NORTHWESTERN
UNIVERSITY

Computer Science Department

**Technical Report**
**NWU-CS-03-18**
**April 30, 2003**

# Automatic Expressive Lighting for Interactive Scenes

**Magy Seif El-Nasr**

## Abstract

Advances in computer graphics have led to the development of interactive entertainment applications with complex 3D graphical environments. Lighting is becoming increasingly important in such applications not only because of its role in illumination, but also because of its utility in directing the viewer's gaze to important areas and evoking moods. However, lighting design is a complex process, and is especially problematic for interactive applications. Rendering time for such applications is linear in the number of lights used, thus restricting the number of lights used in real-time rendering engines to 8 or fewer; yet current practice in the animation industry is to use 32 lights or more per scene. Moreover, the scene's spatial configuration, mood, dramatic intensity, and the relative importance of different characters, all change unpredictably in real time, necessitating continual *re*design as the characters and camera move. Current systems, however, use fixed, manually designed lighting. This manual design is labor intensive and leads to partially invisible characters, unmotivated or distracting color changes, and frustration for the player/audience. In this dissertation, I present a new approach to lighting design for interactive scenes. I will describe this approach using the Expressive Lighting Engine (ELE) that I have developed. ELE uses non-linear constraint optimization to automatically and unobtrusively adjust lighting in real-time, utilizing traditional cinematic and theatrical lighting design theory. This approach accommodates variations in spatial and dramatic configurations that occur during interactive scenes. ELE also allows artists to author lighting changes and override its decisions. ELE accommodates artistic desires, but also maintains style and visual continuity within an interactive scene.

**Keywords:**

Lighting, Interactive Drama, Interactive Entertainment, Agents, Lighting Design, Games, Visual design

**NORTHWESTERN UNIVERSITY**

# AUTOMATIC EXPRESSIVE LIGHTING FOR INTERACTIVE SCENES

**A DISSERTATION**

**SUBMITTED TO THE GRADUATE SCHOOL IN PARTIAL FULFILLMENT OF THE REQUIREMENTS**

for the degree

**DOCTOR OF PHILOSOPHY**

**Field of Computer Science**

By

**Magy Seif El-Nasr**

**EVANSTON, ILLINOIS**

June 2003

# ABSTRACT


## AUTOMATIC EXPRESSIVE LIGHTING FOR INTERACTIVE SCENES

### MAGY SEIF EL-NASR

Advances in computer graphics have led to the development of interactive entertainment applications with complex 3D graphical environments. Lighting is becoming increasingly important in such applications not only because of its role in illumination, but also because of its utility in directing the viewer's gaze to important areas and evoking moods. However, lighting design is a complex process, and is especially problematic for interactive applications. Rendering time for such applications is linear in the number of lights used, thus restricting the number of lights used in real-time rendering engines to 8 or fewer; yet current practice in the animation industry is to use 32 lights or more per scene. Moreover, the scene's spatial configuration, mood, dramatic intensity, and the relative importance of different characters, all change unpredictably in real time, necessitating continual *re*design as the characters and camera move. Current systems, however, use fixed, manually designed lighting. This manual design is labor intensive and leads to partially invisible characters, unmotivated or distracting color changes, and frustration for the player/audience. In this dissertation, I present a new approach to lighting design for interactive scenes. I will describe this approach using the Expressive Lighting Engine (ELE) that I have developed. ELE uses

non-linear constraint optimization to automatically and unobtrusively adjust lighting in real-time, utilizing traditional cinematic and theatrical lighting design theory. This approach accommodates variations in spatial and dramatic configurations that occur during interactive scenes. ELE also allows artists to author lighting changes and override its decisions. ELE accommodates artistic desires, but also maintains style and visual continuity within an interactive scene.

# ACKNOWLEDGEMENTS

Six years ago, I emailed Ian Horswill asking him for references on affective computing. Ian referred me to Andrew Ortony, who was extremely helpful and very friendly. A year later I met Andrew and Ian at a conference, and a year after that I was at Northwestern University working on interactive entertainment.

I would like to thank Ian and Andrew for asking me to apply to Northwestern. I would like to thank Ian for standing by me and believing in me all these years. I would like to thank him for his help and support, and for the endless hours that he spent with me working on the math part of my dissertation. I would like thank him for being such a wonderful and great advisor. I would like to thank Andrew for being extremely supportive of my ideas and my work, and for being a wonderful father figure to me.

I would like to thank Mary Poole for being a wonderful faculty, theatre advisor, and friend. I would like to thank her for all the time she spent with me brainstorming on ideas, techniques, and theories. I would like to thank her for teaching me and nurturing my artistic side. She helped me through my dissertation work from an idea to the complete dissertation.

I would like to thank Annette Barbier whose genuine support and belief in my work have been the extra motivation that I needed to finish my dissertation in four years. She has been a wonderful film advisor and dissertation committee member. She was always there when I needed advice.

I would like to thank Jack Tumblin, who has been very supportive of my work and research and who have helped me immensely with my writing. I would like to thank him for his commitment in making me a better writer. Because of his help, I now get reviews saying ‘a very well written paper.’ I would like to thank him for always being there when I needed him.

I would also like to thank my committee members Ken Fobus and Kris Hammond whose help, comments, and advice helped stimulate more ideas and future research directions. I would also like to thank Ben for being very supportive and a very cool guy, who answered my questions and was always there when I needed him.

Last, but never least, I would like to thank my brother, my parents, and my friends. Without their support and help, I would have never made it.

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

> Words, silences, sound effects, background music, facial expressions, gestures, movements across the stage, lighting, groupings, shadows, shapes, and colors in the costume and décor – all these are telling us something. … Our awareness of what [words] mean is inseparable from our awareness of vocal timber, tone, timing, inflexion and atmosphere.
>
> R. Hayman, *How to Read a Play*

Current advances in computer graphics, particularly in real-time rendering, make it possible to produce visually interesting and engaging images in real-time. Many game designers script lighting and camera effects using real-time rendering systems to produce dramatic effects similar to what is seen in movies. Filmmakers and animators compose visual images that support and shape the narrative and the dramatic action presented. Similarly, interactive entertainment should produce visual images that adapt to the narrative content and dramatic action. However, adapting the visual presentation of an interactive scene to accommodate variations in the narrative and action is a daunting problem.

In interactive entertainment, scripting visual presentation, including changes in lighting or camera angle and position, is difficult, because design parameters, such as character placement and dramatic intensity, cannot be anticipated at design time. In an interactive scene, the participant[1] is free to move about in the scene, which affects camera angle and position. Characters' locations and orientations change depending on the participant's position and orientation, thus they cannot be determined in advance. Finally, the participant's choices and actions affect his/her relationship with other characters, thereby affecting the story, characters' behaviors (including orientation and position), and dramatic intensity.

Game designers approach this problem by limiting the changes afforded by interaction. Most games limit character interactions to a scripted linear, or a small branching, narrative. In these cases, camera and lighting effects are statically configured depending on the scripted linear dialogue and action.

My central interest is visual presentation for interactive entertainment, which includes considerations of many elements, including framing, camera movement, placements of characters, and lighting. This dissertation focuses on lighting. It is important to note that all the elements listed above interact with one another, thus lighting cannot be explored in isolation. The dynamic relationship between lighting and other elements are considered and discussed in this dissertation.

Expressive lighting is an important element of interactive entertainment. I define expressive lighting as a lighting design that emphasizes the narrative and its dramatic

[1] I use the term participant to denote the player who interacts within the interactive scene.

content. It is especially important because it not only affects the participant's mood, but action as well. Unlike in movies, participants in an interactive scene have agency, they can act and react to the perceived visuals. Therefore, in interactive scenes, visibility of objects, their color, and contrast play a major role in affecting participants' actions, as well as engagement and mood. Carson (00) describes examples of such qualities and several techniques borrowed from theme park to describe the impact of such elements on interaction.

Game designers and interactive entertainment researchers (Mateas and Stern 00, Young 00) rely on a static lighting solution, such as lighting based on set design[2] or ambient lighting[3]. Ambient lighting is used in games such as *The Sims*. Ambient lighting ensures fast rendering and visibility of all objects and characters, but it has little or no aesthetic and dramatic function. Set-based lighting, or static lighting in general, is insensitive to the variations in the spatial configuration and dramatic intensity. Set-based lighting is successful only if the designer can anticipate all possible situations (characters' orientations, camera angles, and variations in dramatic tension). Since this is impossible in interactive scenes, current interactive entertainment systems suffer from major distractions, such as inappropriate colors or partially lit characters.

To resolve these limitations, I have developed a lighting system called ELE (Expressive Lighting Engine). ELE is designed to automatically adjust lighting in real-

---

[2] Lighting design based on set design denotes a design where lights are positioned to simulate the effect of practical sources (light sources that are implied by the set, such as torches, windows, lamps, and doors) present in the scenery or the set.

[3] The term 'ambient lighting' is used here to indicate a design where every object in the scene is lit equally using a constant intensity, and where the light has no direction. Thus, the objects are given constant luminance values (Möller and Haines 99).

time based on the narrative and the dramatic situation. To facilitate this process, I have adapted lighting design principles from film and theatre, and represented them formally within ELE. Recognizing lighting to be a complex art, ELE is also designed to allow artists to override its decisions by authoring rules that adopt a particular setting for some controllable high-level lighting parameters, discussed in Chapter 4.

## 1.1 Motivation and Goals

The main motivation of the work described here is to make expressive lighting achievable in interactive entertainment by adapting lighting techniques used in film and theatre. The success of such techniques in film and theatre suggests a possible profound impact on interactive entertainment, and thus the work described here may facilitate engagement and enable artists to develop interactive techniques that were not possible using current methods.

Additionally, ELE can potentially reduce production costs for interactive and non-interactive media, such as film, theatre, and animation. ELE is designed to make quick and automatic lighting design decisions. Therefore, it can be used as a rapid prototyping tool for theatre and film lighting design. Designers and writers of interactive and non-interactive productions may use ELE to quickly render their scenes without investing time and money in constructing a lighting design for each scene, and thus can concentrate their efforts on other aspects of scene design and production.

## 1.2 Lighting Design

Film, theatre, and animation lighting designers have established some requirements for effective lighting design (Calahan 96, Brown 96, Lowell 92, Viera 93, and Birn 00), such as ensuring visibility or paralleling dramatic tension. To achieve these requirements, designers have developed some conventions or rules. However, the lighting design process, as any design process, involves many trade-offs. For instance, ensuring visibility of all characters in a scene may violate visual tension and mood.

Lighting design is therefore an optimization problem. The lighting designer identifies opposing demands, weighs different options, and chooses a lighting configuration that best satisfies the dramatic situation and the lighting style. Lighting designers often establish a lighting *style*, and then continually adjust the lighting between shots to accommodate the movement and dramatic content while maintaining that style. I use the term style in this dissertation to refer to the set of priorities that a designer assigns to the lighting design requirements. For example, a realistic style demands that the designer assigns high priority to establishing visual continuity and ensuring that the lighting angle conforms to the direction of light emitted by practical sources, while expressionistic style denotes high priority to portraying dramatic intensity and mood.

## 1.3 System Outline

ELE uses constraint-based non-linear optimization to select the number of lights used, their placement, color[4], and angles to parallel dramatic intensity, emphasize action, and satisfy desired lighting design goals. ELE is composed of three subsystems designed to: (1) dynamically allocate lights to areas within the scene, (2) select angles for each light in the scene, and (3) select colors for each light in the scene.

[4] Color is used in this dissertation to denote hue, saturation, and lightness (brightness) properties of color, not just chroma.

Depending on the camera position, the focus of the scene, and the number of characters in the scene, ELE first calculates the visible area. It then divides the visible area into several areas: an area for each character or group of characters, foreground areas, and background areas. The differentiation between foreground and background areas is important to simulate the perception of depth. ELE computes a character area surrounding each character to allow fine control of lighting for each character in the scene differentiating between focus and non-focus characters. ELE then allocates some lights to each area.

Dynamically allocating lights to areas is an important problem. Rendering time is proportional to the number of lights used. To achieve 30 frames per second rendering speed, real-time rendering engines impose an upper limit on the number of lights used (e.g. *Wildtangent*[5] limits the number of dynamic lights to 8). However, to achieve depth, mood, and dramatic intensity, lighting design often requires as many as thirty-two lights per scene (— 02). Thus, often fewer lights are available when more are needed. The problem is therefore an optimization problem where the system allocates several lights to each area, such that the sum of allocated lights is less than or equal to the maximum number of lights that can be used to maintain the speed needed for real-time rendering. To solve this problem, I have developed a greedy algorithm that iterates through lighting configurations evaluating them in terms of the desired lighting design goals considering the dramatic situation.

Once lights are allocated to areas, ELE selects two angles for each light: azimuth and elevation angle. ELE uses hill climbing to select an azimuth angle that suits characters'

[5] *Wildtangent* is a real-time rendering engine developed for games and is publicly available.

and camera's orientations, and lighting design parameters specifying the importance of visibility, mood, motivation (conforming to direction of light emitted from the local light sources in the scene), and visual continuity. ELE uses a simple rule-based method to choose an elevation angle given the importance of mood and motivation.

In addition to selecting angles, ELE selects colors for each light in the scene. Since the color of one light interacts with colors of other lights in the scene, and since color palates are typically constrained by the style established, ELE uses a non-linear constraint-based optimization algorithm (gradient decent with a boundary function) to select a color for each light in the scene that conforms to the dramatic focus (important characters or objects that signify the focal point of the scene), dramatic tension, and lighting design parameters specifying the ideal values for and importance of depth, contrast, lightness, hue, saturation, and warmth. ELE then computes light placements and relays color, position, and angle information to the rendering engine.

## 1.4 Contributions and Results

Each component of ELE was tested in isolation. The whole system was then integrated and tested within five scenes from the interactive story *Mirage* (introduced in Appendix B). ELE was able to automatically light the interactive scenes while maintaining visual continuity and style. ELE was also able to meet artists' specified constraints.

The work described here contributes to the research in interactive entertainment in four ways:

- **Adapting Film and Theatre Lighting Design Theory.** It formalizes cinematic and theatrical lighting design theory and adapts it to function in interactive entertainment. Cinematic and theatrical lighting design provides a deep source of theories and practices whose utility and aesthetics have already proven their success.

- **Automating Lighting Design in real-time.** The dissertation describes a system which automatically selects a lighting configuration in real-time to accommodate the evolving dramatic situation, style, and the lighting design requirements.
- **Adequately adjusting the lighting while maintaining style.** ELE focuses on adjusting the lighting design automatically while also maintaining visual continuity and the designer's choice of style. For example, if the designer chose to parallel dramatic tension in terms of warm/cool color contrast, then ELE will continue to accommodate that style while the dramatic tension increases or decreases.
- **Retaining Artistic Control:** Lighting is a complex artistic process. Although the system adjusts lighting automatically, artists may need to intervene to select a specific lighting effect. Thus, the system is designed to allow artists to override its decisions at any time.

Authoring lighting changes for an interactive scene is problematic, because at design time artists have no knowledge of the visual configuration details used in the previous frame, including character or camera placements, light colors, or light angles, and thus artists have no frame of reference when authoring lighting changes for an interactive scene. ELE supplies artists a set of sufficiently high-level parameters that allow artistic control of lighting without the knowledge of specific details describing the visual layout used in the previous frame, such as colors, angles, or physical placements of lights, objects, or the camera. ELE also allows artists to author changes as rules that are fired given a specific story state. ELE then maintains the style and adjusts the lighting from thereafter using the settings and style established by the artist.

## 1.5 Dissertation Organization

The remainder of this dissertation is divided into seven chapters, as follows:

- **Chapter 2: Limitations of Current Lighting Techniques for Interactive Entertainment.** This chapter describes the current lighting techniques used in interactive entertainment research and productions. It discusses the major limitations and advantages of these approaches using screenshots or scenarios from video games and other interactive entertainment projects.

- **Chapter 3: Expressive Lighting – Looking at Traditional Lighting Design.** This chapter describes the lighting design requirements. It also enumerates the conventions used by film and theatre lighting designers.

- **Chapter 4: Controlling Lighting using High-level Parameters.** As mentioned above, lighting control for interactive scenes necessitates the development of a set of sufficiently high-level parameters that allow control of lighting without the knowledge of spatial and visual configuration of the previous frame. This chapter discusses these parameters, while also describing previous efforts in developing artistic lighting tools that hide rendering details. In addition, it also introduces the concept of virtual light sources, and existing methods for manipulating and calculating their effects in real-time rendering engines.

- **Chapter 5: ELE – Expressive Lighting Engine.** This chapter outlines the lighting system. It describes a set of mathematically formulated rules based on guidelines described in cinematic and theatrical lighting design theory, discussed in Chapter 3.

- **Chapter 6: Implementation and Results.** To implement ELE and test it within an interactive story, I have developed and implemented an interactive story engine, a camera system, and a character system, as well as ELE. This chapter describes these systems to give the reader a complete understanding of the implementation of the whole system. It also shows results of components of ELE tested separately, as well as, screenshots of ELE working within some scenes from *Mirage* – an interactive story based on the Greek tragedy, *Electra*.

- **Chapter 7: Applications and Future Directions.**

- **Chapter 8: Summary and Conclusion.**

# CHAPTER 2

# LIMITATIONS OF CURRENT LIGHTING TECHNIQUES FOR INTERACTIVE ENTERTAINMENT

Most visual techniques used in interactive entertainment are static and context-insensitive; they do not accommodate variations in spatial configuration, mood, dramatic intensity, or importance of characters and objects in a scene. Current systems often use static lighting where lights are manually positioned depending on positions of practical sources – a technique that does not support the narrative or accommodate the varying dramatic intensity and action.

## 2.1 Ambient Lighting

Ambient lighting is a lighting method in which every object in the scene is lit evenly using a constant intensity, and where the light has no direction. Thus, the objects are given constant luminance values (Möller and Haines 99). This is a fast and simple model of lighting in which all objects are equally visible.

It, however, has major drawbacks. This type of lighting is unrealistic, and fulfils no dramatic or aesthetic functions (Birn 00). Since the luminance values are fixed for all objects, it is impossible to direct viewer's attention to important artifacts or portray dramatic intensity using color contrast. The technique is more suited for toy-like interactive environments such as *The Sims* and *Sim City*, where dramatic development is not the main focus of the design.

## 2.2 Set-Based Lighting

Set-based lighting is used in most first person shooters and adventure games, such as *Max Payne*, *Quake*, *Half-life*, and *Devil May Cry*. Level (or set) designers determine static positions, orientations, and colors for each light depending on positions of practical sources in the scene. They most often use non-interactive rendering algorithms to generate light maps that provide the lighting required for the scene (Maattaa 02, Carson 00).

While this scheme provides a workable solution for most games, it is inflexible. First, it cannot be used to provide character modeling (a technique where more lights are added and directed towards the character to establish depth and texture). Since set-based lighting depends on the architecture rather than the characters, modeling is not guaranteed, and is frequently not accomplished.

Second, in some situations the character may not be sufficiently lit; a problem that surfaces in many games, such as *The Thing* and *Half Life*. Figure 2.1 illustrates this, showing a screenshot from *Max Payne*, where Max is only partially visible due to his unanticipated movement.

**Figure 2.1 A Screenshot from *Max Payne***

While this effect can be considered dramatic (i.e. mysterious or moody), in interactive entertainment these effects should be controlled, because they hinder visibility. Visibility is important because it conveys information, such as facial expressions or identity. In addition, lack of visibility may cause frustration and diminish engagement. Thus, sometimes dramatic effects need to be sacrificed for visibility.

One major drawback of static lighting is its inability to provide a method for adapting the lighting design to direct viewer's attention to the dramatic focus. The screenshot shown in figure 2.1, for example, shows a set-based lighting where the image seems to suggest that the table or lamp is the dramatic focus of the scene instead of Max. In fast pace games, such as soccer, directing viewer's eyes (especially for inexperienced players) to the dramatic focus is crucial to the success of the game. Additionally, in training simulations, lighting can be employed to visually guide the trainee through a process or a task by directing his/her attention to important areas in the scene.

Furthermore, set-based lighting, in addition to being very time consuming and tedious, cannot accommodate the continuous variation in dramatic tension, since colors and angles cannot be changed during interaction. Filmmakers change color contrast to establish visual intensity, which is set to parallel the dramatic intensity of the scene. To accommodate variations in dramatic tension, game designers often break the tension into discrete points and readjust the lighting for each level as the tension increases or decreases. For example, in the game *Devil May Cry* there are twenty-two missions. To differentiate between the missions and to give the player a sense of dramatic tension, each mission is painted with a color that is perceptually warmer than the preceding mission. The last mission is painted with a very distinct saturated red color – possibly signifying danger and intensifying fear and tension.

