Maresch, P., Hört-Ner, H., Brandl, P., 2009. DEEP SPACE: High Resolution VR Platform for Multi-user Interactive Narratives. In Proceedings of the 2nd Joint International Conference on Interactive Digital Storytelling. pp. 185-196.

Lanier, L., 2009. Professional Digital Compositing: Essential Tools and Techniques. Wiley: Sybex Press, New York, USA.

Lebowitz, M., 1984. Creating characters in a story-telling universe. Poetics, 13 (3).

Lebowitz, M., 1985. Story-Telling as planning and learning. Poetics, 14 (6).

Lee, M.G., 1994. A model of story generation. M.Sc. Thesis. University of Manchester, United Kingdom.

Lima, E.S., 2010. Um Modelo De Dramatização Baseado Em Agentes Cinematográficos Autônomos Para Storytelling Interativo. M.Sc. Thesis, Departamento de Computação Aplicada, Universidade Federal de Santa Maria, Santa Maria, Brazil.

Lima, E.S., Feijó, B., 2014. Video-Based Interactive Storytelling: Film Production, Technical Report TR-3/14, ICAD/VisionLab, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Brazil.

Available at: http://www.icad.puc-rio.br/publications [Accessed June 23, 2014]

Lima, E.S., Feijó, B., Barbosa, S., Silva, F.G., Furtado, A.L., Pozzer, C.T., Ciarlini, A.E.M., 2011B. Multimodal, Multi-User and Adaptive Interaction for Interactive Storytelling Applications. In: Proceedings of the 10th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames). Salvador, Brazil, pp. 206-214.

Lima, E.S., Feijo, B., Barbosa, S.D.J., Furtado, A.L., Ciarlini, A.E.M., Pozzer, C.T., 2014A. Draw Your Own Story: Paper and Pencil Interactive Storytelling. Entertainment Computing, 5 (1), p. 33-41.

Lima, E.S., Feijó, B., Barbosa, S.D.J., Furtado, A.L., Ciarlini, A.E.M., Pozzer, C.T., 2011A. Draw Your Own Story: Paper and Pencil Interactive Storytelling. In Proceedings of the 10th International Conference on Entertainment Computing. pp. 1-12.

Lima, E.S., Feijo, B., Furtado, A.L., Barbosa, S.D.J., Pozzer, C.T., Ciarlini, A.E.M., 2013. Non-Branching Interactive Comics. In: Proceedings of the International Conference on Advances in Computer Entertainment Technology (ACE 2013), Enschede, Netherlands, pp. 230-245.

Lima, E.S., Feijó, B., Furtado, A.L., Pozzer, C., Ciarlini, A., 2012A. Automatic Video Editing For Video-Based Interactive Storytelling. In Proceedings of the 2012 IEEE International Conference on Multimedia and Expo (ICME), Melbourne, Australia, pp. 806-811.

Lima, E.S., Feijó, B., Furtado, A.L., Pozzer, C.T., Ciarlini, A.E.M., Silva, F.G., 2012C. A Multi-User Natural Language Interface for Interactive Storytelling in TV and Cinema. In Proceedings of the XI Brazilian Symposium on Computer Games and Digital Entertainment, Brasília, Brazil, pp. 154-161.

Lima, E.S., Feijó, B., Pozzer, C.T., Ciarlini, A.E.M., Barbosa, S.D.J., Furtado, A. L., Silva, F.G.A., 2012B. Social Interaction for Interactive Storytelling. In Proceedings of the 11th International Conference on Entertainment Computing Bremen, Germany, pp. 1-15.

Lima, E.S., Pozzer, C.T., Feijo, B., Ciarlini, A.E.M., Furtado, A.L., 2010. Director of Photography and Music Director for Interactive Storytelling. In IX Brazilian

Symposium on Games and Digital Entertainment, Florianopolis, Brazil. pp. 122-131.

Lima, E.S., Pozzer, C.T., Ornellas, M., Ciarlini, A., Feijó, B., Furtado, A., 2009. Virtual Cinematography Director for Interactive Storytelling. In Proceedings of the International Conference on Advances in Computer Entertainment Technology, Greece, pp. 263-270.