**Figure 2.2 A Screenshot from *Blade of Darkness***

## 2.3 Dynamic Lighting

Some games adopt a different scheme for lighting – dynamic lighting based on practical sources in the scene. Lighting is calculated in real-time during interaction based on positions of practical sources, such as torches or lanterns. While this technique can produce effective mood, it has several drawbacks. Since the participant controls the practical sources, the designer cannot control the lighting to ensure that appropriate visual focus, visual intensity, and mood is provided and emphasized in the scene. Also, visibility is problematic; figure 2.2 shows this problem. In the figure, the participant (whose back is towards us) is fighting another character (facing us) who is barely visible, and in this case the participant cannot determine if his/her sword touched the other character or not, which frustrates him/her and risks decreased engagement.

## 2.4 Lighting for Emotions

Tomlinson developed a system that changes light colors and camera movements to present the user with an interpretation of the world based on the characters' emotions (Tomlinson 99). The system selects camera movements and light colors to show traits or feelings of each character in the scene. For example, the system uses low camera angles to show that a character is powerful or harsh red light to make a character look demonic.

He categorized lights as global lights (lights lighting the architecture and the environment) and personal lights. Global lights have a default scheme. They are fixed, and are mainly used to provide the key source of illumination and maintain visual continuity. Personal lights are fixed on characters. They follow characters around; their colors depend on the emotional state and feelings of the characters lit.

**Figure 2.3 Figures from *Swamped*, an Interactive Entertainment Created by Tomlinson**

Tomlinson did not address lighting design goals as discussed by film and theatre. In his work, Tomlinson restricted the function of light to portraying the emotional states, feelings, or traits of each character rather than providing a coherent mood or visual focus. Figure 2.3 shows two screenshots from an interactive narrative called *Swamped*, created by Tomlinson. The images show red and blue lights projected on the characters to signify frustration and sadness, respectively. In contrast, film lighting designers often adjust positions and colors of all lights in the scene, including what Tomlinson refers to as global lights, to accommodate camera angle, movement, dramatic intensity, dramatic focus, and mood.

# CHAPTER 3

# EXPRESSIVE LIGHTING –

## LOOKING AT TRADITIONAL LIGHTING DESIGN

> Every light has a job to do, every light must fit and balance within the overall shot, every light interacts with others and with the action. They all work together in a web of complexity.
>
> – Brown 96

## 3.1 Expressive Lighting – Goals of Lighting Design

B. Foss (92) defined multiple functions for narrative events and their filmic presentation. I am extending these functions to define lighting design goals, and the use of lighting as a medium of visual expression in interactive entertainment. I distinguish between four types of lighting design goals: *realistic*, *lyrical*, *dramatic*, and *aesthetic*.

A lighting design achieves realism by conforming to a realistic color palette, conservatively changing light colors to preserve visual continuity, and selecting angles of light that adhere to the direction of practical sources. These decisions increase the credibility of the scene and help the audience identify with and relate to the scene.

A lighting design achieves lyrical goals when designed to evoke moods or emotions from the audience. Lighting designers use several perceptual rules to adjust colors and angles of lights to achieve a desired mood (Alton 95, Campell 99). For instance, lighting designers vary the degree of visibility of a character's face to affect the audience's emotions and feelings since it is known that less visible faces elicit uneasiness (Gillette 98). Additionally, lighting designers increase the amount of darkness in an image to elicit fear and a sense of mystery (Gillette 98, Alton 95).

Lighting is often configured to serve various dramatic goals, including emphasize dramatic tension, attract viewer's attention to important objects or characters, and provide good visibility for the action and characters within the scene. A scene typically follows a dramatic shape (Baid 73, Else 67), which describes the increase/decrease of dramatic tension through time. Lighting designers design lights to parallel such escalation or drop of tension. They use contrast or affinity of saturation, brightness, or warmth/coolness of color to show tension (Block 01). Even though game designers don't adjust lighting during interaction, they manually select colors to parallel the anticipated tension and mood (Carson 00). For example, in *Silent Hill*, designers used darkness and red colored tints to signify danger and increase tension when fighting zombies or when zombies are near.

Lighting designers select angles, colors, and positions for each light in a scene to provide information about the action, the characters, or their relationships. For instance, if the scene's goal is to show distress of character *y*, then the lighting should emphasize this goal by focusing on character *y* and adjusting the colors to parallel the negative mood of the character. Also, the elevation of light angles communicates character traits that emphasize the dramatic motive. For example, a character that is under-lit is often characterized as sinister or mysterious (Campell 99).

Lighting designers also consider aesthetic qualities when lighting a scene. These qualities include modeling and depth (Birn 00). A widely used technique for modeling is 3-point lighting, where three lights are used to light a character: a backlight, a key light, and a fill light. Backlight is positioned behind the character with a slight downward angle, and is used to separate the character from the background. Key light provides the key source of illumination and is positioned at a front offset angle to emphasize texture and shape. Fill light is positioned to mirror the effect of the key light.

Another important aesthetic quality of light is depth. Depth is established by varying the colors and contrast between lights lighting the background and those lighting the foreground of a scene.

## 3.2 The Lighting design Problem

The role of a lighting designer is to establish a lighting design that serves the goals described above. However, as mentioned in chapter 1, these goals are often in conflict with one another.

To illustrate this problem consider the following example. A scene is established in a room with one window. The lighting designer adjusts the angles of light to appear as if coming from the window. At some point the character stands in a corner that is not directly lit by the window. According to the director, this moment is very important for the dramatic development of the scene, and the character's face should be sufficiently visible. Therefore, to achieve the dramatic goal, the lighting designer should add lights to establish good visibility. On the other hand, if the lighting design is to serve realism, lights that do not conform to the direction of light emitted by practical sources should not be added – and there lies the lighting designer's dilemma.

This example illustrates how lighting design, as many design problems, involves trade-offs. Lighting designers favor some goals over others depending on the lighting style chosen and the dramatic situation. In the example above, for instance, if the lighting designer chooses a realistic style, he/she will sacrifice the dramatic goal. On the other hand, if he/she chooses to conform to the dramatic and realistic goals, then character blocking and camera placement will need to be changed.

To summarize lighting design goals can be divided into the following:

**Dramatic Goals:**

- Adequately light all characters in a shot;
    1. for visibility of a character's face to show reactions and emotional expressions
    2. for visibility of a character's body to emphasize actions and gestures
- Establish visual attention by directing viewers' gaze to important characters/objects in the scene

- Establish visual tension by adjusting contrast to parallel the dramatic intensity of the scene.

**Lyrical Goals:**

- Provide mood

**Aesthetic Goals:**

- Conform to lighting style chosen for the piece
- Establish depth
- Establish character modeling

**Realistic Goals:**

- Establish logical motivation for the direction of light. For example, if a window is present in the scene then the scene have to be lit using the window as the motivational source of light
- Establish visual continuity between frames

To achieve these goals, lighting designers manipulate several lighting parameters, including the number of lights used, types of lights, and for each light: position, angle, and color. In subsequent sections, I will illustrate these parameters and show their utility in providing realistic, lyrical, dramatic, and aesthetic goals. I will first illustrate the techniques used by film and theatre designers to configure a lighting design. Since film and theatre lighting designers use the same techniques for selecting and adjusting lighting angles and colors, I will devote two sections at the end of this chapter to discuss angle and color selection in detail.

## 3.3 Lighting Design Methods in traditional media (film/theatre)

Film and theatre lighting designers use a set of documented methods and techniques to develop a lighting configuration for a performance. There are, however, several differences between film and theatre lighting design. Unlike theatre designers, filmmakers use the camera extensively to direct viewers' attention to important areas in the scene. In addition, since film is a two-dimensional medium, filmmakers rely on light to establish depth and model characters. Film lighting designers reconfigure lights and adjust their angles between each shot; which is not possible in theatre. Additionally, since many films are shot on-site, designers may use naturally available light, a privilege not available to theatre designers. Even though, film lighting design is inherently different from theatre, the underlying principles are similar. I will discuss lighting design techniques in this section outlining the differences between the rules used in film and theatre whenever necessary.

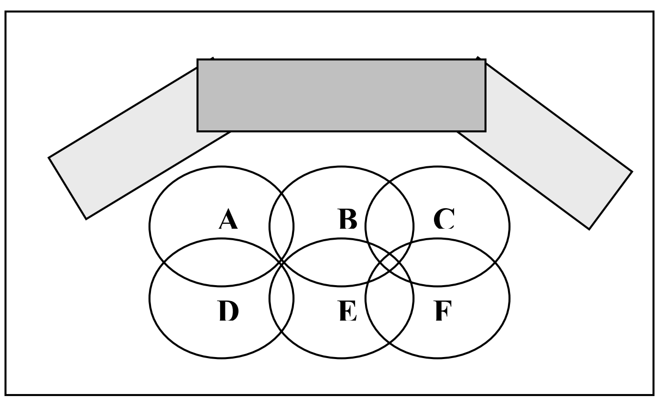



**Figure 3.1 Lighting Areas**

In lighting a theatre/film production, the designer must first analyze and define the mood and dramatic development of the performance. Such information can be gathered from reading the script, meeting with the director and the crew, and attending rehearsals. Other information collected at this stage may include the set layout and costume colors.

A theatric lighting designer first sketches the stage from three different angles: (1) left cross-section angle, (2) right cross-section angle, and (3) top view angle. He/she then divides the stage into several overlapping areas, called acting areas, whose sizes are roughly determined by the diameter of the cone of light emitted from the instruments used. In general, a lighting area is around 8 to 10 feet in diameter and 7 feet tall (Gillette 98). Figure 3.1 shows an example of a stage sketch from a top view angle, where the set shown consists of three blocks forming a platform. The figure illustrates six lighting areas.

For each area, a theatre designer designates a lighting key. A lighting key describes the angles of light projected on the given lighting area. The number of lights used depends on the type of modeling needed, the style, the mood, and the practical sources present. Additionally, other elements are taken into consideration, such as the number of dimmers and channels available. The angles of each light are chosen depending on several factors: the characters' orientation, movement, mood, and the general direction established by practical sources.

In film, on the other hand, lighting design considerations differ depending on the location of the shot. For outdoor scenes, designers use the natural light present, but adjust it by adding lights or adjusting exposure values. Lighting an indoor scene, however, similar to lighting a theatre play, requires the designer to simulate the effect of practical sources, such as the sun rays entering from the window, and balancing such effects with the mood and visibility required for the shot.

The process of designing a lighting key for film and theatre is similar, and is best described by an example. I will adopt an example from film documented and discussed by Brown (Brown 96). The set is shown in figure 3.2. The source of motivation for the light is the window, as shown. The director decides that the scene happens around the late afternoon and the action occurs over the entire set. Thus, the designer chooses a color palette that matches a daylight scene and designs the lights so that all areas of the set are sufficiently lit.

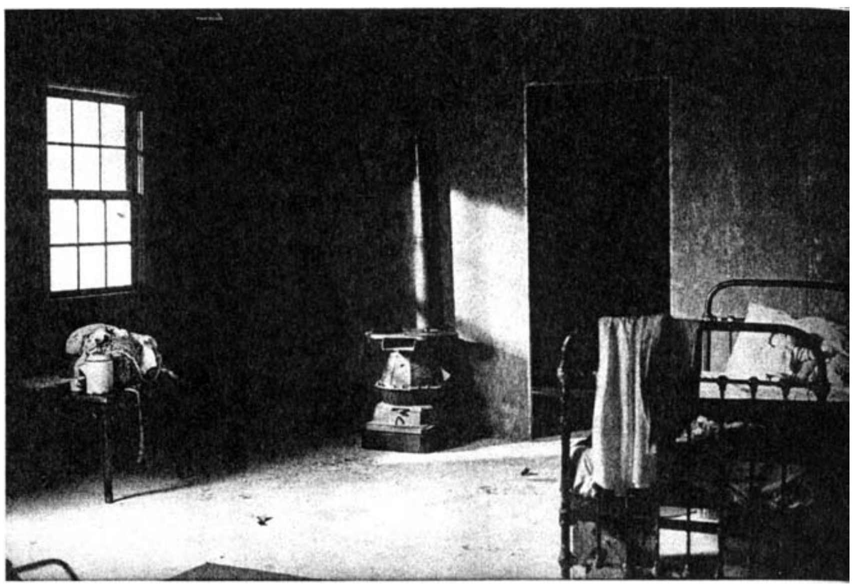


**Figure 3.2 A Scene**

(From B. Brown. (96). *Motion Picture and Video Lighting.* Focal Press © Focal Press)

The next step is to choose a lighting angle for the key light of the scene. We know that the general direction of light is the window (camera left). Depending on the mood, the lighting designer may use a key light that is at a 90° angle, which will produce a very shadowy and intense scene. On the other hand, if the director decides that the mood should be less shadowy, then an angle between 45° to 50° (left of the camera) can be used. This angle will also light the walls and the set, as shown in figure 3.2.

Notice from the figure, however, that there are several unlit areas, such as the camera upper left corner. A lighting designer can add a small diffuse light and place it behind the window with an angle directed towards these areas. This is a cheat that lighting designers often employ in their designs (Brown 96). Lighting design, whether it is for film or theatre, is an art of expression; it does not have to be purely realistic. Even if the lighting style is purely realistic, sometimes realism is sacrificed for showing some important details in an image (Brown 96). Furthermore, as Brown mentioned, viewers don't normally notice these minor lighting adjustments.

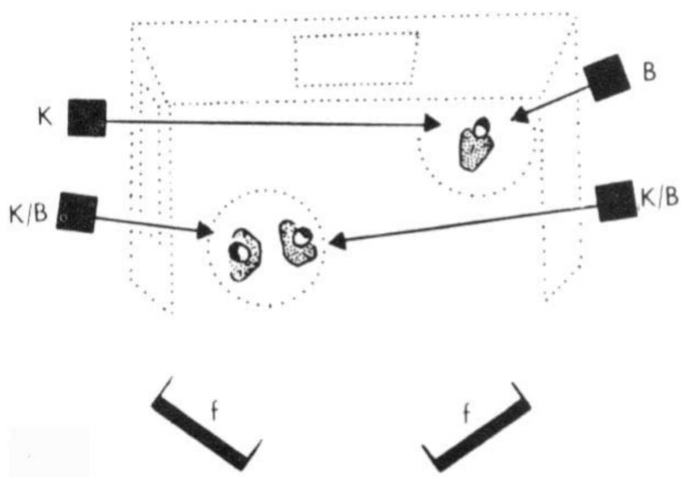



**Figure 3.3 A light Configuration**

(From Technique for Film and Television by Gerlad Millerson. Reprinted by permission of Elsevier Science Limited)

After placing lights to carefully light the set conforming to the general direction of light established by practical sources (window), the lighting designer places lights to light the characters. At this stage, the lighting designer reviews his/her notes to determine the number of characters in the scene and their movements. Film lighting designers will typically design lights such that each character has a key, a fill, and a backlight. In a two-character conversation scene, a film lighting designer may reuse the key light used for one character to backlight the other and vice versa, as shown in figure 3.3.

In both film and theatre, rehearsals and meetings with the director offer the designer guidance and knowledge of the director's desired mood and style requirements, which determine the color palette. For example, if the director wants to establish a realistic lighting style, then the color palette chosen should match the realistic color of the light emitted by the practical sources present, for instance amber can be used to match the color of light emitted by a candle or a lantern, and blue can be used to match the color of light emitted by the moon.

Lighting a theatre or a film production is dynamic; lights most often move and change color[6] (brightness, in most cases) to elicit emotions, shape the mood of the play/film, and direct audience's attention to the important areas in the scene. The designer outlines when a lighting change happens (i.e. what cue evokes a light change) and how the light changes. Light color can change in terms of both hue and brightness. During rehearsals the lighting designer will note the cue upon which a light change happens. In a theatre performance, for example, a designer will record the cue, hue (wheel number rotation) and brightness level, e.g. 'In scene 3, on Elizabeth's entrance, bring up lights connected to channel 11 to level 10'. In this example, the director or lighting designer chose to change the brightness of the lights hooked to channel 11 to a level of 10 (full brightness) upon the cue of Elizabeth's entrance.

Lighting changes should not be distracting. To achieve more gradual changes, theater designers often use a 'cross fade' effect (among others), where the lights focused on a

[6] Color is defined in terms of hue and saturation as well as intensity or brightness

specific character fade slowly on a cue and at the same time lights lighting other areas or characters are brought up slowly. This effect shifts the audience's view smoothly from one character or location to another.

In film, lighting configurations must also accommodate movements and close-ups. The lighting designer notes all movements after viewing the rehearsals. Additional lights may be placed to accommodate these movements. Designers may reuse some lights for different purposes as shown in figure 3.4. The figure illustrates a change in the camera position and character's orientation. In this case, one key light is used to ensure visibility for the two camera positions, as illustrated. However, two fill lights are needed to accommodate the change.

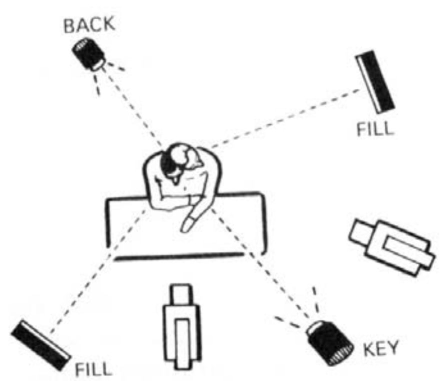



**Figure 3.4 Character Movement and Light Configuration**

(From Technique for Film and Television by Gerlad Millerson. Reprinted by permission of Elsevier Science Limited)

Lighting long shots is different from lighting close-ups. In a long shot, the character's body is visible, while his/her face is not as visible as in a close-up. Thus, lighting designers note the timing of shot changes and camera movements. They add lights to compensate for the shift between long and close-up shots. When adding these lights, they must ensure visual continuity between shots, and that the mood or contrast established carries from one shot to the other, unless otherwise specified by the director.

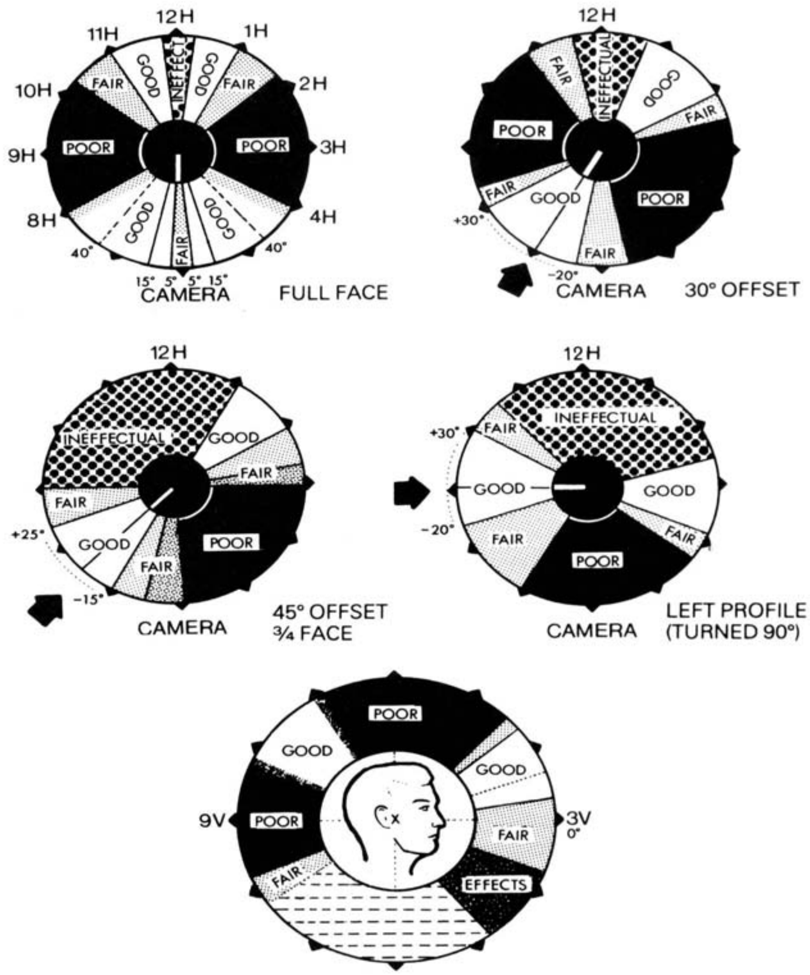



**Figure 3.5 Evaluating Angles of Key light According to Visibility**

(From Technique for Film and Television by Gerlad Millerson. Reprinted by permission of Elsevier Science Limited)

Film lighting designers select light intensities depending on the desired exposures. They use different techniques and instruments to determine exposure values of different areas in the scene, including the characters and background. They have developed techniques to adjust the exposure for different parts of the scene. For example, they often manipulate the brightness level of background areas compared to the brightness level on characters' faces to accommodate visual contrast and focus (Lowell 92).

## 3.5 Angles of Light

There are many guidelines established in film and theatre lighting design theory to help designers select angles for each type of light to achieve the desired goals. For example, numerous photographic principles are documented to guide the selection of a key light angle to establish good visibility and modeling (Millerson 91). These guidelines are illustrated in figure 3.5. The figure evaluates different placements of key light angles relative to subject's orientation. The figure shows (top to bottom, left to right) a subject facing the camera, a subject at a 30° angle from the camera, a subject at a 45° angle from the camera, and a subject in profile. In addition, the figure shows evaluations of light angles as good, fair, or poor, where poor angles are marked in black, fair angles are shaded in gray, and good angles are shown in white. The figure also shows evaluations of elevation angles. As shown, a good elevation angle is at 30° to 60° from the camera level. This range creates a natural effect similar to the effect produced by the sun (Millerson 91).

Fill light angle selection depends on the key light angle and the subject's orientation. Millerson addressed different possibilities shown in figure 3.6 (Millerson 91). In general, if the subject is facing the camera, it is best to place the fill light at an opposite angle, as seen in the figure. For all other cases, the fill is best placed at a 10° or 20° offset angle with respect to the keylight, as shown in the figure.

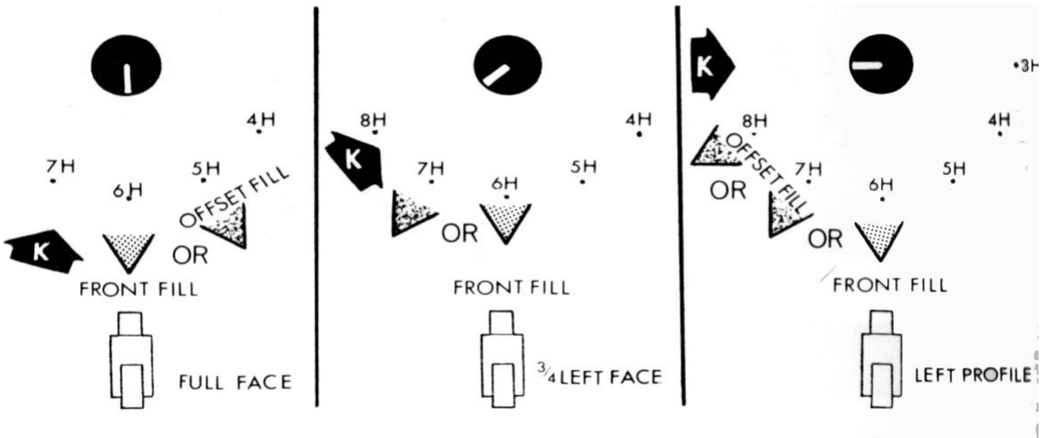



**Figure 3.6 Placements of Fill Light**

(From Technique for Film and Television by Gerlad Millerson. Reprinted by permission of Elsevier Science Limited)

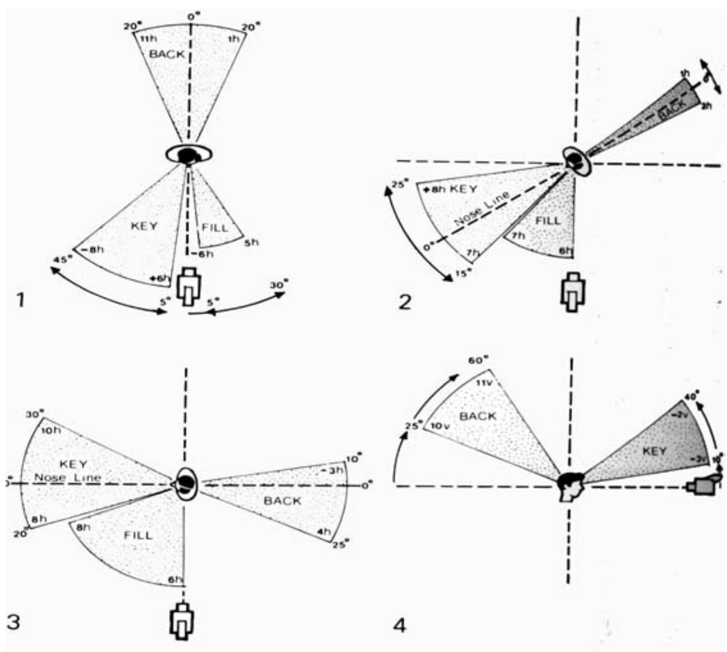



**Figure 3.7 Backlight Angles**

(From Technique for Film and Television by Gerlad Millerson. Reprinted by permission of Elsevier Science Limited)

Figure 3.7 shows different possible backlight placements depending on the key light angle and the subject to camera angle. Parts 1, 2, and 3 illustrate the selection of azimuth angles depending on the key light angle, subject orientation, and camera orientation. Part 4 illustrates the selection of an elevation angle.

Using these guidelines a lighting designer can assess the key, fill, and back light angles selected. However, these guidelines are used to assess angles in terms of visibility and modeling. As discussed above, several other goals can affect the final angle selection for each light.

## 3.6 Color of Light

Color is one of the most important elements of a lighting design. Although the appearance or dynamic range of color depends on the media used, such as live theatre, film, video, and computer graphics, the psychological impact is the same. For example, increasing the brightness contrast of a scene to 80% elicits the same response in theatre as in film. Therefore, it is not surprising to see similarities among methods used by lighting designers in film, theatre, and animation.

Many research projects have studied color and color interaction (Block 01). When choosing a light color, a lighting designer considers the utility of the selected color and its impact and interaction with other colors in the scene.

Colors are often used by lighting designers to achieve multiple goals. For example, lighting designers use color contrast to parallel the dramatic tension of the scene and direct the viewer's attention to important characters or objects. Lighting designers also carefully choose a color palette that closely matches the real colors of light emitted by the practical sources present, and this achieves realistic goals. Additionally, designers use warm and cool colors to provide mood (Block 01).

Film lighting designers use different kinds of color contrast in a scene. In film, color contrast can occur in one shot, from shot to shot, or from sequence to sequence (Block 01). Film lighting designers use contrast as a function of warm/cool, brightness, or saturation of colors. Warm/cool color contrast was used in *The Shinning*, and *Wings of the Dove*. One technique for incorporating this effect is to light an actor using a warm color (e.g. amber or candle light color) and light the background using a cool color (e.g. cyan). Saturation contrast was used in *Pennies from Heaven* and *Memento*. In these films, the cinematographer chose to create saturation contrast by alternating between saturated and de-saturated colors from one sequence to the other. Film noir movies, such as *Touch of Evil* and *Citizen Kane*, use brightness contrast where the cinematographer focuses bright white light on one area while leaving the rest of the scene in total darkness.

## 3.7 Summary

To summarize, traditional lighting design is a complex problem that involves balancing lyrical, dramatic, realistic, and aesthetic goals. Lighting design is then an optimization problem, where a lighting designer struggles to satisfy the lighting design requirements and constraints imposed by the dramatic situation and the director's choices concerning style and mood. Film and theatre lighting designers have developed guidelines, which include methods and rules for configuring placements, angles, and colors for each light in a scene to address this problem.

# CHAPTER 4

# CONTROLLING LIGHTING USING HIGH-LEVEL PARAMETERS

Adjusting the lighting configuration for an interactive scene is problematic because the visual configuration, including light positions and colors, used in the previous frame is not known at design time. ELE uses a set of sufficiently high-level parameters that allow artists and the system (ELE) to control lighting without prior knowledge of character and camera placements, or colors, angles, and placements of lights in the previous frame.

In this chapter, I will review lighting from the graphics perspective listing the rendering parameters currently used by artists and interactive entertainment systems. I will also review some of the previous research targeting the development of lighting parameters that hide rendering details.

In computer graphics, *virtual* light sources are distinctly different from the real light sources used in theatre and film. Virtual lights can be reconfigured instantaneously, positioned anywhere, with no instrument or source to disrupt the visual image.

Moreover, the calculation of the behavior of virtual light sources is not an accurate depiction of real life lighting. In real-time engines virtual light rays are not blocked by objects, and thus they can pass through walls and affect objects behind the walls. In addition, in reality light rays are absorbed by, reflected off, and refracted off surfaces. Since these effects require many hours to generate, they are often ignored by real-time rendering engines (Möller and Haines 99). However, they are approximated in non-real-time animations, such as *Toy Story*.

Graphics engines simulate different types of light sources, such as point lights and spot lights. Most graphics engines allow designers to manipulate lighting in a scene by adding, moving, or removing lights, changing their colors, setting their fall-off ratios (attenuation), and, depending on the type of light, setting and changing their orientations. Designers can also set masking parameters for each light used. Masking parameters are parameters that specify the objects a light affects, and those that it does not affect. For example, I can add a light in the scene that affects only the set and none of the characters.

In non-interactive scenes, lighting designers search for specific positions, colors, and angles that portray the desired effect. This process is done relative to characters' positions and orientations, the dramatic intensity of the event, colors and angles of light in the previous frame, and camera position and orientation. In interactive scenes, however, this process is impossible, because many of these parameters are only known during execution. Lighting designers often need to control parameters, such as visual continuity, direction of light relative to practical sources in the scene, color contrast or affinity, focus of light in the scene, and the warmth or saturation of colors within the scene. However, the methods currently offered by rendering engines, such as positioning lights and setting their orientation and colors are too specific, i.e. depend on knowledge that may not be available at design time.

I use the term *rendering parameters* to denote the parameters described above, including changing positions, colors, and orientations. I developed a set of high-level parameters, which I will call *lighting design control parameters*, to provide high-level control of lighting in an interactive scene. The parameters allow manipulation of lighting in terms of the lighting design requirements and goals described in Chapter 3.

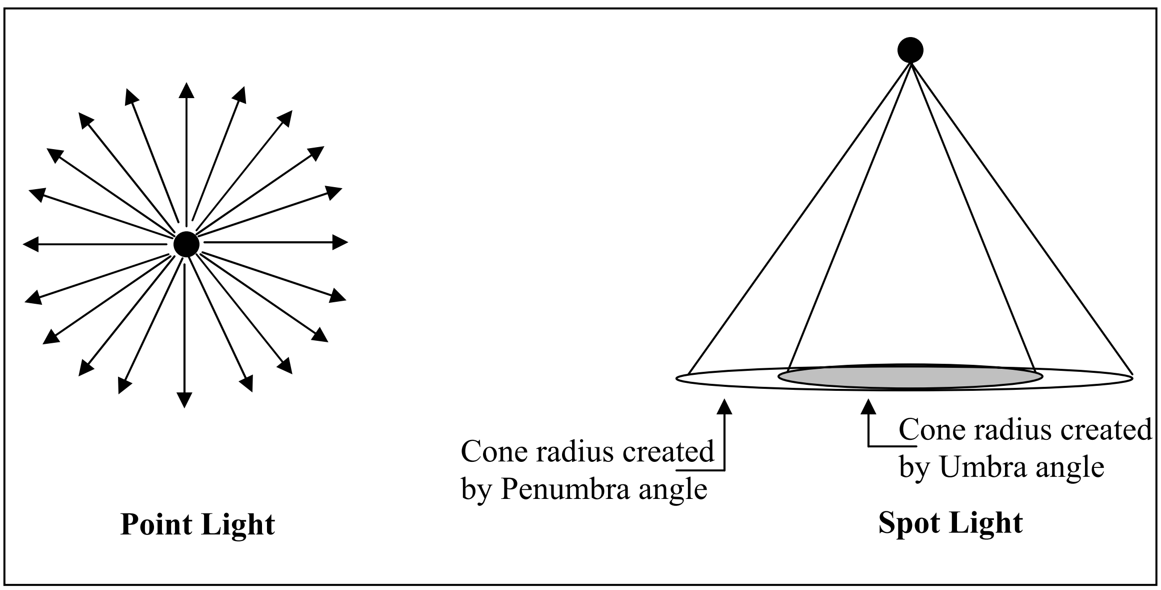



**Figure 4.1 Spot and Point Lights**

## 4.1 Rendering Parameters

Most rendering engines distinguish between three different types of lights: directional light, point light, and spot lights. Directional lights are light sources positioned infinitely far away from the subjects (e.g. the sun); they have direction, but no position. Point lights are positioned in a particular location; they emit rays in all directions, as shown in figure 4.1. Thus, point lights have position, but no direction. Spot lights, shown in figure 4.1, have position and direction. In addition, spot lights have umbra and penumbra angles defining the angle of the inner cone and outer cone projected by the light source, respectively. The light is concentrated within the inner cone; the concentration or intensity falls off from the inner cone to the outer cone section.

All three lights have color[7] parameters. The color is described as a point in red, green, blue (RGB) color space. In addition, all light types, except directional lights, have an attenuation parameter. Attenuation describes the fall off of the intensity or color from the position of the light source. In real life, light drops off by the square of the distance from the source. In computer graphics, most rendering engines simulate three types of attenuation: constant attenuation, linear attenuation, and quadratic attenuation, depending on the function they use to attenuate intensity.

## 4.2 Towards More Artistic Control Using High-Level Parameters

The parameters described above are often too low-level or specific to describe a lighting design quality in a scene. In addition, developing a lighting design by manipulating these parameters, although flexible, is a time consuming and tedious process. Several research papers have addressed this problem in the past.

[7] Color is defined in terms of hue and saturation as well as intensity or brightness

### 4.2.1 Interactive Artistic Control

Schoeneman *et al.* (93) proposed a technique that varies light color to match adjustments made by artists. Artists interactively adjust colors of areas within an image using a paintbrush tool. Given a set of fixed lights placed in a scene, the system uses linear optimization adjusting colors to closely match the artistic alterations made by the artist. To achieve interactive speed, lighting calculations were limited to direct illumination from each light, accounting for distance and visibility but not secondary reflections.

### 4.2.2 Goal-based Rendering

Kawai *et al.* (Kawai et al. 93) proposed a system that uses user specified design goals, such as energy conservatism, and impressiveness of the room (e.g. pleasant, private) to select light source emissivities, element reflectivities, and spotlight directionality. Their work is based on Flynn's research (77) that defines subjective evaluation of room illumination in terms of visual clarity, relaxation, and privacy. Kawai uses non-linear unconstrained optimization to select parameters for a radiosity-based renderer that matches the design goals specified by the user.

### 4.2.3 User Friendly Rendering

Ward proposed a user-friendly global illumination system (Ward 95). He stated that most global illumination techniques let users specify control parameters, but require substantial knowledge of the underlying implementation details. Alternatively, he proposed a system where the user sets parameters describing trade-offs between quality and time, exposure value, level of detail (high, low, etc.), and the region of interest in the image. The system then adjusts the lighting parameters given by the rendering engine to render the scene.

### 4.2.4 High-level Control of Lighting in Interactive Scenes

The goal of the work described here is to develop a system that can be used to make lighting design decisions during an interactive scene, where the spatial configuration, the importance of characters/objects, and the dramatic intensity of the moment are unknown at design time. The system addresses four main issues not addressed by the work described above. First, artists cannot interactively edit a scene nor can they supply parameters for each frame. Therefore, the system allows artists to set lighting changes in terms of rules that are triggered depending on the event/situation. Second, when authoring changes for a particular situation, artists don't know particular details of the previous frame, such as the whereabouts of characters or objects, or even current colors, angles, and positions of lights in the scene, because parameters, such as lighting colors and positions of characters are adjusted automatically in real-time depending on the events fired. Therefore, the lighting changes should be specified at a sufficiently high level. Third, the system changes the lighting to adapt to the interaction and the artistic choices made by the artist while maintaining visual continuity and style. Adjusting the lighting for every change in a scene is a daunting task. Thus, the fourth issue that ELE tackles is to automatically adapt the lighting design to the interaction.

To address these issues, I have developed a set of high-level lighting design control parameters that allows artists to manipulate the lighting design without prior knowledge of character or the camera placement or the colors, angles, or placements of lights. The system described in this dissertation allows artists to author rules that trigger a particular setting for the lighting design control parameters given the story state. The system then makes real-time decisions adjusting the lighting to match the desired effect while maintaining style and visual continuity.

Following traditional lighting design, ELE makes decisions concerning angles, colors, and light allocation depending on the established style, the visual configuration of the previous frames, and the dramatic situation. The following parameters guide the system's choices, and are used by artists to override the system's decisions:

**Parameters that target realistic goals:**

- Visual continuity variation cost (how much the artist cares about visual continuity)
- Motivation cost (describes how much the artist cares that the angle exactly matches that of the light emitted by practical sources)

**Parameters that target dramatic goals:**

- Visibility cost

**Parameters that target aesthetic goals:**

- Modeling cost (how much the artist cares about modeling the subjects)
- Depth cost (how much the artist cares about depth)

**Parameters that target lyrical goals:**

- Mood cost
- Mood angle (e.g. side-light, silhouette, rim-light, under light)

**More specific parameters that target lyrical and dramatic goals:**

- Color ideal values in terms of Hue, Saturation, Warmth, and Lightness.
- Costs of variation from the color ideal values

- Color limits, restricting color palette to realistic, cinematic, or theatric color schemes

Using these parameters an artist can manipulate the lighting design or initiate a specific change prompted by a given event. For example, an artist can adopt a high-contrast mood lighting style by setting high costs to parameters favoring expression of mood and contrast. He/she can also enforce a change of color contrast or hue triggered on an event or user action.

# CHAPTER 5

# ELE – EXPRESSIVE LIGHTING ENGINE

I have designed an Expressive Lighting Engine (ELE) that automatically configures the lighting in a scene and adjusts positions, colors, and angles of each light in real-time based on the dramatic situation and action. To make appropriate lighting decisions, ELE uses a formalized lighting design model based on cinematic and theatrical lighting design theory. While varying the lighting automatically at run-time during interaction, ELE maintains the style and mood of the interactive scene, and allows artists to override its decisions using high-level parameters. In this chapter, I will discuss ELE in detail illustrating the formalized lighting design model I have developed.

## 5.1 ELE in a Black Box

I assume that there exists a story/game engine that passes parameters to ELE. These parameters specify:

- artistic preferences concerning light change, including style and mood (the lighting design control parameters, described in Chapter 4)
- stage configuration
- character dimensions
- dramatic intensity
- anticipated camera movements for the next time step

Using these parameters ELE determines the number of lights to use, and for each light, its color, position, and orientation. ELE uses the list of anticipated camera movements to enhance its performance; this issue will be addressed in more detail in later sections. This information is then given to the rendering engine for display.

There are several other parameters needed by the rendering engine; e.g. range, penumbra and umbra values, distance from the subject, and attenuation. ELE uses several rules to determine values for these parameters given preset values, examples of these rules are discussed in Appendix C. The decision to preset these parameters while varying others was made for multiple reasons. First, I wanted to limit the number of decision variables used. Second, artists recommend specific values for some parameters. For example, animation artists recommend the use of spotlights as the default light type (Birn 00), because they offer artists control over the direction and angle of light. Third, while all of these lighting parameters affect the rendered image, only some of them directly affect the scene being rendered (Birn 00, Block 01, Viera 93). For example, the impact of color and angle on the lighting design goals (described in Chapter 3) is much more significant than the umbra and penumbra angles, distance, and attenuation.