Liu, B., 2010. Sentiment Analysis and Subjectivity. Handbook of Natural Language Processing, Second Edition, Chapman and Hall/CRC, UK.

Lobrutto, V., 2002. The Filmmaker's Guide To Production Design. Allworth Press, New York, USA.

Loyall, A.B., 1997. Believable Agents: Building Interactive Personalities, Ph.D. Thesis. School of Computer Science, Carnegie Mellon University, Pittsburgh, USA.

Loyall, A.B., Bates, J., 1991. Hap – A Reactive, Adaptive Architecture for Agents. Technical Report CMU-CS-91-147, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. Available at: http://www.cs.cmu.edu/Groups/oz/papers/CMU-CS-91-147.ps [Accessed June 23, 2014]

Loyall, A.B., Bates, J., 1993. Real-time Control of Animated Broad Agents. In Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society, Boulder, Colorado.

Ma, Y.F., Lu, L., Zhang, H.J., Li, M.J., 2002. A user attention model for video summarization. In Proceedings of the 10th ACM international conference on multimedia, Juan-les-Pins, France, pp. 533-542.

Magerko, B., 2005. Story representation and interactive drama. In Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment (AIIDE). Marina del Rey.

Magerko, B., Laird, J., 2003. Building an interactive drama architecture. In Proceedings of the First International Conference on Technologies for Interactive Digital Storytelling and Entertainment. Darmstadt, Germany, pp. 226-237.

Magerko, B.S., 2006. Player Modeling in the Interactive Drama Architecture. Ph.D. Thesis, University of Michigan, Ann Arbor, USA.

Manovich, L., 2001. The Language of New Media. The MIT Press, Cambridge, USA.

Marneffe, M., Manning, C.D., 2008. The Stanford typed dependencies representation. In Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation, Manchester, pp. 1-8

Mascelli, J., 1965. The Five C's of Cinematography: Motion Picture Filming Techniques. Silman-James Press, Los Angeles, USA.

Mateas, M., 2002. Interactive Drama, Art, and Artificial Intelligence. Ph.D. Thesis. School of Computer Science, Carnegie Mellon University, Pittsburgh, USA.

Mateas, M., Stern, A., 2003. Façade: An Experiment in Building a Fully-Realized Interactive Drama. In Game Developers Conference, Game Design track. pp. 4-8.

Mateas, M., Vanouse, P., Domike, S., 2000. Generation of Ideologically-Biased Historical Documentaries. In Proceedings of the Seventeenth National Conference on Artificial intelligence and Twelfth Conference on innovative Applications of Artificial intelligence. AAAI Press, pp. 236-242.

McGuire, M., Matusik, W., Pfister, H., Hughes, J.F., Durand, F., 2005. Defocus video matting. ACM Transactions on Graphics, 24 (3), pp. 567-576.

Medler, B., Magerko, B., 2006. Scribe: A General Tool for Authoring Interactive Drama. In 3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment, Darmstadt, Germany, pp. 139-150.

Meehan, J., 1977. TALE-SPIN, an interactive program that writes stories. In Proceedings of the Fifth Interactional Joint Conference on Artificial Intelligence, pp. 91-98.

Meehan, J., 1981. TALE-SPIN. In Inside Computer Understanding: Five Programs Plus Miniatures, Schank, R., Riesbeck, C. (eds.), Lawrence Erlbaum Associates, Hillsdale, USA, pp. 197-226.

Miller, W., 1997. Screenwriting for Narrative Film and Television. Pearson Press, New Jersey, USA.

Millington, I., Funge, J., 2009. Artificial Intelligence for Games. 2nd Edition, Morgan Kaufmann, Burlington, USA.

Mitchell, T., 1997. Machine Learning, McGraw–Hill Science/Engineering/Math, New York, USA.

Mott, B.W., Lester, J.C., 2006. U-director: a decision-theoretic narrative planning architecture for storytelling environments. In Proceedings of the Fifth international Joint Conference on Autonomous Agents and Multiagent Systems. Hakodate, Japan, pp. 977-984.