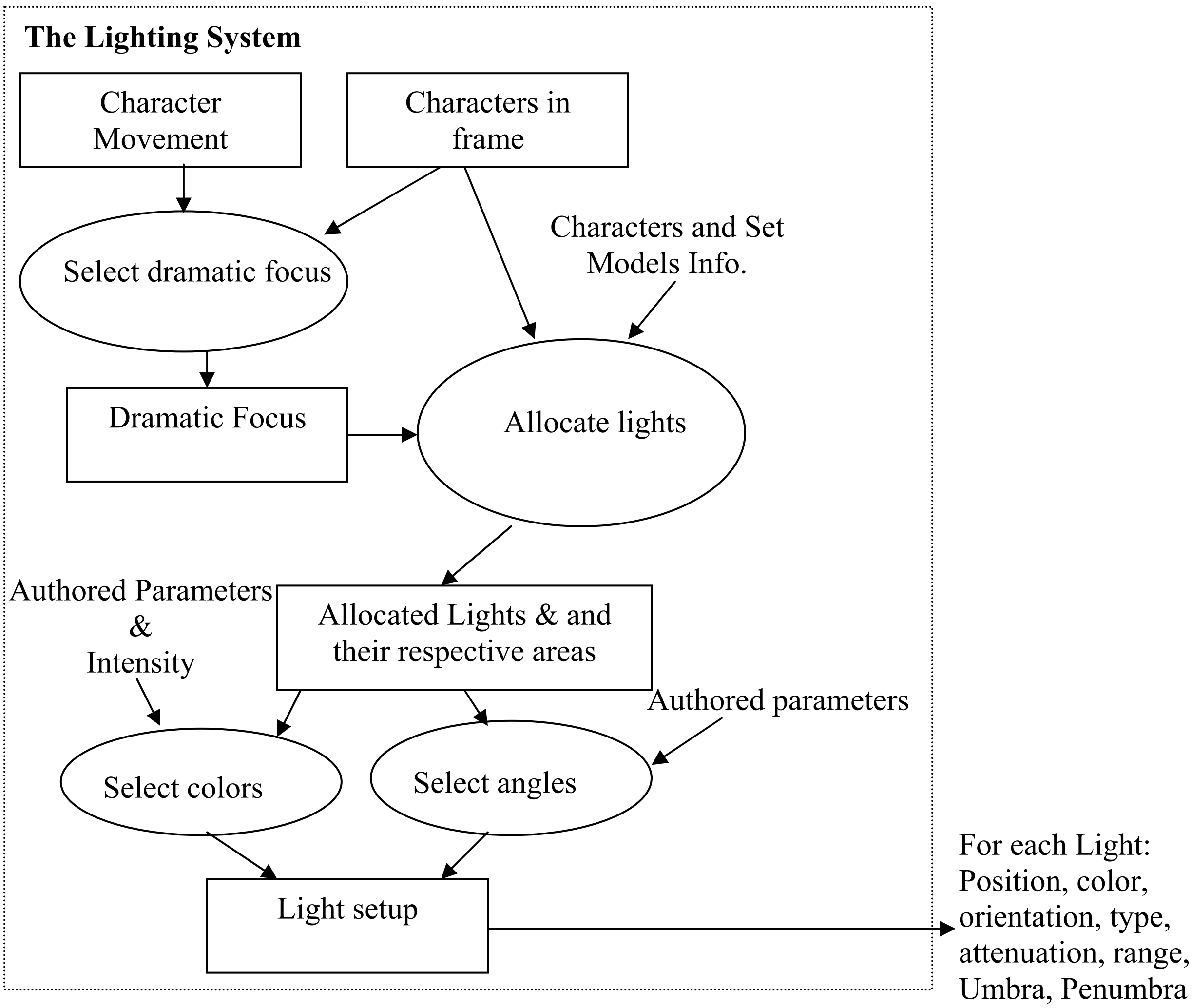



**Figure 5.1 Lighting System**

## 5.2 ELE's Architecture

As shown in Figure 5.1, ELE is composed of several subsystems that work to produce a light setup (or configuration). First, ELE determines where to direct viewers' attention given the number of characters in the frame and the dramatic importance of their action. I use the term dramatic focus to denote the area where attention should be directed. ELE then allocates lights to visible areas in the scene. Once lights are allocated to areas, ELE selects angles and colors for each light in the scene, thus forming a light setup, which is then given to the rendering engine to render the frame.

## 5.3 Selecting Dramatic Focus

Film and theatre lighting design techniques differ in terms of their function, as was discussed in Chapter 3. Film lighting designers often use the camera rather than lights to shift audience's focus. On the other hand, theatre lighting designers and painters use light to direct the viewer's attention to a specific object or character. An interactive scene can be designed to use either technique depending on the level of interaction and who controls the camera. For instance, if the participant controls the camera, then a theatrical lighting technique would be more appropriate.

To accommodate both techniques, ELE is designed to differentiate between focus, non-focus, and background areas. A focus area (dramatic focus of the scene) is a character, a group of characters, or an object. ELE selects the dramatic focus as follows: a character *c* is the dramatic focus, if:

- The camera is in a close-up, medium close-up, medium, or full shot on *c*
- The only character in view is *c*
- Character *c* has the most dramatic action (i.e. the action that has the most impact on the plot). Authors can write rules that rate actions on a scale from one to ten (ten being a very important action). For example, a running action will be judged as more dramatic than breathing or walking. In addition, ELE uses built-in common sense rules, such as talking is more dramatic than listening.

Color differences between focus and non-focus areas are used to communicate tension not just attention. Artists create visual tension by varying color contrast between lights lighting the focus and non-focus areas. For example, artists increase contrast by increasing warmth or brightness of lights lighting the dramatic focus, or decrease the brightness or warmth of lights lighting non-focus and background areas.

## 5.4 Dynamic Light Allocation

Lights are a scarce resource and need to be allocated and managed efficiently to comply with the lighting design requirements discussed in Chapter 3, while achieving real-time rendering speed. Rendering time is proportional to the number of lights used. To achieve real-time rendering speed (i.e. 30 frames/second) most rendering engines limit the number of lights used in a scene, e.g. *Wildtangent* (a publicly available rendering engine) restricts the number of lights to eight lights. On the other hand, lighting designers use many lights to provide modeling and depth, and to control the visibility of each area shown in the scene. Lighting designers, at Pixar, for example, use eight lights or more to light one character and thirty-two lights or more to light a complete scene (— 02).

To tackle this problem, ELE allocates lights only to visible areas, because allocating lights to non-visible areas is a waste of resources. For each shot or camera movement, ELE reallocates the lights. This may create performance problems, however, if the camera is panning or tilting. Given the anticipated camera movements determined by the story engine[8], ELE determines the probability of the next camera movement being a pan

[8] The story engine is a component of the interactive narrative architecture and is discussed in detail in the next chapter. The story engine determines the next possible camera movements by computing all the possible story events that could be fired next, given the story situation and the possible participant's actions.

or a tilt assuming that all camera movements anticipated have equal probability of being fired. It then calculates the visible area accordingly.

To direct viewer's attention to the dramatic focus of the scene, and to control the visibility of characters' faces, ELE divides the visible area into several areas depending on the maximum number of lights that can be used, the number of characters in the scene, and the dramatic focus computed. It then allocates lights to each area depending on the level of visibility, modeling, and depth needed.

In summary, ELE follows the steps listed below to dynamically allocate lights in a scene:

1. Calculate visible area

   If the set of next likely camera shots (given by story engine) include a pan or a tilt, then increase the visible area by some factor $\rho$ which is calculated depending on the anticipated camera speed.

2. Divide visible area into several areas differentiating between focus, non-focus, foreground, and background areas.
3. Allocate lights dynamically depending on the artistic goals and their importance as supplied by artists or determined by ELE depending on the dramatic situation, as will be described later.

### 5.4.1 Selecting and Configuring Areas

Given the visible region, ELE creates a number of areas *A* to cover the background, the foreground, and the characters within the visible region. In the next sections, I will discuss the methods by which ELE creates these areas and achieves the lighting design requirements discussed in Chapter 3.

### 5.4.1.1 Selecting and Configuring Acting (foreground) Areas

To divide the stage into several acting areas, ELE uses an approach similar to the theatrical techniques discussed in Chapter 3. According to theatre rules, it is preferable that the areas intersect (Gillette 98). Figure 5.2 shows a division of the stage into areas that overlap by *o* units. All areas have the same radius. Thus, given a desired number of acting areas, *n*, desired overlap, *o*, and the visible stage area, *V*, ELE finds a configuration of cylindrical areas $a_i$ that cover *V* with overlap *o*. In our current implementation, the restriction of 8 lights imposed by the rendering engine, limits us to 2 acting areas. Thus, we define the radius of the cylinder of any acting area, *a*, to be as follows:

$$r(\text{cyl}(a)) = \frac{w(\text{bbox}(V)) + o}{4}, \tag{1.1}$$

where $w(\text{bbox}(V))$ is the width of the bounding box of the visible region of the stage. ELE then lays out the foreground areas horizontally (or vertically if the visible region is narrow and deep vs. wide and shallow).

The height of an acting area, *a*, is then defined as:

$$h(\text{cyl}(a)) = \max_{\text{object } x \in \text{bbox}(a)} h(x) \tag{1.2}$$

Each area is assigned a spot light. ELE uses default values for the beam and field angles of the spot lights used depending on the area type (see Appendix C).

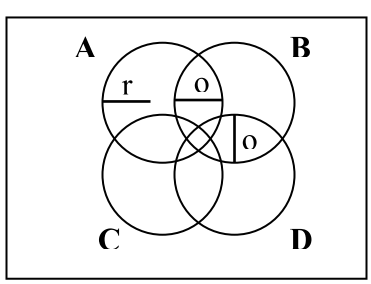


**Figure 5.2 Foreground Areas**

### 5.4.1.2 Allocating and Configuring Character Areas (focus/non-focus)

ELE uses a greedy algorithm to create areas for characters, such that all characters are assigned to an area. The algorithm is as follows:

Step 1. For each character *c* create a new area and assign *c* to it

Step 2. Repeat

For each area *a*

if $\exists\, a'$ s.t. $|a - a'| < \varepsilon$, and both are focus areas (or non-focus)

then merge *a*, *a′*

Each area *a* is lit within a cylinder cyl(*a*) with center, radius, and height given by:

$$\mathrm{center}(\mathrm{cyl}(a)) = \mathrm{center}(\mathrm{bbox}(a)) \tag{1.3}$$

$$r(\mathrm{cyl}(a)) = \|\mathrm{bbox}(a)\|_\infty + \max_{\text{character } c \in a} \|c\|_\infty + s \tag{1.4}$$

$$h(\mathrm{cyl}(a)) = \max_{\text{object } x \in \mathrm{bbox}(a)} h(x) \tag{1.5}$$

Where bbox(*a*) is the bounding box of all characters in area *a*, h(*y*) is the height of some object *y*, and *s* ("slop") is a constant. The notation $\|y\|_\infty$ is used to denote the maximum dimension of object *y*.

## 5.4.2 Allocating lights to areas

The system sets a maximum limit to the number of lights that can be assigned to each area. Non-character areas are assigned a maximum of one light. Visible character areas, however, are assigned a maximum of five lights, because character areas require fine control to establish depth and modeling.

Spot lights are used for character and acting areas. On the other hand, the type of light used to light a background area depends on the practical sources present. For example, point lights are used to simulate light emitted by torches or candles, while spot lights or directional lights are used to simulate the effect of sunlight projected from a window or a door.

As discussed in Chapter 4, many parameters affect the allocation of lights. ELE allocates lights according to cost parameters associated with visibility, modeling, depth, and visual continuity. We will define a light allocation $p : L \rightarrow A$ to be an assignment of lights to areas. Note that not all areas will be assigned a light, i.e. $p$ may not be onto. We, therefore, define the visibility, V($p$), of a light allocation to be the percentage of visible areas that are assigned lights by $p$, or:

$$V(p) = \frac{\left| \left\{ a \in A \middle| p^{-1}(a) \neq \emptyset \right\} \right|}{|A|}, \tag{1.6}$$

We define modeling as the average number of lights assigned to character areas, or:

$$M(p) = \frac{\sum_{a \in A_{character}} \left| p^{-1}(a) \right|}{\left| A_{character} \right|}, \tag{1.7}$$

The depth, D($p$), of a light allocation $p$ is the difference between the number of lights assigned to the background and foreground areas, or:

$$D(p) = \sum_{a \in A_{background}} \left| p^{-1}(a) \right| - \sum_{a \notin A_{background}} \left| p^{-1}(a) \right| \tag{1.8}$$

The visual continuity, VC($p$), of a light allocation $p$ is defined as the difference between the configuration being evaluated and the one used in the previous frame:

$$VC(p_t, p_{t-1}) = \frac{1}{|L|}\sum_{a\in A}\left\| p_t^{-1}(a) \right| - \left| p_{t-1}^{-1}(a) \right\| \tag{1.9}$$

Designers can set the style and other preferences by adjusting weights associated with each design control parameter. For example, a designer may associate higher weights to depth than modeling and visibility. In this case, the system gives higher priority to assigning lights to areas in the background and the foreground than to adding lights to character areas, which would have resulted in increased modeling.

Hence, given positions of practical sources, the stage configuration and dimensions, design parameters specifying the level of importance of depth, visibility, visual continuity, and modeling, and the formulae defined above, ELE uses a multi-objective function, which is a weighted sum of the formulae described above, where weights correspond to the lighting design control parameters supplied by the artist or calculated automatically by ELE. ELE iteratively searches for a light allocation that maximizes the objective function:

$$p_{opt} = \arg\max_{p}\left(\lambda_v V(p) + \lambda_d D(p) + \lambda_m M(p) + \lambda_{vc} VC(p)\right), \tag{1.10}$$

where $\lambda_v$ is the weight associated with visibility, $\lambda_d$ is the weight associated with depth, $\lambda_m$ is the weight associated with modeling, $\lambda_{vc}$ is the weight associated with visual continuity.

### 5.4.3 Algorithm

I formulated a greedy algorithm that allocates some lights to each visible area in the scene, as follows:

1. Each area is assigned the maximum number of lights it can have
2. Remove one light that will incur the smallest loss
3. Repeat step 2 until the number of lights assigned is less than or equal to the maximum

## 5.5 Selecting Angles

ELE selects an angle for each light in the scene. In this section, I will discuss the method by which ELE selects angles for character areas including key, fill, and backlight angles. The same techniques are used to calculate angles for acting and background areas, and thus will not be repeated.

### 5.5.1 Calculating Angles for Character Areas with Only One Character

Lighting characters involves choosing angles for the key, fill, and back lights. ELE selects values for these angles based on the lighting design control parameters and guidelines provided by the cinematic and theatrical lighting design.

The system selects an azimuth angle and an elevation angle for each key light. The azimuth angle is selected according to the lighting design control parameters: visual continuity cost, motivation cost (maintaining the illusion of a practical source by preserving its direction through the angle chosen), visibility cost (ensuring that all the characters are lit at a good visibility angle), mood cost, and ideal mood angle (e.g. side-light, backlight).

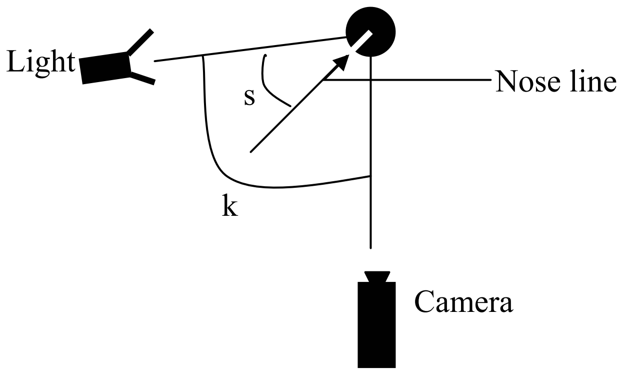



**Figure 5.3 Angles between Subject, Camera, and Light**

These parameters are not mutually exclusive, as described in Chapter 3. This problem is then formulated as a minimization problem where optimization is used to select an angle that supplies the minimum cost, and where the cost function is a multi-objective function defined as follows:

$$\text{cost}(k,k^{-},m) = \lambda_v(1-V(k,s)) + \lambda_{-}VC(k,k^{-}) + \lambda_{m_a}\left|k-m_a\right| + \lambda_l M(k,l), \quad (1.11)$$

where $k$ is the key light angle with respect to the camera as shown in figure 5.3, $s$ is subject angle with respect to the key light as shown in figure 5.3, $k^{-}$ is the previous key light angle, $l_i$ is the angle of light emitted from a practical source $i$, and $m_a$ is the mood azimuth angle (like $k$, $m_a$ is expressed with respect to the camera) suggested by the artist, and

- $\lambda_v$ is the cost of deviation from an orientation of light, which establishes best visibility and modeling
- $\lambda_{-}$ is the cost of deviation of angle $k$ from the previous light angle value $k^{-}$ (to enforce visual continuity)

- $\lambda_l$ is the cost of deviation from an established practical source angle/direction. In addition, $i$ is the number of practical sources, and $l_i$ is the angle of light from the practical source $i$ on the subject or area in question
- $\lambda_{m_a}$ is the cost of deviation from an azimuth angle that shows a specific mood

ELE calculates the degree of visibility and modeling $V(k,s)$ of a certain angle relative to the orientation of the model/objects in view. I formalized the informal guidelines given in Chapter 3, as follows:

$$V(k,s) = \sin(k)\cos(s), \tag{1.12}$$

In other words, the function is penalizing the angle difference between the subject and the key light to ensure visibility, while rewarding the angle difference between the key light and the camera to increase modeling.

Motivation is defined as follows:

$$M(k,l) = \min_i |k - l_i| \tag{1.13}$$

Visual continuity is described as the difference between the angle being evaluated and the previous angle, or:

$$VC(k,k^-) = |k - k^-| \tag{1.14}$$

The system uses a non-linear optimization algorithm based on hill climbing to find an azimuth angle that best minimizes the cost function described above. The elevation angle is set as follows:

$$\text{cost}(e,m) = \lambda_{m_e}|e - m_e| + \lambda_l \min_i |e - l_i| \tag{1.15}$$

where $e$ is the elevation light angle, $m_e$ is the mood elevation angle suggested by the artist or computed by ELE, $l_i$ is the elevation of practical source $i$, $\lambda_{m_e}$ is the weight associated with mood, and $\lambda_l$ is the weight of motivation. Gradient descent is used to find the best $e$ that minimizes the cost function above.

As discussed in Chapter 3, fill and backlight azimuth angles are calculated depending on the value of the key light angle, camera's orientation, and subject's orientation. According to the guidelines described by Millerson (Millerson 91) discussed in Chapter 3, fill light azimuth and elevation angles are calculated to be the mirror image of the key light angle. Backlight azimuth angle is calculated as follows:

$$b = (k + \pi) \bmod 2\pi, \tag{1.16}$$

Backlight elevation angle is set to $\pi/4$ (45°) as recommended by Millerson (Millerson 91).

### 5.5.2 Calculating Angles for Character Areas with More than One Character

Millerson (91) discussed several light setups that can be adopted for areas with more than one character. Since characters may be facing different directions, ELE uses an azimuth angle of $\pi/4$ (45 degrees) from the camera and an elevation angle of $\pi/6$ (30 degrees). ELE sets penumbra and umbra angles given the diameter and height of the area.

## 5.6 Selecting Colors

As discussed earlier, colors of lights in a scene portray visual tension and aid in directing viewer's attention to the dramatic focus of the scene. A change of one light's color may affect the entire image. Manipulating light color involves selecting colors for each light while evaluating the utility of that color in portraying and satisfying lighting design goals for the scene. In this section, I will discuss various methods and color spaces that are used by ELE and are relevant to discussions that will appear in later sections.

### 5.6.1 Color Spaces

Several color models, also called color spaces, have been developed (Castleman 96). These models describe color in one-, two-, three-, or four-dimensional space. Two particular color spaces are relevant to the work described here: the RGB (Red Green Blue) color space, because graphics engines use it to represent color, and HSL (Hue Saturation Lightness), because it is more familiar to designers. In addition, since the lighting design process involves manipulating color contrast and color warmth, I will review and discuss some of the formulae used to address color difference and color contrast. Since there is no published work formalizing the perceptual quality of color warmth, I will discuss methods I have developed to formalize this perceptual property.

#### 5.6.1.1 RGB Color Space

The RGB color model represents color in terms of a mixture of the three primary colors: red, green, and blue. RGB color space can be visualized as a cube, shown in figure 5.4.

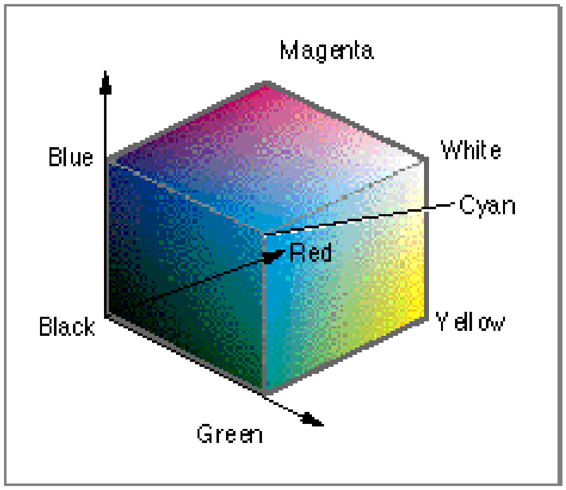



**Figure 5.4 RGB Color Space**

Any color C can then be described as a point in 3-dimensional space (r, g, b), where *r*, *g*, and *b* are values describing the amounts of red, green, and blue in the color *c*, respectively. In most graphics engines, the value of red, green, and blue range from 0 to 255; where point (0, 0, 0) corresponds to black and (255, 255, 255) corresponds to pure white.

## 5.6.1.2 Hue, Saturation, and Lightness (HSL) Color Space

HSL color model is a transformation of RGB color model into a double cone, as shown in figure 5.5. The principal hues red, green, blue, and their complements (magenta, yellow, and cyan) lie on the vertices of the hexagon shown. The hue is then measured as an angle (0°-360°). Lightness is measured as a point along the vertical axis of the double cones, as shown in the figure. The value of lightness range from 0 to 1, which corresponds to the distance along the diagonal of the RGB cube from black (lightness=0) to white (lightness=1). Saturation is measured as the radial distance from the lightness (vertical) axis of the HSL double cone space.

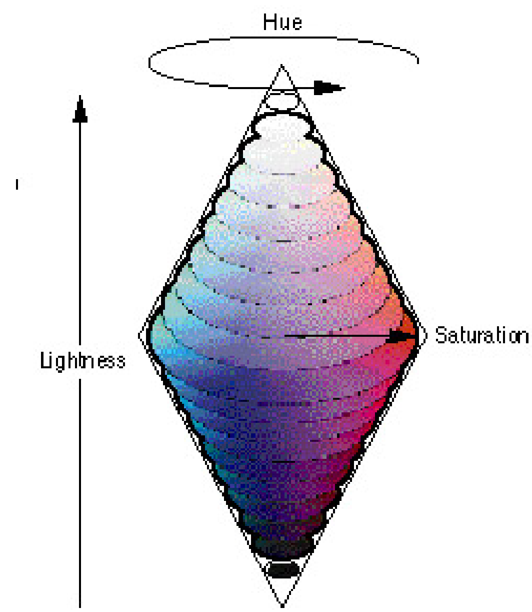



**Figure 5.5 HSL Color Space**

Many techniques were proposed to convert a color $c$ from RGB color space to HSL color space. I use the technique presented in (Castleman 96). Lightness is represented as follows:

$$\frac{1}{\sqrt{3}}[R+G+B] \tag{1.17}$$

Saturation is calculated as the radial distance as follows:

$$\sqrt{x^2+y^2} \tag{1.18}$$

where $x=\frac{1}{k_x}[2r-g-b]$, $y=\frac{1}{k_y}[g-b]$, and where $k_y$ and $k_x$ are constants.

Hue is calculated as the angle around the double cone, as follows:

$$\tan^{-1}(x,y) \tag{1.19}$$

where $x=\frac{1}{k_x}[2r-g-b]$, $y=\frac{1}{k_y}[g-b]$, and where $k_y$ and $k_x$ are constants.

### 5.6.1.3 CIE-Lab Color Space

CIE (Commission Internationale de l'Êclairage) defined three saturated primary colors (X, Y, and Z) that correspond to the way the retina behaves (*Light and the Eye* 01). The transformation from RGB to XYZ color space is a linear transformation, and is described as follows:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.24 & -1.53 & -0.49 \\ -0.96 & 1.87 & 0.04 \\ 0.05 & -0.20 & 1.05 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{1.20}$$

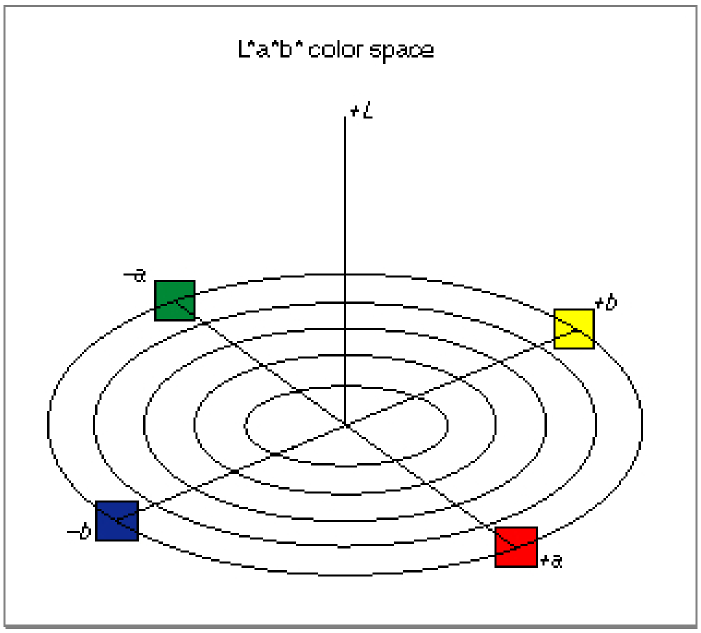



**Figure 5.6 L a* b* Color Space**

Another relevant color space is CIE-Lab L a* b* color space. It represents colors relative to a reference white point, which corresponds to what is defined as white light, represented in terms of XYZ, and usually based on the whitest light generated. The perceived difference between any two colors in this color space can be computed as the geometric distance in the color space between their color values. Thus, such color space is useful in computing color difference to determine visual continuity of color within a scene.

### 5.6.2 Color Difference

Considerable work has been published focusing on the problem of determining perceptual color difference by comparing two colors. One popular color difference formula is the CIELAB color difference equation recommended by CIE in 1976. The color difference $\Delta E^*$ between two colors is defined as the Euclidian distance between lightness L axis, red-green ($a^*$), and yellow-blue ($b^*$) opponent color axes. The formula is as follows:

$$\Delta E_{ab}^* = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \quad (1.21)$$

It can be expressed in hue $H$, lightness $L$, and Chroma $C$, coordinates as follows:

$$\Delta E_{ab}^* = \sqrt{(\Delta L^*)^2 + (\Delta C_{ab}^*)^2 + (\Delta H_{ab}^*)^2} \quad (1.22)$$

It has been found that this formula is not effective in computing color differences involving blue colors. As a consequence, researchers have proposed several modifications. CIE formed a technical committee to find a general and reliable formula to describe color difference. In 2001, CIE approved a color difference formula called CIE2000 (Hill et al. 97, Luo et al. 00). The formula is as follows:

$$\Delta E_{ab}^* = \sqrt{\left(\frac{\Delta L^*}{K_L S_L}\right)^2 + \left(\frac{\Delta C'}{K_C S_C}\right)^2 + \left(\frac{\Delta H'}{K_H S_H}\right)^2 + R_T\left(\frac{\Delta C' \Delta H'}{S_C S_H}\right)} \quad (1.23)$$

where $C' = \sqrt{a' + b^{*2}}$ , $h' = \tan^{-1}\dfrac{b^*}{a'}$, $\Delta H' = 2\sin\dfrac{\Delta h'}{2}\sqrt{C_1'C_2'}$ , $a'$ depends on a$^*$ and C$^*$, $R_T$ depends on the mean of both h′ and C′, and $C'_1$ and $C'_2$ are the C′ values for color 1 and color 2, respectively.

And where $\Delta R = R_T f(\Delta C \Delta H)$, and ΔL, ΔC, ΔH are CIELAB metric lightness, chroma, and hue differences respectively, $S_L$, $S_C$, $S_H$ are weighting functions for the lightness, chroma and hue components, and $k_L$, $k_C$, $k_H$ are parameters to be adjusted depending on model material information. *ΔL, ΔC,* and *ΔH* is the difference between $c_1$ and $c_2$ in terms of lightness, saturation, and hue, respectively.

## 5.6.3 Measuring Color Warmth

In addition to using HSL, artists often need a method to manipulate light colors in terms of their warmth or coolness (Block 00). The warmth of a color is a perceptual property. Warm colors are defined to be colors with high proportion of reds and greens (*Light and the Eye* 01), while cold colors are defined as colors with high proportion of blues relative to the reds and greens (*Light and the Eye* 01).

Several psychology and psychophysics papers describe warm and cool colors. However, none of them presented a controlled study or results that can be used to formulate warmth and coolness of colors in terms of hue, saturation, and lightness or RGB color values. The best effort to measure this elusive quality is described in an unpublished paper by Katra and Wooten. They gathered results from several experiments in which subjects rated colors on a scale of –5 to 5, where 5 is warm and –5 is cold. The stimuli were controlled for hue and saturation (Katra and Wooten 95).

The results, plotted in the graph shown in figure 5.7, describe the warmth/coolness of a color relative to hue and saturation. The function plotted in the graph is similar to a sine function, where the orange is judged to be the warmest color and blue is judged to be the coolest color. The effects of saturation and lightness were concluded to be minimal, although saturation affects the judgment of color warmth more than lightness.

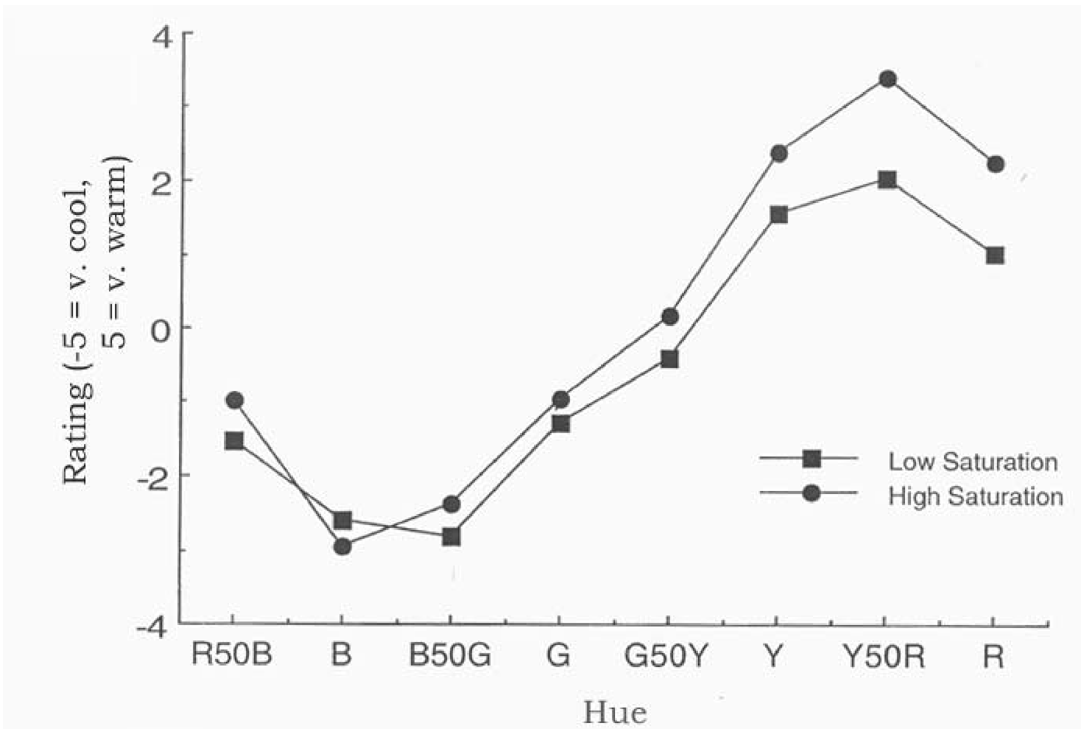



**Figure 5.7 Warmth /Coolness Ratings**

Based on the results collected by Wooten and described in the graph depicted in figure 5.7, I used a multiple linear regression method to formulate an equation describing warmth of a color, described in RGB color space. The formula is as follows:

$$warmth\left(\begin{bmatrix} R \\ G \\ B \end{bmatrix}\right) = \begin{bmatrix} 0.008 \\ 0.0006 \\ -0.0105 \end{bmatrix}^T \begin{bmatrix} R \\ G \\ B \end{bmatrix} - 0.422 \qquad (1.24)$$

### 5.6.4 Measuring Color Contrast

Lighting designers differentiate between intensity, warmth, and saturation contrast to establish visual tension. Contrast is defined relative to the dramatic focus of the scene. Therefore, color contrast is the difference between colors of lights lighting the focus area and a weighted sum of the colors lighting the other areas, that is:

$$\text{contrast}_{\phi}(c) = \sum_{i \neq focus} w_i \left| \phi(c_{focus}) - \phi(c_i) \right| \quad (1.25)$$

where $\phi$ is the color component (lightness, warmth, or saturation) over which we're computing contrast, $c$ is a vector of the light colors, $c_i$ is a color for an area type $i$, where $i \in \{focus, non-focus, background\}$ and $focus$ is the index of the dramatic focus area.

| Parameter | Value | Description |
|---|---|---|
| $\delta$ | 0-100 | Ideal value for the contrast of the image |
| $w_f$ | 0-100 | Ideal warmth value for focus areas |
| $w_n$ | 0-100 | Ideal warmth value for non-focus areas |
| $w_b$ | 0-100 | Ideal warmth value for background areas |
| $l_f$ | 0-100 | Ideal lightness value for focus areas |
| $l_n$ | 0-100 | Ideal lightness value for non-focus areas |
| $l_b$ | 0-100 | Ideal lightness value for background areas |
| $s_f$ | 0-100 | Ideal saturation value for focus areas |
| $s_n$ | 0-100 | Ideal Saturation value for non-focus areas |
| $s_b$ | 0-100 | Ideal saturation value for background areas |
| $h_f$ | 0-100 | Ideal hue value for focus areas |
| $h_n$ | 0-100 | Ideal hue value for non-focus areas |
| $h_b$ | 0-100 | Ideal hue value for background areas |
| $\lambda_o$ | 0-100 | Cost panelizing change while lights are not in view |
| $\lambda_i$ | 0-100 | Cost panelizing change while lights are in view |

| | | |
|---|---|---|
| $\lambda_C$ | 0-100 | Cost panelizing deviation from ideal $c$ |
| $\lambda_{w_f}$ | 0-100 | Cost panelizing deviation from ideal $w_f$ |
| $\lambda_{w_n}$ | 0-100 | Cost panelizing deviation from ideal $w_n$ |
| $\lambda_{w_b}$ | 0-100 | Cost panelizing deviation from ideal $w_b$ |
| $\lambda_{l_f}$ | 0-100 | Cost panelizing deviation from ideal $l_f$ |
| $\lambda_{l_n}$ | 0-100 | Cost panelizing deviation from ideal $l_n$ |
| $\lambda_{l_b}$ | 0-100 | Cost panelizing deviation from ideal $l_b$ |
| $\lambda_{s_f}$ | 0-100 | Cost panelizing deviation from ideal $s_f$ |
| $\lambda_{s_n}$ | 0-100 | Cost panelizing deviation from ideal $s_n$ |
| $\lambda_{s_b}$ | 0-100 | Cost panelizing deviation from ideal $s_b$ |
| $\lambda_{h_f}$ | 0-100 | Cost panelizing deviation from ideal $h_f$ |
| $\lambda_{h_n}$ | 0-100 | Cost panelizing deviation from ideal $h_n$ |
| $\lambda_{h_b}$ | 0-100 | Cost panelizing deviation from ideal $h_b$ |

**Table 5.1 Parameters for the Color System**

### 5.6.5 Color Selection by ELE

ELE uses the parameters listed in table 5.1 to select values for colors of each light within the scene. The parameters listed in 5.1 are either supplied by the author/artist or set by ELE using some default rules (described in Appendix C).

ELE uses a multi-objective cost function, where each objective is a squared error describing the deviation from the desired saturation, hue, lightness, depth, warmth, or contrast. These errors are weighted by costs specified by the user or selected by ELE:

$$cost(c^t, c^{t-1}) = \lambda_d \left(D(c^t) - d\right)^2 + \lambda_c \left(\text{contrast}_\phi(c^t) - \delta\right)^2 + v(x) + \sum_{i\in\{f,n,b\}} P(c_i^t, c_i^{t-1}), \qquad (1.26)$$

where $$p(c_i^t, c_i^{t-1}) = \begin{array}{l} \lambda_{s_i}\left(S(c_i^t) - s_i\right)^2 + \lambda_{h_i}\left(H(c_i^t) - h_i\right)^2 + \\ \lambda_{l_i}\left(L(c_i^t) - l_i\right)^2 + \lambda_{w_i}\left(W(c_i^t) - w_i\right)^2 + \lambda_{ch} E(c_i^t, c_i^{t-1}), \end{array} \qquad (1.27)$$

and where $c^t$ is a vector of light colors for focus, *f*, non-focus, *n*, and background, *b*, areas at frame *t*. Color $c_i^t$ is represented in RGB color space, *S(c)* denotes the saturation of color *c* and is calculated using equation (1.18), *H(c)* denotes the hue of color *c* and is calculated using equation (1.19) , *L(c)* denoted lightness of color *c* (in RGB color space) and is calculated using equation (1.17), and ELE uses CIEDE2000 described above in equation (1.23).

The depth, *D*(*c*), of a color vector, *c*, is defined as the color difference between colors of lights lighting the background areas and those lighting other areas, formulated as follows:

$$D(c) = \sum_{b \in B} \sum_{n \in NB} E(c_b, c_n) \qquad (1.28)$$

where B are the indices of the background lights and NB are the indices of the non-background lights, and where E is color difference function described in (1.23).

Artists also need to specify constraints on the color palette. These constraints are inequality constraints of the form g(x) ≤ 0. v*(x)* (defined in equation(1.26)) is a function that bounds the solution to feasible colors as described by the constraints. In the next section, I will discuss the method by which ELE ensures that the solution (i.e. colors chosen) computed satisfies the constraints and minimizes the multi-objective function above.

### 5.6.6 Algorithm Outline

The optimization problem discussed above is a constraint-based optimization problem. Operations Research literature identifies two approaches that tackle this problem. The first approach is to transform the constraint problem into several unconstrained problems and then use unconstrained non-linear optimization algorithms to solve the problem. The second approach is to transform the problem into simpler constrained problems of a linear or quadratic form and then use linear or quadratic constraint-based algorithms to solve them.

ELE follows the first approach where it uses a boundary method to bind the feasible solutions. This method is used to solve constrained non-linear programming problems with inequality constraints only. The method introduces a barrier function $v(x)$, such that $v(x) \to \infty$ as x approaches the boundary defined by the feasibility region. Thus, the problem can be reduced to solving the following unconstraint problem:

$$V(x,\varepsilon) = f(x) + \varepsilon v(x) \quad (1.29)$$

where f(x) is the objective function.

The function v*(x)* (defined in equation (1.26)) is a function that approaches ∞ when the value of the color *c* approaches the boundary specified by the constraints. Different functions can be used for v(x) (the barrier function). ELE uses the following formula for *v(x):*

$$\varepsilon \sum_{j}^{\rho} \log(-g_j(x)). \qquad (1.30)$$

The problem is not solved, however. A mechanism is still needed to solve the non-linear problem. Many algorithms are discussed in Operations Research literature to solve non-linear optimization problems (documented in Appendix A). Although gradient descent has major drawbacks, including occurrence of oscillations and being easily stuck in a local minimum, ELE uses gradient descent for several reasons. First, it provides a fast and simple solution. Second, a local minimum in this case is preferable because it provides a solution closer to the older one, thus ensuring visual continuity. Third, alternative methods rely on the existence of a second derivative, which is not necessarily true in this case.

Gradient descent is used to select RGB values for focus, non-focus, and background areas. The algorithm first starts from the previous color values then restarts the search using different random points and selects the solution with the minimum cost. The purpose of starting from the previous color values is to get to a local minimum that ensures visual continuity. However, we acknowledge that sometimes artists may sacrifice visual continuity for visual tension or visibility, thus we also restart the search randomly for several iterations and choose the color values with the minimum cost. The algorithm is as follows:

Step 1. set $x_0$ to color values from previous frame

Step 2. if $\nabla f(x_k) = 0$, then reached a minimum.

If $k < \varepsilon$ (number of iterations is below a threshold) then

set $x_0$ randomly, and go to step 3

Step 3. $d_k = -\nabla f(x_k)$, compute $x_{k+1} = x_k + \alpha_k \partial_k$

Step 4. Goto step 2

## 5.7 Automatic Selection of Weights

As mentioned in Chapter 4, artists control parameters specifying the importance of the lighting design goals, as well as constraining the colors and contrast. Manipulating these parameters for every change in a scene is a very tedious and time consuming task. Therefore, ELE automatically selects values for these parameters depending on the current situation, current dramatic intensity (given by the story engine), dramatic focus, and the established style. It uses rules, such as if dramatic intensity increases, increase contrast, but maintain contrast type established in the previous frame. These rules are defined in appendix C.

## 5.8 Interaction between Director Agent and Lighting System

ELE uses camera cuts to hide distracting lighting changes, and thus ensure visual continuity. Neuroscience and psychology literature have found that human vision does not perceive changes made between camera cuts (Henderson and Hollingworth in press). A Director Agent (as presented in the architecture) coordinates between ELE and the camera to achieve this purpose. I will outline these interactions, describing when they occur and methods by which the Director Agent handles them.

As discussed previously, ELE uses optimization to adjust colors and angles depending on the artist's cost parameters. If the cost is higher than a specific threshold and if the dominating factor is the visual continuity cost, then the lighting system sends a message to the Director Agent asking for the camera to make a cut or move to a different shot. The Director Agent first checks if the camera can do a cut. In some situations, the camera may not be able to accommodate a cut; for example, the camera has performed a cut within the past *n* seconds or it is showing an important moment which cannot be suspended. If a camera cut is not appropriate then the Director Agent will ask ELE to make the change gradually. Otherwise, it will check the camera status. Below I summarize some cases of interactions that can occur between the camera and the Director Agent:

- Camera is idle and has not performed a cut in the last *n* seconds. In this case, the Director Agent requests a cut. The Director Agent waits until the camera performs the cut and then signals ELE that a cut occurred.
- Camera is performing a cut or is going to perform a cut in the next time step. The Director Agent waits until the camera performs the cut, then it suspends the rendering process and signals ELE that a cut occurred. When ELE finishes its update, it signals the Director Agent to resume the rendering process.
- Camera is performing a movement, which is almost done. The Director Agent adds a cut to the camera action queue. After the camera finishes its actions, it signals the Director Agent. The Director Agent suspends the rendering process and signals ELE that a cut occurred. When ELE finishes its update, it signals the Director Agent to resume the rendering process.
- Camera is performing a movement, but it is going to take a long time. In this case, the Director Agent signals a fail on cut. ELE then reverts to gradual change.