Müller, W., Spierling, U., Stockhausen, C., 2013. Production and Delivery of Interactive Narratives Based on Video Snippets. In Proceedings of the 6th International Conference on Interactive Digital Storytelling, Istanbul, Turkey, pp. 71-82.

Nack, F. Parkes, A. 1997. The Application of Video Semantics and Theme Representation in Automated Video Editing. Multimedia Tools and Applications, 4 (1), pp. 57-83.

Nakasone, A., Ishizuka. M., 2007. ISRST: An Interest based Storytelling Model using Rhetorical Relations. In Proceedings of the Edutainment 2007, Hong Kong, China, pp. 324-335.

New Line Cinema, 2003. The Lord of the Rings: The Return of the King. New Line Cinema.

Newman, R., 2008. Cinematic Game Secrets for Creative Directors and Producers. Focal Press, Oxford, UK.

Nissen, S., 2003. Implementation of a Fast Artificial Neural Network Library (FANN), Technical report, Department of Computer Science University of Copenhagen (DIKU), Copenhagen, Denmark.

O'Brien, M., Sibley, N., 1995. The Photographic Eye: Learning to See with a Camera. Sterling, Worcester, USA.

Okun, J., Zwerman, S., 2010. The VES handbook of visual effects: industry

standard VFX practices and procedures. Focal Press, Boston, USA.

OpenCV, 2014. About – OpenCV. Available at: http://opencv.org/about.html [Accessed June 23, 2014].

Packard, E., 1979. The Mystery of Chimney Rock, Choose Your Own Adventure #5, Bantam Books, New York.

Paiva, A., Machado, I., Prada, R., 2001. Heroes, villains, magicians, ... Dramatis personae in a virtual story creation environment. In Proceedings of the 6th international conference on Intelligent user interfaces, pp. 129-136.

Pellinen, T., 2000. Akvaario (Aquarium), Media Lab, Helsinki University of Art and Design Finland. Broadcast by The Finnish Broadcasting Company.

Pérez y Pérez, R. Sharples, M., 2001. MEXICA: A computer model of a cognitive account of creative writing. Journal of Experimental and Theoretical Artificial Intelligence. 13, pp. 119-139.

Pérez y Pérez, R., 1999. MEXICA: a Computer Model of Creativity in Writing. Ph.D. Thesis, University of Sussex, United Kingdom.

Piacenza, A., Guerrini, F., Adami, N., Leonardi, R., Julie Porteous, Jonathan Teutenberg, Marc Cavazza. 2011. Generating Story Variants with Constrained Video Recombination. In Proceedings of the 19th ACM International Conference on Multimedia 2011, Scottsdale, AZ, USA, Nov.-Dec. 2011.

Pizzi, D., Cavazza, M., 2007. Affective Storytelling Based on Characters' Feelings. In AAAI Fall Symposium on Intelligent Narrative Technologies, Arlington, Virginia.

Pizzi, D., Cavazza, M., 2008. From Debugging to Authoring: Adapting Productivity Tools to Narrative Content Description. In: Proceedings of the 1st Joint International Conference on Interactive Digital Storytelling, Erfurt, Germany, pp. 285-296.

Pomi, A., Slusallek, P., 2005. Interactive Ray Tracing for Virtual TV Studio Applications. Journal of Virtual Reality and Broadcasting, 2 (1).

Porteous, J., Benini, S., Canini, L., Charles, F., Cavazza, M., Leonardi, R., 2010. Interactive storytelling via video content recombination. In Proceedings of the

International Conference on Multimedia 2010, Firenze, Italy, pp. 1715-1718.

Pozzer, C.T., 2005. Um Sistema para Geração, Interação e Visualização 3D de Histórias para TV Interativa. Ph.D. Thesis, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Brazil.

Prada, R., Machado, I., Paiva, A., 2000. TEATRIX: Virtual Environment for Story Creation. Proceedings of the 5th International Conference of Intelligent Tutoring Systems. Springer Verlag, pp. 464-473.

Propp, V., 1968. Morphology of the Folktale. Laurence Scott (trans.), University of Texas Press, Austin.