In addition to hiding lighting changes caused by the color and angle changes, the Director Agent attempts to hide addition and removal of lights issued by ELE. These events need to be synchronized with camera movements. Lights are only removed or added when the areas affected are not visible or when the camera changes perspective. Events are queued and are performed when the Director Agent signals permission to make such changes. The Director Agent monitors camera movements and the queue of these changes. It keeps track of areas in view and out of view, and signals ELE when an area goes out of view where changes need to be made.

# CHAPTER 6

# IMPLEMENTATION AND RESULTS

## 6.1 Implementation

ELE was implemented using *Wildtangent* – a publicly available game engine. *Wildtangent* offers a set of libraries for creating a scene, adding 3D models exported from *3D Studio Max*, blending and calculating animations exported from *3D Studio Max*, manipulating camera movement in real-time, manipulating actors' movement in real-time, and manipulating light orientations, colors, and positions (among other parameters) in real-time.

To implement and test the lighting system, I have developed an interactive story called *Mirage* (see Appendix B). *Mirage* is an interactive Greek drama based on the Greek tragedy *Electra*.

### 6.1.1 Interactive Narrative Architecture

I have designed and implemented an interactive narrative engine. The interactive narrative architecture is composed of several systems that select story events and methods to visually present them, such as camera actions and character animations.

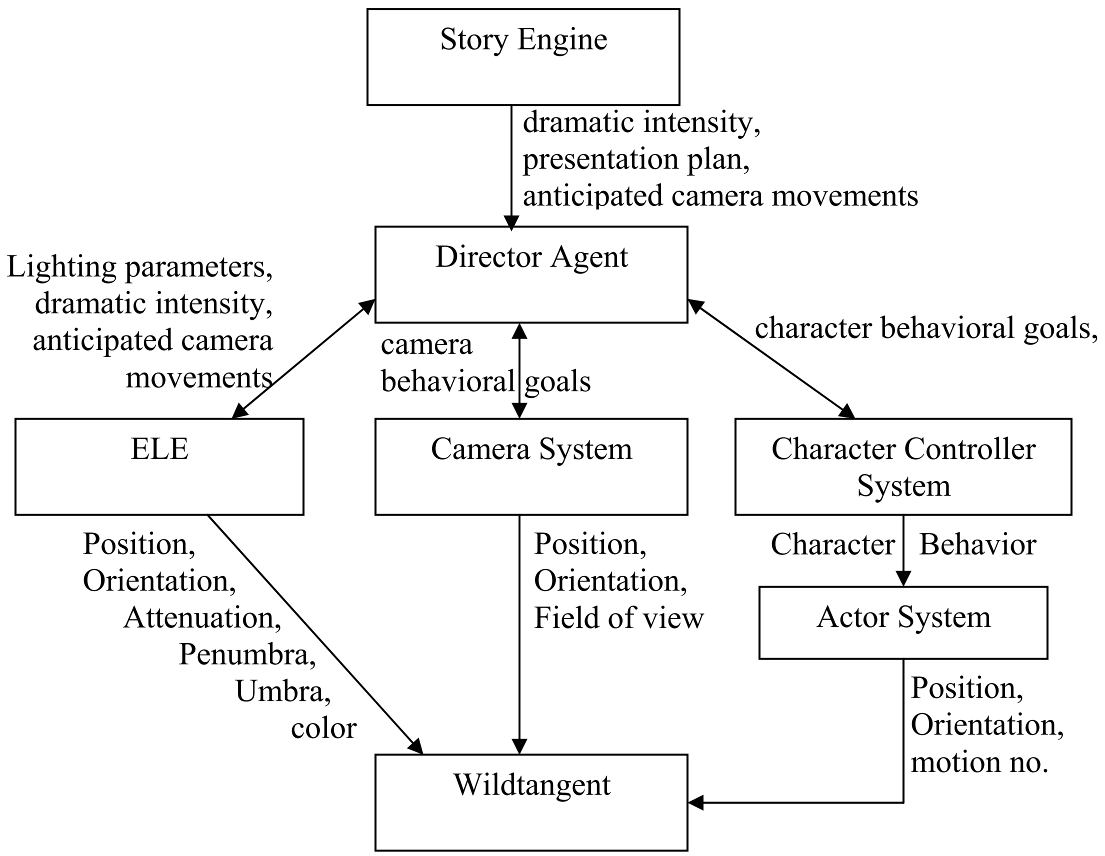



**Figure 6.1 Interactive Narrative Architecture**

Figure 6.1 shows the interactive narrative architecture. The figure shows a story engine that determines a presentation plan, dramatic intensity, and a list of anticipated camera movements. A presentation plan is composed of a collection of parallel and sequential camera and character behavioral goals, and a list of lighting parameters. Dramatic intensity is represented as a number (0-100) that describes the tension of the story situation. The presentation plan and dramatic intensity are passed to the Director

Agent. The Director Agent distributes the lighting parameters to ELE, and the behavioral goals to the camera system and character controller system. The Director Agent facilitates interactions between these systems, and supervises the executions of the behavioral goals defined in the presentation plan to ensure the right order and timing. The character controller system selects actions that achieve the behavioral goals given the current situation, and relays these actions to appropriate actors while ensuring correct timing and synchronization. The actor performs the action by either issuing a 'play' command of a predefined animation or by rotating and positioning joints or bones. ELE determines lights' placements, colors, orientations, attenuation, type, and penumbra and umbra angles (if a spotlight is used), and relays this information to the rendering engine. The camera system selects actions that achieve the behavioral goals given by the presentation plan. The rendering engine (*Wildtangent*) then renders the frame.

### 6.1.3 Story Engine

Mateas and Stern have developed a story engine that adapts basic principles of screenwriting, such as the beat construct (defined later) (Mateas and Stern 00). I use a similar architecture. However, I distinguish between narrative events and their presentation. Instead of following screenwriting theories, I augment filmmaking and acting theories (Foss 92, Benedetti 94).

First, I will define some terms:

- ***Story beat:*** this term "beat," first introduced by Stanislavski, refers to the smallest unit of action that has its own complete shape, with a central conflict and a minicrisis (Benedetti 94). According to Benedetti's description, a beat is a dramatic action that occurs in a scene to achieve a narrative goal.

- ***Visual beat:*** Foss differentiated between events that occur at the narrative level and their visual presentation. Visual beats represent visual goals for a story beat. Visual goals are concerned with the perception and intent of the action, which involves grasping attention, supplying visibility, evoking moods, and the use of body and space to convey action. The visual goals act to reinforce and present the story beat.

The story engine is divided into two levels: story or plot level, presentation or visual level. At the plot level, beats are selected depending on the sequence of events that have occurred, the relationship values between the characters, the user's modeled personality[9], and the user's history of actions. The story engine first selects a story beat then it proceeds to select a visual beat depending on dramatic intensity, the beats and actions used, the user's modeled personality, and the user's history of actions.

The story engine uses a RAP-like architecture (Firby 89) for both levels. Given a scene goal and story situation, the story engine selects a story beat that will achieve the scene goal. The algorithm iteratively loops selecting beats and breaks them down to simpler beats, until a simple beat or a group of simple beats is chosen. It then proceeds to choose a visual beat to present the story beat. The same process applies. However, visual beats may have timing constraints that will need to be propagated down the behavior tree. Once the process ends, we have a presentation plan that consists of simple visual beats

[9] The story system determines the user's personality depending on the user's history of actions given a personality model described in 4-dimentional scale: heroism, violence, self-interestedness, and cowardice.

with timing constraints, such as `(concurrent (say Electra line1) (Wave-sword-at Electra user))`, i.e. Electra should say line1 while waving the sword at the user.

There are two reasons for breaking down the architecture into two levels: reusability and failure detection and handling methods. Visual beats can be reused with different story beats, thus enhancing scalability by abstracting visual details from story beats. In addition, the methods utilized to detect and handle failure at the visual level are inherently different from those used at the story level. At the visual level, continuous checking of the user's action is important to ensure that the message is being delivered. For example, if a character is threatening the user, and the user starts playing with the objects around him/her then the message is not correctly portrayed visually, and different methods for portraying the message should be selected.

In addition to selecting a list of camera and character behaviors, the story engine determines the dramatic intensity of the moment and the list of anticipated camera movements. It determines the dramatic intensity based on a set of authored rules, e.g. if beat#2 is followed by beat #5 then dramatic intensity is increased by 10 increments. The story engine determines the next possible camera movements by computing all the possible story events that it can be fired next, given the story situation and the possible participant's actions.

Even though ELE automatically controls the lighting, an artist may force a light change directly through authoring rules. The story engine determines if a light change is required by attempting to fire several authored lighting rules given the narrative goal, the presentation plan selected for execution, and the story state. Authored lighting rules are of the following form:

```
(rule     (    (condition) )
          (    (set param1 value1)
               (set param2 value2)
               …
               (set paramn valuen)) ),
```

where `condition` is a true or false statement that involves a check on a success or failure of a narrative goal or a particular story state. For example `(:TRUE (begun scene1))` is a condition that checks if the narrative goal `(begun scene1)` is true; if it is, then the condition is true, otherwise the condition is false. $param_i$ are the lighting design parameters, specifying costs and ideal values as described in Chapter 5.

### 6.1.3 Director Agent

The Director Agent schedules and monitors the execution of the behaviors of the character controller system, the camera system, and ELE. It facilitates interaction and ensures correct timing and synchronization between the behaviors as dictated by the presentation plan. The heuristics that the Director Agent uses to synchronize lighting behaviors with camera movements, discussed in Chapter 5, are implemented.

### 6.1.4 Camera System

I have developed a simple camera system that selects and executes camera behaviors that satisfy the behavioral goals supplied by the Director Agent. Camera behaviors are defined by a camera action and several action parameters. The camera system has built-in shot types borrowed from cinematography theory (Vineyard 00, Cheshire and A. Knopf

79), such as close-ups, long shots, birds-eye views, medium shots, pans, tilts, and over the shoulder shots. An example of a camera behavior is `(close-up Electra)`, where the camera action is `close-up` and the parameter is `Electra`.

The camera system can establish two kinds of shots: cut or movement. A cut completes in one time step. However, a movement may require more than one time step to complete. The camera system keeps track of its state. It also keeps track of the remaining time it needs to complete the execution of a given shot.

Depending on the action, the camera system calculates field of view angle, orientation, and position based on the shot type and the characters' height, width, position, and orientation.

### 6.1.5 Character Controller System

The character controller system uses a HAP-based architecture (Loyal 97) to break down the character behavioral goals into simple behaviors. Simple behaviors are represented by an action, an adverb, and an actor; for example `(Walk Electra slowly)` is a behavior where the action is *walk*, the actor is *Electra,* and the manner in which an action is performed is *slowly*. Therefore, an action can be animated in different manners defined by the adverb. For example, 'take the sword' is an action that is defined as three animations 'take sword eagerly', 'take sword hesitantly', and 'take sword regretfully'.

The character controller system controls several actors. When the character system receives a behavior from the Director Agent, it relays the behavior to the appropriate actor as defined in the presentation plan. Each actor has many actions that it can do.

These actions are preset to animations with specific attitudes or character traits that pertain to the specific character executing them. Thus, for example, a female character named Electra will walk differently than the male characters Archemedis or Aegisthus. In addition, since Electra has a distinguished personality, she will walk differently than other female characters, such as Clytaemnestra. The actor will issue the appropriate animation of the behavior given by the Director Agent.

### 6.1.6 ELE

ELE described in Chapter 5, has been implemented using *Java*. Several limitations exist in the implementation, due to limitations imposed by *Wildtangent* (the rendering engine with which ELE is implemented). For example, the maximum number of lights was limited to eight, and, therefore, the number of areas was limited to a maximum of two. Also, inter-reflections and shadows were omitted from the current implementation.

## 6.2 Experiments with Lighting Components

The components for dynamically allocating lights, selecting angles, and selecting colors, were each implemented, tested, and evaluated separately. They were all then integrated and tested within the context of five scenes from *Mirage*. These scenes were shown to several film, theatre, and animation professionals.

### 6.2.1 Selecting angles

The lighting design control parameters, discussed in the previous chapters, provide a wide range of visually expressive angles. Figure 6.2 shows an example of angles produced by varying cost parameters. The figure shows the results of the system's angle selection for key, fill, and back lights. Below each image the cost parameters are documented.

As discussed earlier, artists can enforce a specific mood by setting a high mood cost and an ideal mood angle. Some of the images shown in figure 6.2 vary contrast and angle to achieve mood. Contrast, in these images, is calculated as the brightness difference between fill and key lights. Figure 6.2 (h) shows an enforced silhouette angle with a cost of 100. Figures 6.2 (b) and (c) show a light angle selected by setting the cost of mood to 100 and setting the mood angle to be a side angle (90°). The difference between images 6.2 (c) and (b) is the contrast. The contrast in image (b) is set to 100, while in image (c) it is set to 0. Thus, the effect of the fill light is very evident in image (c), while image (b) portrays a moodier and emotionally stronger image. Such expressionistic effects, especially the ones shown in images (b) and (h), are evident in many movies, such as *Ivan the Terrible* (Buckland 98).

Other images in figure 6.2 show various angles selected by the system when the artist's primary interest is visibility (image d) or mimicking the direction of the existing practical sources (images e and f). In the images shown, the practical sources are six torches positioned around the room (due to the camera angle chosen, however, only two are shown in the image). The light angle shown in the images (e) and (f) is produced from a torch, which is at a 62° angle to the right of the character's face.

### 6.2.2 Color System

The lighting design control parameters, discussed in the previous chapters, provide a wide range of expressive colors that achieve the lighting design goals. Figure 6.3 shows some examples. In particular, the figure shows cost parameters controlling contrast, color warmth, saturation, and lightness.

The first image shows a rendering of the scene where the cost of the background and foreground warmth is set to an ideal value of 100%, and the cost of deviation from that ideal value is set to 100 (very high); the lightness on the dramatic focus (the character in the scene) is set to a 100% with a cost of 100 (very high). Thus, the rendering produces a very warm image with a brightly lit subject.

The second image[10] shows a rendering of the scene where the artist assigned a hue of green to the background and foreground areas and a cost of 100 (very high) for that desired hue. The scene rendered projects the desired green hue.

The third and fourth image show a rendering of the scene where the contrast is set to a 70% (high), with a cost of 100, the contrast type is set to warm/cool contrast, and the lightness of background and foreground areas is set to a 70% (high) with a cost of 80 (high). As shown in the figure, the rendering of the scene shows a cool background with warm lights lighting the subject (as is evident in the close-up shot shown in the fourth image), thus satisfying the contrast desired.

The fifth image shows a rendering of the scene where the saturation is dropped to a 20% (low) and the associated cost with that desired saturation is 100 (very high). The lightness of the subject is set to 70% with a cost of 80 (high). Thus, the image shows low saturation with a brightly lit subject compared to the background and the foreground.

[10] The ordering used is top to bottom left to right.

The last image shows a rendering where the saturation is set to 40% with a cost of 100 (very high), and the warmth of all areas is set to 70% with a cost of 90 (very high). The image depicts such settings, as shown in the figure.

## 6.3 Results of ELE in *Mirage*

ELE has been implemented and tested in five interactive scenes from *Mirage*. All the rules used by ELE to set default parameters are documented in Appendix C. Due to limitations imposed by the media, only screenshots are displayed here; for a better illustration of ELE's performance within some interactive scenes the reader is referred to the following website: http://www.cs.northwestern.edu/~magy/.

Figure 6.4 illustrates the use of ELE to accommodate movement in an interactive scene; it shows two screenshots at the beginning of the scene; the one on the left uses ELE and the one on the right uses a static lighting design. The screenshot on the right, where the character is standing at an unanticipated orientation, was prepared by an artist. The character's face is only partially lit due to the unanticipated movement.

Figure 6.5 illustrates the utility of ELE in accommodating and portraying the evolving dramatic tension. It shows two screenshots in which Electra unsheathes a sword to attack the user. The figure shows two renderings of the scene, one using a camera fill[11] approach, and the other using ELE. As shown in the figure, ELE supports the dramatic intensity of the moment much more effectively. One viewer said that the sword was much more pronounced in the scene rendered by ELE than in the scene rendered by the other technique.

[11] A spot light attached to the camera.

Figure 6.6 shows two renderings using ELE but where the design parameters were altered. Figure 6.6 (a) shows a screenshot where the designer adjusted the parameter settings to emphasize mood and depth with realistic color palette constraints. In contrast, Figure 6.6 (b) shows a screenshot of the same moment, but where the designer adjusted the parameters to emphasize mood and depth, where color contrast is specified to emphasize the warmth/coolness dimension.

Figure 6.7 shows the effectiveness of ELE in portraying perceptual depth compared to other techniques, such as camera fill. In the figure lit by ELE (shown on the right), depth is much more pronounced because ELE assigned different lights to the background and the foreground and because the lights lighting the background were given lower intensity than those lighting the foreground.

## 6.4 Lighting Design Issues and Rendering Engines

ELE was designed using lighting techniques from film and theatre. One could argue that these lighting techniques are based on behaviors exhibited by real light sources, which are approximated in computer graphics, and thus are not exactly similar to real-life lighting. Even though this is true, animators routinely use film techniques to design light setups for their animations (Birn 00 and Calahan 96). Thus, theories used in film lighting design can be used to build animations in a computer graphics world.

Real-time rendering incurs more limitations. Since real-time rendering requires faster processing, several issues arise, such as those relating to shadows, inter-reflections, and refractions. These are very important properties of light, and acquire very important narrative properties.

Even though such limitations affect the rendered image greatly, these limitations of the current implementation. Computer hardware is advancing very fast. We anticipate that the limitation imposed on the current implementation of the real-time rendering engine will be overcome in the future. In addition, the system was built to manage the light design process and not the actual rendering process. In the dissertation, the two systems are separated; the lighting design system (described here) sits on top of the graphics engine. Thus, one can replace the graphics engine without changing ELE.

Moreover, Even though the real-time rendering engine used exhibits many limitations compared to other non-interactive rendering engines, it can still be used to portray different styles, direct viewer’s attention, and accommodate movements, as shown by the results documented here.

(a)

Long Shot, Mood Angle = 90,

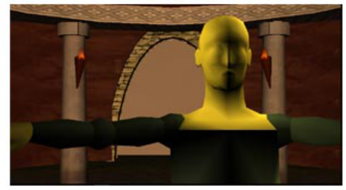
Mood Cost = 100, all other costs = 0

(b)

Medium Shot, Mood Angle = 90,

Mood Cost = 100, contrast = 100,

Contrast cost = 100, all other costs = 0



(c)

Medium Shot, Mood Angle = 90,

Mood Cost = 100, all other costs = 0

(d)

Medium Shot, Visibility Cost = 100,

Contrast = 0, Contrast cost = 100,

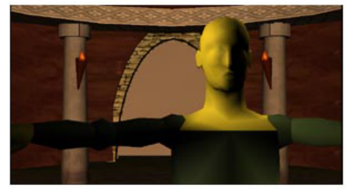
all other costs = 0

(e)

Medium Shot, Motivation Cost = 100,

contrast = 0, Contrast cost = 100,

all other costs = 0

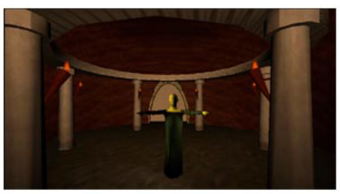


(f)

Medium Shot, Motivation Cost = 100,

Contrast = 100, Contrast Cost = 100,

all other costs = 0

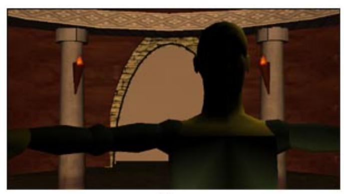


(g)

Long Shot, Motivation Cost = 100,

Contrast = 100, Contrast Cost = 100,

all other costs = 0

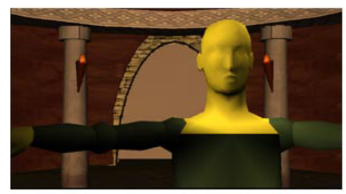


(h)

Medium Shot, Mood Cost = 100,

Mood Angle = Silhoutte,

all other costs = 0



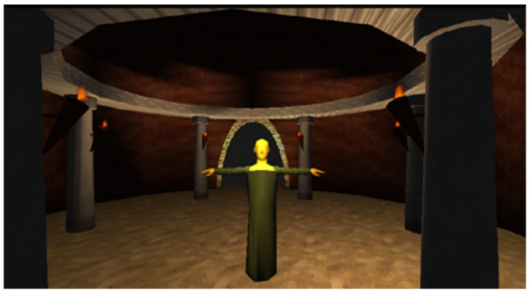
(i)

Medium Shot, Motivation Cost = 100,

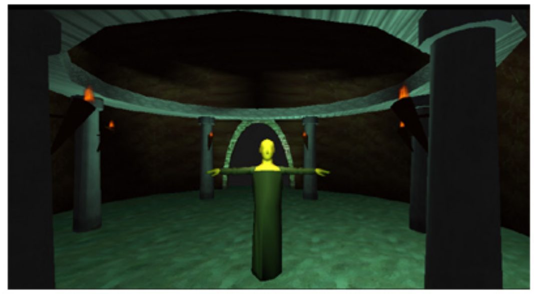
Visibility Cost =100, Contrast = 0,

Contrast cost = 100, all other costs = 0

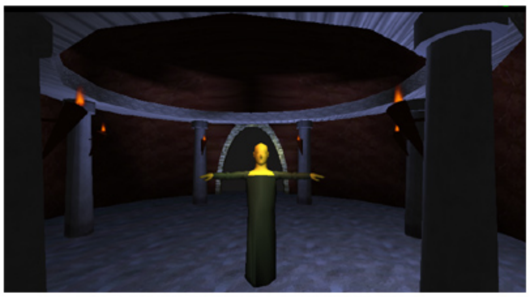
**Figure 6.2 Angles Produced by the Angle System**



Warmth= 100%, cost = 100
Lightness of Focus = 100%, cost = 100

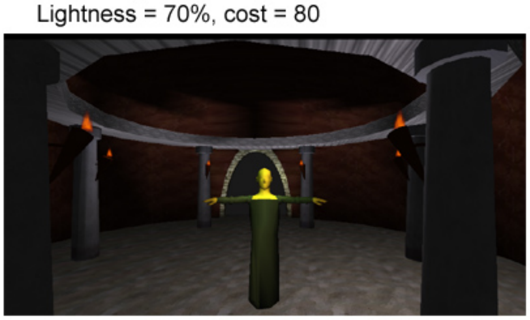


Hue= green, cost = 100
Lightness = 100%, cost = 100

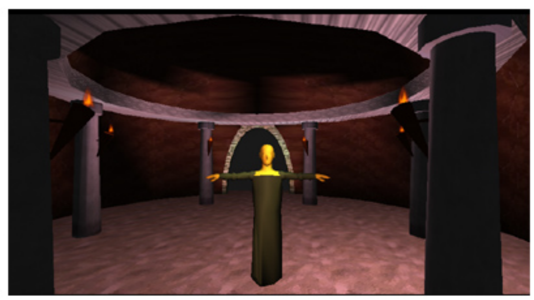


Contrast = 70%, cost = 100,
contrast type = warmth/coolness
Lightness = 70%, cost = 80

Closeup on last image



Saturation = 20%, cost = 100,
Lightness = 70%, cost = 80



saturation = 40%, cost = 100
warmth on focus and non-focus = 70%, cost = 90

**Figure 6.3 Colors Produced by the Color System**

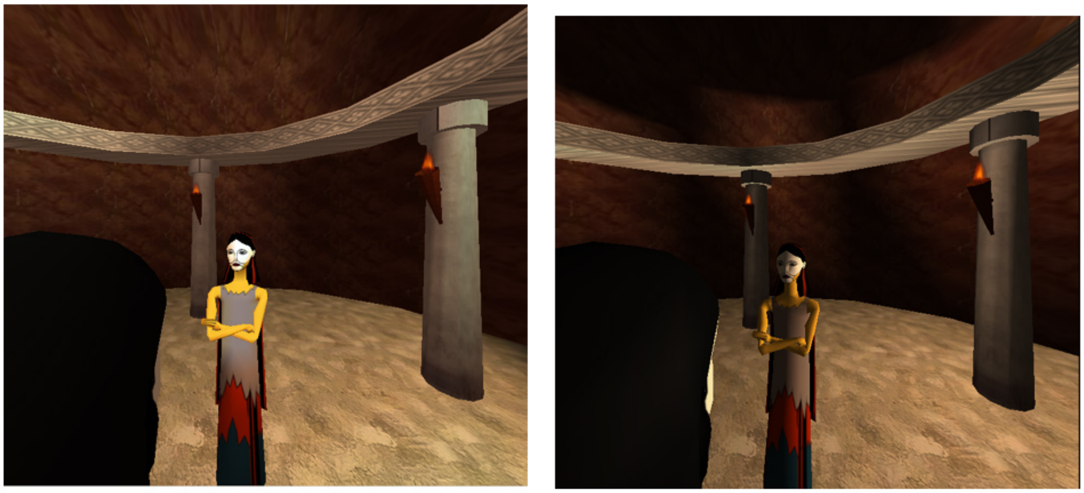


**Figure 6.4 Shows two screenshots from the beginning of the scene; the one on the left uses ELE and the one on the right uses a static lighting design. In the frame on the right, which was prepared by an artist, light angles do not adjust to accommodate character orientation.**

**Figure 6.5 Shows a screenshot from scene 8 from *Mirage* where Electra unsheathes the sword to attack Archemedis. The two images compare renderings of the same scene using the lighting system (on the right), camera fill (on the left)**

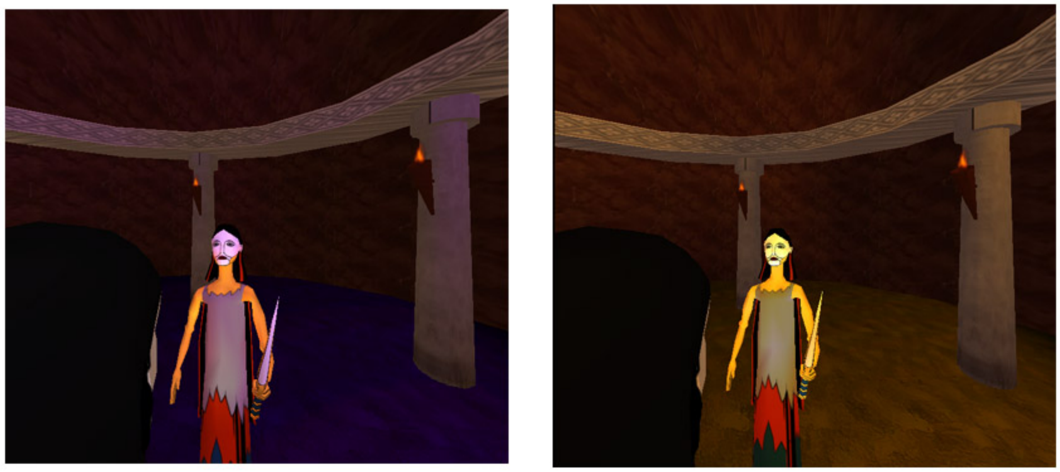
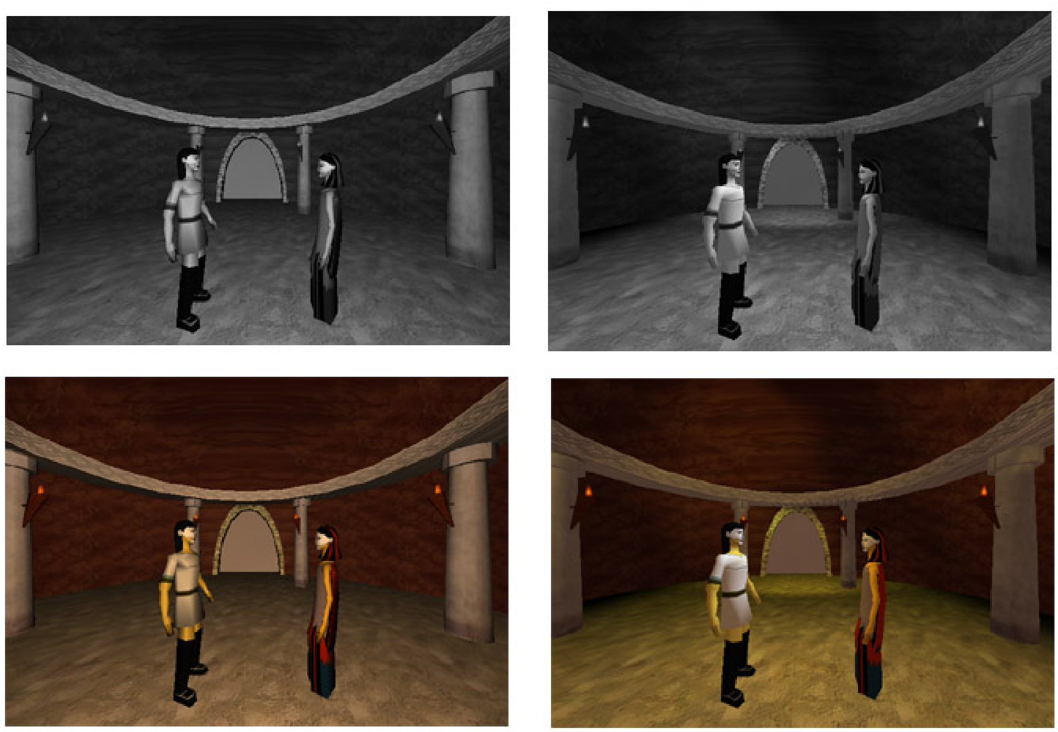


**Figure 6.6 Shows a screenshot from scene 8 from *Mirage* where Electra unsheathes the sword to attack Archemedis. The two images show different renderings of the same scene using the lighting system, but with different cost parameters.**



(a) (b)

**Figure 6.7 (a) shows a screenshot from scene 8 from *Mirage* where camera fill is used, (b) shows the same moment as (a) but with the lighting system where depth is given a high priority pictures on top portion of the image are b/w versions of the colored images shown**

# CHAPTER 7

# APPLICATIONS AND FUTURE DIRECTIONS

## 7.1 Applications

### 7.1.1 ELE as a tool for lighting interactive stories/drama

Lighting is especially important for an interactive story or drama because of its utility in shaping and supporting the narrative and the dramatic situation. Lighting forms and guides the user's perception of the characters and the world around him/her. Lighting can intensify the action, evoke emotions, direct attention to specific objects or characters within the scene, or portray information about the characters and their relationships.

Although manipulating the lighting in a scene to serve such goals is important, it is often ignored in interactive scenes due to the unpredictability of the situation. The lighting design process used in theatre, film, and animation involves predetermining characters and camera movements and setting up the lights to accommodate these movements and the dramatic shape of the scene. As discussed in earlier chapters, interaction causes variations in many of these parameters.

### 7.1.2 ELE as a lighting tool for video games

Graphics has become an essential component for video games. Lighting, in particular, has become an increasingly important element in game design, because of its influence on immersion and visual perception. Yet, current techniques for lighting design in games fail to capture the dramatic shape of the scene or suit the continuous change in the scene's spatial and dramatic configuration. ELE provides a tool for controlling lighting design in games. Since ELE makes lighting decisions based on cinematic and theatrical lighting design theory, it can strongly highlight and maintain dramatic tension, thereby enhancing immersion.

ELE also automates the lighting design process and thus eliminates massive time and effort that designers undergo to script lighting changes for a game. Game designers usually redesign the lighting for each mission/level in the game as the dramatic tension increases/decreases.

### 7.1.3 Other interactive graphical worlds

One can see the utility of ELE in many interactive environments where lighting is practical or aesthetic, requiring artistry or technical utility, such as training simulations, educational software, and virtual navigational environments. Although ELE may provide many features that such applications may not use, it can be configured to emphasize realism, visibility of objects, or provide visual focus, which are all elements that are of great utility to such applications.

### 7.1.4 ELE as a tool for animation

ELE can be used to accelerate non-interactive animation. The process of developing an animation typically involves designing the sets, the characters, the behaviors of each character, camera motions and positions, and lighting. Designing and adjusting lights in a scene is very time consuming. Designers spend hours in a trial and error process where they setup lights and evaluate their effects for every frame.

Using ELE, artists are freed from the tedious problem of positioning and angling lights, and thus can concentrate their efforts on other parts of the process. ELE provides high-level control of the lighting design process. Thus, as an alternative to positioning lights by setting 3D coordinates and orientation vectors, ELE allows artists to manipulate lighting through high-level parameters that give artists the freedom to express commands, such as ‘Rim light character x’, ‘increase contrast 50%’, or ‘increase warmth of the focus area, but keep the same contrast’.

In addition to allowing artists to control the lighting at a high-level, ELE also automatically adopts default solutions or designs by following cinematic and theatrical lighting design rules, and thus allowing artists to dodge the lighting design process all together. This speeds up the design process and frees artists to concentrate on other elements of the animation.

### 7.1.5 ELE as a tool for writers

It is sometimes hard to visualize or test an interactive story without viewing the story as it unfolds. Thus, writing an interactive scene is inherently different from writing a screenplay. In interactive media, in order to accommodate, visualize, and understand the interaction, writers most often need a tool to visualize their stories while they are writing them.

ELE is one step towards providing such a tool. ELE provides a tool that automatically lights an interactive scene. Thus, if we augment ELE with a camera and character blocking tool, then writers can write their stories in terms of visuals and evaluate the interaction visually and immediately.

## 7.2 Future Research

This dissertation has explored the role of lighting design in interactive storytelling. Many research directions still remain. First, I would like to explore the use of different lighting algorithms, such as interactive ray tracing or radiosity; I will call this area lighting appearance. Second, I would like to explore the utility of adapting theatre and acting principles to guide character blocking, postures, behaviors, and motion. Third, I would like to explore the use of framing composition principles to guide the process of selecting camera angles and positions. Last, I would like to explore the use of user modeling and its impact on interaction in interactive entertainment.

### 7.2.1 Lighting appearance

The lighting calculations and techniques used have a major impact on the image rendered; and thus on the success of the lighting design methods chosen. Real-time rendering, for example, doesn't allow self-imposed shadows or inter-reflections of light. Images produced by ELE could have been greatly enhanced had this feature been allowed. I would like to explore the possibility of adding shadows and calculation of one or two level reflection. I would also like to experiment with existing interactive global illumination methods, such as Tole et al.'s work at Cornell (Tole et al. 02).

Lighting designers in film often use exposure techniques to manipulate the amount of light projected in some areas in an image (Viera 93). Sometimes the lighting in an image appears different than expected due to interaction of light colors, and properties of the materials used (Birn 00). Graphics designers often use a histogram to judge if a picture has some exposure-related problems, such as overexposure or underexposure (Birn 00). A histogram is defined as a chart that shows tone frequency in an image (Birn 00). One future direction to enhance lighting appearance is to use techniques such as histograms to evaluate contrast, depth, and visual continuity, and to ensure that the image portrays the qualities that the designer intended.

### 7.2.2 Framing and Composition

Framing and composition is a very important area that has received little or no attention from the interactive narrative research community. Block (Block 01) discussed several theories showing the impact of shape and lines on the picture's perceived depth. Mascelli also outlines several techniques based on the arrangements of characters and camera to direct viewer's attention and show relationships between characters, such as power or weakness (Mascelli 65). I would like to adapt these theories and formalize them to create a system that enhances immersion and engagement in interactive scenes.

### 7.2.3 Character Behaviors and Blocking

Character expressiveness and behaviors is an area that has received much attention. Many researchers addressed parts of this problem. An example is Justine Cassell's (00) work on gestures and multi-modal conversation. In addition, Norman Badler et al. (Allbeck and Badler 02) used performance theories (such as LABAN) to guide selection of body motion and facial expression for synthetic characters. Also, the OZ project created a method for expression using motion speed (Loyal 97). I would like to use acting theories to explore the use of shape, space, and speed as parameters for character motion and their implication as control parameters for the expressiveness of character behaviors in interactive scenes.

### 7.2.4 Interaction Model

Through the creation of *Mirage*, we scripted the story and then experimented with various interactivity models. We decided to model the user's behavior and fit it into a multi-dimensional model describing a character stereotype; in *Mirage* the stereotypes were a reluctant hero, a coward, an impulsive character, a self-interested character, or a violent character. The system evaluates users' actions and estimates his/her stereotype as a point in the multi-dimensional stereotype model. The story events are selected depending on the user's modeled stereotype among other variables and states. I would like to further explore this method of user modeling, and, furthermore, explore its utility in interactive scenes.

### 7.2.5 Interactive Narrative Architecture

Writing an interactive narrative is hard. Writing ten-minute or more stories using a branching narrative technique is extremely difficult. Michael Mataes and Andrew Stern (Mataes and Stern 00) developed a new Interactive Narrative Architecture that facilitates the writing process. They adopted the beat construct from screenwriting theory. While working on *Mirage*, I found that the beat construct, although flexible, does not scale. I have developed an interactive narrative architecture (discussed in Chapter 6) based on filmmaking and acting theories. In the future, I would like to further explore techniques for building interactive narrative architectures that are scalable, and that can facilitate the writing process and enhance the level of engagement and interaction in interactive entertainment.

# CHAPTER 8

# SUMMARY AND CONCLUSION

Lighting has received much attention in film and theatre. Cinematic and theatrical lighting designers advocate the importance and utility of lighting design in supporting and shaping the narrative through providing mood, ensuring visibility, directing viewers' attention, and emphasizing dramatic tension. However, directly adopting such techniques for interactive scenes is problematic due to the unpredictability of design parameters, such as camera position, characters' positions, and dramatic intensity.

In this dissertation, I have discussed the Expressive Lighting Engine (ELE) that automatically and unobtrusively adapts lighting in real-time, based on cinematic and theatrical lighting design theory to accommodate movement and the continuously evolving dramatic situation while maintaining style and visual continuity. ELE represents the rules and techniques developed by cinematic and theatrical lighting designers as cost functions. It uses non-linear optimization algorithms to configure lights and select colors and angles for each light in the scene based on the high-level parameters supplied by the designer or computed by ELE.

I discussed several results comparing ELE to the currently used techniques in games and other interactive entertainment systems. The results produced by ELE suggest that it can provide more efficient, flexible, and affective lighting compared to other techniques.

In conclusion, the contributions of ELE can be summarized as follows:

- ELE automatically adjusts lighting in real-time to suit the interaction, accommodating the real-time rendering speed (30 frame/sec.) needed, and adapting to the unpredictability of the spatial and dramatic configurations of interactive scenes.
- ELE adapts cinematic and theatrical lighting design theory and formulates them into cost functions that can be optimized based on the dramatic situation and cost parameters signifying the importance of the lighting design goals.

- ELE adjusts the lighting in a scene while maintaining the established style and visual continuity. It does that by employing rules discussed by cinematic and theatric lighting designers.

- ELE adjusts the lighting automatically, but also grants artistic control. ELE provides artists a set of high-level parameters to control lighting and a language to express lighting changes. The parameters supplied to the artists are sufficiently high-level; they provide control of the mood and style without the knowledge of exact visual configuration of the previous or current frame.

ELE provides techniques that dynamically adjust the visual elements of a scene to parallel the narrative and the dramatic content, and thus enhancing, deepening, and enriching the dramatic and aesthetic engagement of interactive entertainment productions.

# REFERENCES


— About the film Toy Story. *http://tide.it.bond.edu.au/inft340/INFT340Site/340ToyStory.htm,* 2002.

J. Allbeck and N. Badler. Toward Representing Agent Behaviors Modified by Personality and Emotion. *Workshop on Embodied Conversational Agents at AAMAS 2002*, Bologna, Italy, 2002.

D. Arijon. *Grammar of the Film Language.* Focal Press: New York, 1976.

J. Alton. *Painting with Light.* University of California Press: CA, Berkely, 1995.

D. Ball. Backwards & Forwards: A Technical Manual for Reading Plays. IL: Southern Illinois University Press, 1983.

C. E. Baid. (ed.). *Drama*. New York, NY: W. W. Norton & Company, 1973.

J. Bates. The Role of Emotion in Believable Agents. *Communications of the ACM*, Vol. 37, No. 7, pp.122-125, 1992.

J. Bates, A. B. Loyall, and W. S. Reilly. An Architecture for Action, Emotion, and Social Behavior. School of Computer Science, Pittsburgh, PA: Carnegie Mellon University, Technical Rep. CMU-CS-92-144, 1992.

J. Bates, A. B. Loyall, and W. S. Reilly. Integrating Reactivity, Goals and Emotion in a Broad Agent. School of Computer Science, Pittsburgh, PA: Carnegie Mellon University, Technical Rep. CMU-CS-92-142, 1992.

W. Bares, S. McDermott, C. Boudreauz, and S. Thainimit. Virtual 3D Camera Composition from Frame Constraints. *Proceeding of International Conference on Multimedia,* CA, USA, 2000.