Pulli, K., Baksheev, A., Kornyakov, K., Eruhimov, V., 2012. Real-Time Computer Vision with OpenCV. Communications of the ACM, 55 (6), pp. 61-69.

Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D. Manning, C., 2010. A Multi-Pass Sieve for Coreference Resolution. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Boston, pp. 492-501.

Reeves, J., 1989. The Rhapsody Phrasal Parser and Generator. Technical Report, Computer Science Department, University of California. Available at: http://ftp.cs.ucla.edu/tech-report/198_-reports/890064.pdf [Accessed June 23, 2014].

Riedl, M., Rowe, P., David, K.E., 2008. Toward Intelligent Support of Authoring Machinima Media Content: Story and Visualization. In: 2nd International Conference on Intelligent Technologies for Interactive Entertainment, n 4.

Roth, C., Vorderer, P. Klimmt, C., 2009. The Motivational Appeal of Interactive Storytelling: Towards a Dimensional Model of the User Experience. In Proceedings of the 2nd Joint International Conference on Interactive Digital Storytelling: Interactive Storytelling. pp. 38-43.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning internal representations by error propagation. Parallel Distributed Processing, 1, pp. 318-362.

Russell, S. Norvig, P., 2010. Artificial Intelligence: A Modern Approach. Third edition, Prentice Hall, New Jersey, USA.

Samsel, J., Wimberley, D., 1998. Writing for Interactive Media: The Complete Guide. Allworth Press, New York.

Sawhney, N., Balcom, D., Smith, I., 1996. HyperCafe: narrative and aesthetic properties of hypervideo. In Proceedings of the Seventh ACM Conference on Hypertext, pp. 1-10.

Sawicki, M., 2011. Filming the Fantastic: A Guide to Visual Effects Cinematography. Focal Press, Waltham, USA.

Shen, Y.T., Lieberman, H., Davenport G., 2009. What's Next? Emergent Storytelling from Video Collections. In Proceeding of International Conference on Human factors in computing systems, ACM Press, pp. 809-818.

Si, M., Marsella, S.C., Riedl, M., 2008. Integrating Story-centric and character-centric designs: An mixed-initiative framework for interactive drama. In Proceedings of the 4th Conference on Artificial Intelligence and Interactive Digital Entertainment. Palo Alto, California, pp. 203-208.

Sikov, E., 2009. Film Studies: An Introduction. Film and Culture Series edition, Columbia University Press, New York, USA.

Silva, F.G.A., 2010. Geração de enredos com planejamento não-determinístico em storytelling para TV interativa. M.Sc. Thesis, Departamento de Informática Aplicada, Universidade Federal do Estado do Rio de Janeiro, Brazil.

Silva, F.G.A., Ciarlini, A.E.M, Siqueira, S.W.M., 2010. Nondeterministic Planning for Generating Interactive Plots. In 12th Ibero-American Conference on AI, Bahía Blanca, Argentina, Springer, pp. 133-143.

Silva, F.G.A., Furtado, A.L., Ciarlini, A.E.M, Pozzer, C.T., Feijó, B., Lima, E.S., 2012. Information-gathering Events in Story Plots. In Proceedings of the 11th International Conference on Entertainment Computing, Bremen, Germany, pp. 30-44.

Space Ace, 1984. Advanced Microcomputer Systems.

Spierling, U., Braun, N., Iurgel, I., Grasbon, D., 2002. Setting the scene: playing

digital director in interactive storytelling and creation. Computers & Graphics, 26, (1), pp. 31-44.

Spierling, U., Weiß, S.A., Müller, W., 2006. Towards Accessible Authoring Tools for Interactive Storytelling. In 3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment, Darmstadt, Germany, pp. 169-180.

Stanford, 2014. The Stanford NLP (Natural Language Processing) Group. Available at: http://nlp.stanford.e du/software/lex-parser.shtml [Accessed June 23, 2014].

Sullivan, K., Alexander, K., Schumer, G., 2008. Ideas for the Animated Short: Finding and Building Stories. Focal Press, Boston, USA.

Sun, J., Jia, J., Tang, C.-K., Shum, H.-Y., 2004. Poisson matting. ACM Transactions on Graphics, 23 (3), pp. 315-321.

Sun, J., Li, Y., Kang, S.B., Shum, H.-Y., 2006. Flash matting. ACM Transactions on Graphics, 25 (3), 772–778.

Swain, D.V., Swain, J.R., 1988. Film Scriptwriting: A Practical Manual. Second Edition, Focal Press, Boston, USA.

Swartjes, I., Theune, M., 2009. Iterative Authoring Using Story Generation Feedback: Debugging or Co-creation? In: Second Joint International Conference on Interactive Digital Storytelling, Guimarães, Portugal, pp. 62-73.

Szilas, N., 1999. Interactive Drama on Computer: Beyond Linear Narrative. In Proceedings of the AAAI Fall Symposium on Narrative Intelligence, North Falmouth, AAAI Press.

Szilas, N., 2003. IDtension: a narrative engine for Interactive Drama. In First International Conference on Technologies for Interactive Digital Storytelling and Entertainment, Darmstadt, Germany, pp. 187-203.

Szilas, N., 2007. A Computational Model of an Intelligent Narrator for Interactive Narratives. Applied Artificial Intelligence, 21 (8), p. 753-801.

Thompson, R., Bowen, C., 2009. Grammar of the Shot. Second Edition, Focal Press, Oxford, USA.

Thue, D., Bulitko, V., Spetch, M., Wasylishen, E., 2007. Interactive storytelling: A player modelling approach. In The Third Conference on Artificial Intelligence and Interactive Digital Entertainment, Stanford, USA, pp. 43-48.

Turner, S., 1992. MINSTREL: a computer model of creativity and storytelling. Ph.D. Thesis, Computer Science Department, University of California, USA.

Universal Studios, 1960. Psycho, Universal Studios.

Ursu, M.F., Kegel, I.C., Williams, D., Thomas, M., Mayer, H., Zsombori, V., Uomola, M.L., Larsson, H., Wyver, J., 2008. ShapeShifting TV: Interactive screen media narratives. Multimedia Systems, 14, no. 2, pp. 115-132.

VideoClix, 2014. Product homepage, VideoClix Technologies. Available at: http://www.videoclix.tv/ [Accessed June 23, 2014].

Wang, J. Cohen, M., 2007. Image and video matting: a survey. Foundations and Trends in Computer Graphics and Vision, 3 (2), pp. 97-175.

Wenger, A., Gardner, A., Tchou, C., Unger, J., Hawkins, T., Debevec, P., 2005. Performance Relighting and Reflectance Transformation with Time-Multiplexed Illumination. ACM Transactions on Graphics, 24 (3), pp. 756-764.

Williams D., Ursu M.F., Cook, J.J., Zsombori, V., Engler, M., and Kegel, I.C, 2006. ShapeShifter TV: Enabling Multi-Sequential Narrative Productions for Delivery over Broadband. In the Proceedings of the 2nd Institution of Engineering and Technology Multimedia Conference 2006: From IT to HD" London, UK, pp. 29-30.

Williams, D., Ursu, M.F., Meenowa, J., Cesar, P., Kegel, I., Bergström, K., 2001. Video mediated social interaction between groups: System requirements and technology challenges. Telematics and Informatics, 28, pp. 251-270.

Wolf, M.J.P., 2007. The Video Game Explosion: A History from PONG to PlayStation and Beyond. Greenwood Publishing Group, Westport, USA.

Wright, S., 2010. Digital Compositing for Film and Video. Focal Press, Waltham, USA.

Young, M., 2001. An Overview of the Mimesis Architecture: Integrating Intelligent Narrative Control into an Existing Gaming Environment. In The

Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, Stanford, California, pp. 77-81.

Zettl, H., 2012. Video Basics. Cengage Learning Press, Stamford, USA.

Zhao, H., 2012. Emotion-driven interactive storytelling. Ph.D. Thesis, Media School, Bournemouth University, United Kingdom.

Zhou, Z., Cheok, A.D., Tedjokusumo, J., Omer, G.S., 2008. wIzQubesTM – A Novel Tangible Interface for Interactive Storytelling in Mixed Reality. International Journal of Virtual Reality, 7 (4), pp. 9-15.