R. Benedetti. *The Actor at Work.* ($6^{th}$ Ed.). Englewood Cliffs, New Jersy: Prentice Hill, 1994.

D. Bertsekas. *Nonlinear Programming*. ($2^{nd}$ Ed.). Athena Scientific: MA, 1999.

J. Birn. *[Digital] Lighting & Rendering.* New Riders: IN, 2000.

B. Block. *The Visual Story: Seeing the Structure of Film, TV, and New Media.* Focal Press: USA, 2001.

D. Bordwell and K. Thompson. *Film Art: An Introduction* ($6^{th}$ Ed.). Mc Graw Hill: NY, 2001.

W. Buckland. *Film Studies.* Hodder & Stoughton, teach yourself books: Chicago, IL, 1998.

D. H. Brainard. Color appearance and color difference specification. To appear in The Science of Color, 2cd edition, S. K. Shevell (ed.), Optical Society of America, Washington D.C, in press.

B. Brown. *Motion Picture and Video Lighting.* Focal Press: Boston, 1996.

J. Bucanan. Conversation with John Bucanan and Magy Seif El-Nasr during John's visit to Northwestern University, 2002.

S. Calahan. Storytelling through lighting: a computer graphics perspective. SIGGRAPH course notes, 1996.

D. Campbell. *Technical Theatre for Non-technical People.* Allworth Press, 1999.

D. Carson. Environmental Storytelling: Creating Immersive 3D Worlds Using Lessons Learned from the Theme Park Industry. *Gamasutra.* March 01, 2000.

J. Cassell, J. Sullivan, S. Prevost, and E. Churchill. *Embodied Conversational Agents.* MIT Press: MA, 2000.

K. R. Castleman. *Digital Image Processing.* Prentice Hall, New Jersey, 1996.

D. Cheshire and A. Knopf. *The Book of Movie Photography.* Alfred Knopf, Inc: London, 1979.

B. Crowther. *Film Noir: Reflections in a Dark Mirror*. New York, NY: Continuum, 1989.

G. B. Dantzig. Programming of Interdependent Activities. II. Mathematical Model. *Econometrica* 17, 200-211, 1949.

M. El-Nasr, T. Ioerger, and J. Yen. PETEEI: A PET with Evolving Emotional Intelligence. *Proceedings of the Third International Conference on Autonomous Agents*, Seattle, Washington, 1999.

M. El-Nasr, J. Yen, and T. Ioerger. FLAME – Fuzzy Logic Adaptive Model of Emotions. *Autonomous Agents and Multi-Agent Systems.* Vol. 3, No. 3, pp. 219-257, 2000.

G. F. Else (translation). Aristotle. *Poetics*, 1967.

R. Fletcher. *Practical Methods of Optimization*. (2nd Edition). John Whiley & Sons: NY, 1987.

J. E. Flynn. A study of subjective responses to low energy and nonuniform lighting systems. *Lighting Design and Application*, 1977.

J. Firby. Adaptive Execution in Complex Dynamic Worlds. Department of Computer Science, Yale University, PhD Thesis, 1989.

D. Freeman. Four Ways to Use Symbols to Add Emotional Depth to Games. *Gamasutra.* July 24, 2002.

K. Forbus, and J. De Kleer. *Building Problem Solvers*. MIT Press: Cambridge, MA, 1993.

B. Foss. *Filmmaking: Narrative & Structural Techniques.* Silman-James Press: LA, CA, 1992.

M. Gillette and J. Gillette. *Designing with Light.* (2nd Ed.). Mayfield Publishing Company, 1989.

R. Harmon. *The Beginning Filmmaker's Guide To Directing.* Walker and Company, NY, 1993.

B. Hayes-Roth and R. van Gent. Improvisational puppets, actors, and avatars. *In Proceedings of the computer Game Developers' Conference*, Santa Clara, CA, 1996.

B. Hayes-Roth. An architecture for adaptive intelligent systems. *Artificial Intelligence: Special Issue on Agents and Interactivity,* 72:329-365, 1995.

R. Hayman. *How to Read a Play.* New York, NY: Grove Press Inc., 1977.

J. M. Henderson and A. Hollingworth. Global transsaccadic change blindness during scene perception. *Psychological Science*, in press.

B. Hill, Th. Roger, and F. W. Vorhagen. Comparative Analysis of the Quantization of Color Spaces on the Basis of the CIELAB Color-Difference Formula. *ACM Transactions on Graphics,* Vol. 16, No. 2, pp. 109-154, 1997.

H. Huston. The Actor's Instrument: Body, Theory, Stage. Ann Arbor, MI: University of Michigan Press, 1995.

J. Hooker. *Logic Based Methods for Optimization: combining Optimization and Constraint Satisfaction*. John Wiley & Sons: NY, 2000.

F. H. Imai, N. Tsumura, and Y. Miyake. Preceptual Color Difference metric for Complex Images based on Mahalanobis Distance. *In Journal of Electronic Imaging*, Vol. 10, No. 2, pp. 385-393, 2001.

S. D. Katz. *Film Directing Shot by Shot visualizing from concept to screen.* Michael Wiese Productions: CA, 1991.

E. Katra and W. Wooten. Perceived lightness/darkness and warmth/coolness in chromatic experience, 1995 *Unpublished*.

J. K. Kawai, J. S. Painter, and M. Cohen. Radioptimization – Goal Based Rendering. *Proceedings of SIGGRAPH*, 1993.

B. F. Kawin. *How Movies Work.* University of California Press: LA, CA, 1992.

I. Kerlow. *The Art of 3-D Computer Animation and Imaging.* (2nd Ed.). John Wiley & Sons, Inc: NY, 2000.

R. Lowell. *Matters of Light and Depth.* Lowel-light Manufacturing, Inc.: NYC, NY, 1992.

P. Lucey. *Story Sense.* New York, NY: McGraw-Hill, 1996.

Light and the Eye. http://www.handprint.com/HP/WCL/color1.html. Handprint Media, 2001.

C. L. Lisetti. Emotion Generation for Artificial Agents via a Hybrid Architecture. In the *Proceeding of the Workshop on Emotion-Based Agent Architectures (EBAA'99) at the Third International Conference on Autonomous Agents (Agents'99)*, 1999.

B. Loyall. Believable Agents. Ph.D. Thesis, Tech. Report CMU-CS-97-123, Carnegie Mellon University, 1997.

M. R. Luo, G. Cul, and B. Rigg. The Development of the CIE 2000 Colour Difference Formula: CIEDE2000. http://www.ifra.com/Website/ifra.nsf/html/colorqualityclub.html, 2000.

A. Maattaa. GDC 2002: Realistic Level Design for Max Payne. *Gamasutra.* May 8, 2002.

J. V. Mascelli. *The Five C's of Cinematography*. Silman-James Press: LA, 1965.

M. Mateas. Subjective Avatars. *Autonomous Agents Conference*, 1998.

M. Mateas and A. Stern. Towards Integrating Plot and Character for Interactive Drama. In *Proceeding of Socially Intelligent Agents: The Human in the Loop AAAI Fall Symposium 2000*, 2000.

M. Mateas Interactive Drama, Art, and Artificial Intelligence: A thesis Proposal, 2001.

N. McCaslin. *Creative Drama in the Classroom and Beyond.* (7th. Ed.). Addison Wesley Longman: New York, 2000.

R. Mckee. *Story: Substance, Structure, Style, and the Principles of Screenwriting*. New York: New York: HarperCollins, 1997.

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller. Equation of State Calculations by Fast Computing Machines, *J. of Chem. Phys.*, Vol. 21, No. 6, pp. 1087-1092, 1953.

G. Millerson. *The Technique of Lighting for Television and Film*. (3rd Ed.). Oxford, England: Focal Press, 1991.

S. Morgan. A Comparison of Simplex Method Algorithms. Masters Thesis, 1997.

T. Möller and E. Haines. *Real-time Rendering.* MA: USA, A K Peters, Ltd, 1999.

J. Murray. *Hamlet on the Holodeck,* Cambridge, MA: MIT Press, 1998.

R. H. Palmer. *The Lighting Art: The Aesthetics of Stage Lighting Design.* Prentice-Hall, Inc.: Englewood Cliffs, NJ, 1985.

P. M. Pardalos and M. G. C. Resende. (eds.). *Handbook of Applied Optimization*. Oxford University Press, Oxford, 2002.

F. Pellacini, P. Tole, and D. P. Greenberg. A User Interface for Interactive Cinematic Shadow Design. *Proceedings of SIGGRAPH 2002*, 2002.

K. Perlin, and A. Goldberg. Improv: A system for scripting interactive actors in virtual worlds. *Computer Graphics*, Vol. 29, No. 3, 1995.

K. Perlin. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 1, No. 1, 1995.

K. Perlin. Layered Composting of Facial Expression. In *Technical Sketch and applications part of SIGGRAPH 1997*, 1997.

F. Reid. *The Stage Lighting Handbook 4th. Ed.* London, Great Britain: A & C Black, 1992.

F. Reid. *Lighting the Stage.* Boston, MA: Focal Press, 1995.

F. Reid. *Discovering Stage Lighting 2nd. Ed.* Boston, MA: Focal Press, 1998.

W. S. Reilly. Believable Social and Emotional Agents, School of Computer Science, Pittsburgh, PA: Carnegie Mellon University, Ph.D. Thesis CMU-CS-96-138, 1996.

S. Russell and P. Norvig. *Artificial Intelligence A Modern Approach*, Upper Saddle River, NJ: Prentice-Hall Inc, 1995.

R. Rouse III. *Game Design Theory and Practice.* Wordware Publishing Inc., 2000.

C. Schoeneman, J. Dorsey, B. Smits, J. Arvo, and D. Greenburg. Painting with Light. *Proceedings of Computer Graphics, Siggraph*, 1993.

L. Seger. *Making a Good Script Great*. (2nd Ed.). Samuel French: Hollywood, CA, 1994.

M. Shapiro. *An Actor Performs*. Harcourt Brace. New York, NY, 1997.

T. Shih, J. Huang, C. Wang, J. C. Hung, and C. Kao. An Intelligent Content-based Image Retrieval System Based on Color, Shape, and Spatial Relations. *Proc. Natl. Sci. Counc. ROC(A)*, Vol. 25, No. 4, pp. 232-243, 2000.

Stanislavski. *An Actor Prepares*. Theatre Arts Books, New York, NY, 1936.

Stanislavski. *Building a Character*. Theatre Arts Books, New York, NY, 1949.

A. Stern. Virtual Babyz: Believable Agents with Narrative Intelligence. In Working Notes of *the AAAI-99 Symposium on Narrative Intelligence*, 1999.

M. Stokes, M. D. Fairchild, and R. S. Berns. Precision Requirements for Digital Color Reproduction. *ACM Transaction on Graphics,* Vol. 11, No. 4, pp. 406-422, 1992.

J. Styan. *The Elements of Drama*. Cambridge, Great Britain: Cambridge University Press, 1960.

R. Sundaram. *A First Course in Optimization Theory*. Cambridge University Press: UK, 1996.

P. Tole, F. Pellacini, B. Walter, and D. P. Greenberg. Interactive Global Illumination in Dynamic Scene. *Proceedings of SIGGRAPH 2002*, 2002.

W. Tomlinson. Interactivity and Emotion through Cinematography. Master's Thesis. MIT Media Lab, MIT, 1999.

S. R. Turner. *The Creative Process: A Computer Model of Storytelling and Creativity.* Lawrence Erlbaum Assoc.: New Jersey, 1994.

J. Velásquez. Modeling Emotion-Based Decision-Making. In: *Proceedings of the 1998 AAAI Fall Symposium Emotional and Intelligent: The Tangled Knot of Cognition*. Orlando, FL: AAAI Press, 1998.

*Vermeer: Master of Light Video*. National Gallery of Art, Washington. (also shown in Siggraph Electronic Theatre 2002), 2001.

D. Viera. *Lighting for Film and Electronic Cinematography.* Wadsworth Publishing Company:Belmont, CA, 1993.

J. Vineyard. *Setting up your shots: Great Camera Moves Every Filmmaker Should Know*. Michael Wiese Productions, Inc.: Studio City, CA, 2000.

*Visions of Light*. Beverly Hills, CA: FoxVideo: CBS/Fox Video, 1994.

P. Ward. *Picture Composition for Film and Television.* Focal Press: Oxford, 1996.

G. Ward. Making Global Illumination User-Friendly. *Proceedings of Eurographics Workshop on Rendering*, 1995.

G. Walne. (ed.) *Effects for the Theatre.* London, Great Britain: A & C Black, 1995.

Web1. http://www-fp.mcs.anl.gov/otc/Guide/OptWeb/continuous/constrained/stochastic/

Web2. http://www.isye.gatech.edu/%7Eashapiro/publications.html

P. Weyhrauch. Guiding Interactive Drama. Ph.D. Thesis, Tech. Report CMU-CS-97-109, Carnegie Mellon University, 1997.

M. Wood and G. Dantzig. Programming of Interdependent Activities. I. General Discussion. *Econometrica*, Vol. 17, pp. 193-199, 1949.

M. Young. Notes on the use of plan structures in the creation of interactive plot, in the Working Notes of the AAAI Fall Symposium on Narrative Intelligence, Cape Cod, MA, 2000.

# APPENDIX A

# OPTIMIZATION THEORY[12]

Optimization problems are concerned with finding the best solution that minimizes or maximizes a certain function *f(x)* given a number of constraints. Optimization has been studied for many years. Numerous algorithms and theories have been developed to solve optimization problems in many disciplines, including economics, engineering, and science. Far from being comprehensive, this appendix aims to briefly describe some essential concepts related to optimization theory and several optimization algorithms. These methods provide background information as a supplement to materials discussed in Chapter 5.

[12] For more comprehensive study of optimization, interested readers are referred to Bertsekas's book (99) for a more detailed discussion on Nonlinear Programming methods, Fletcher's book (87) and Sundaram's book (96) for a more comprehensive study on practical optimization methods, and Pardalos and Resende's edited book (02) for detailed discussion of research methods in the area of optimization.

# A.1 Optimization

Optimization is defined as the process of finding the best (or optimal) solution given an objective function, some constraints, and several decision variables.

Optimization is defined as computing:

$$\min f(x), x \in X \subset \Re^n \;\; or \;\; \max f(x), x \in X \subset \Re^n$$

Constraints define limitations imposed on the variables involved. For example, the objective function above can have a constraint on the variable $x$ such that $x \leq 1$ and $x \geq 0.5$. Such constraints limit the set of possible solutions. Decision variables are measurable quantities that have a definite numerical value at a given state. For example, $x$ in the formula above is a decision variable.

Optimization problems can be categorized as unconstrained or constrained, linear or non-linear optimization problems. Constraint optimization problems are problems where decision variables have some constraints that limit their feasible values. On the other hand, unconstrained optimization problems are problems where decision variables have no constraints. For these problems the optimization process will proceed to minimize or maximize the objective function without taking any constraints into account.

To further illustrate the optimization problem, I will adopt an example: expenditure minimization. A consumer consumes $x$ units of a product $X$ and $y$ units of product $Y$. Given that the price for $X$ is $a$, and the price for $Y$ is $b$. Then, to minimize expenditure one would want to minimize $ax+by$ subject to a budget $bu$, such that $ax+by \leq bu$.

Such a problem is defined to be a constraint optimization problem, because the problem includes at least one constraint.

Many optimization algorithms were devised to tackle optimization problems. These algorithms are classified as linear programming algorithms, non-linear optimization algorithms, integer programming methods, and quadratic optimization methods. In this appendix, I will discuss some of the most widely used algorithms to solve non-linear optimization problems, since these are the type of problems being addressed in this dissertation.

## A.2 Unconstrained Non-linear Optimization

Solving an unconstrained nonlinear problem is traditionally solved by generating a sequence of $x_k$, starting from $x_0$ using the following rule:

$$x_{k+1} = x_k + \alpha_k \partial_k$$

where $\alpha_k$ is the step size, and $\partial_k$ is the direction of the search. The algorithms developed to solve non-linear programming problems vary in terms of the methods they use to compute $\alpha_k$ and $\partial_k$. I will discuss some of these algorithms, including Gradient Methods, Newton Methods, Conjugate Direction Methods, and Quasi-Newton Methods. These methods are all derivative based methods, i.e. they rely on the existence of a first and/or second derivative of the objective function. Sometimes such derivatives are hard to compute or do not exist. Therefore, in this section, I will also discuss some derivative-free methods.

## A.2.1 The Gradient Method

The gradient method, steepest decent, is one of the basic methods developed and used to solve non-linear programming type problems. In this method, the direction $\partial_k$ is set to $-\nabla f(x_k)$. Thus, the algorithm works as follows:

Step 1. choose a starting point $x_0$, use constant $\alpha_k$

Step 2. if $\nabla f(x_k) = 0$, then stop reached a minimum

Step 3. $\partial_k = -\nabla f(x_k)$, compute $x_{k+1} = x_k + \alpha_k \partial_k$

Step 4. Goto step 2

While this method is attractive because of its simplicity, it does not guarantee a global optimal solution. The algorithm stops at a stationary point, which may be local minima depending on the structure of the objective function. Depending on the shape of the function, the algorithm may have a slow convergence rate due to the occurrence of oscillations (Pardalos and Resende 02 and Bertsaekas 99).

## A.2.2 Newton's Method

Newton's method is one of the most powerful methods developed for solving unconstrained non-linear programming problems. It is based on the first and second derivative of $f(x)$.

The direction of the search is as follows:

$$\partial_k = \frac{-\nabla f(x_k)}{\nabla^2 f(x_k)}$$

Thus, the standard Newton's algorithm is as follows:

Step 1. set $\alpha_k$ to 1(constant step size)

Step 2. choose a starting point $x_0$

Step 3. if $\nabla f(x_k)=0$, then stop reached a minimum

Step 4. Compute $x_{k+1}=x_k-\frac{\nabla f(x_k)}{\nabla^2 f(x_k)}$

Step 5. Goto step 3

The algorithm is very fast, and, unlike the steepest descent, it displays no oscillations in their search process. However, the purest form of the Newton's method described above has several drawbacks. First, the second derivative of the objective function may not exist in which case the method fails. Second, the pure form of the search is not a descent, and thus it may be attracted to local maxima as much as local minima (Bertsekas 99). Also, depending on the starting point, the algorithm may not converge (Pardalos and Resende 02). Therefore, a number of modifications were made to the standard Newton's method, yielding Line search Newton's algorithm and Trust-region Newton's algorithm (Pardalos and Resende 02).

The line search Newton's algorithm computes the direction as follows:

$$\partial_k=\begin{cases}\dfrac{-\nabla f(x_k)}{\nabla^2 f(x_k)}, & \text{if } \nabla f(x_k)' d_k \le c_1 \left\|\nabla f(x_k\right\|^q, \left\|d_k\right\|^p \le c_2 \left\|\nabla f(x_k)\right\| \\ -\nabla f(x_k), & \text{otherwise}\end{cases}$$

where $p\geq 2$, $q\geq 3$, $c_1>0$, and $c_2>0$.

### A.2.3 The Quasi-Newton Method

Quasi-Newton methods are methods of the same form as Newton's methods. However, quasi-Newton's methods avoid using the second derivative. Thus, instead of using the second derivative quasi-Newton algorithms use a direction $\partial_k$, which is computed as a vector that approximates the Newton direction.

The basis of the quasi-Newton method is that the successive iterates $x_k$ and $x_{k+1}$ with their corresponding gradients $\nabla f(x_k)$ and $\nabla f(x_{k+1})$ yield curvature information that can be used to approximate the second derivative relation, as follows:

$$\nabla^2 f(x_{k+1}) \approx \frac{\nabla f(x_{k+1}) - \nabla f(x_k)}{x_{k+1} - x_k}$$

Using this information, several methods have been proposed to calculate direction, $\partial_k$. For more information readers are referred to (Pardalos and Resende 02 and Bertsekas 99).

Quasi-Newton methods converge fast. Their convergence rate was shown to be $O(n^2)$ (Bertsekas 99). However, for large-scale problems forming and storing the vector for computing the direction could be impracticable and problematic.

### A.2.4 The Conjugate Gradient Method

The conjugate gradient method is a very popular optimization method for solving non-linear programming problems due to its simplicity. It also requires less space and is less computationally expensive than Newton's methods. It was first developed to solve quadratic optimization problems, and then it was generalized to handle arbitrary nonlinear optimization problems. The following describes the algorithm:

Step 1. choose a starting point $x_0 \in \Re$

Step 2. if $\nabla f(x_k) = 0$, then stop reached a minimum

Step 3. Compute $\alpha_k$ using armjio's algorithm

Step 4. Compute scalar value $\beta_k = \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2}$

Step 5. compute direction $\partial_k$ = $\begin{cases} -\nabla f(x_k) & k = 0 \\ -\nabla f(x_k) + \beta_{k-1}\partial_{k-1} & k > 0 \end{cases}$

Step 6. compute $x_{k+1} = x_k + \alpha_k \partial_k$

Step 7. goto step 2

The method is fast and simple. However, it does not guarantee a global optimal solution.

## A.2.5 Non-Derivative Methods

Non-derivative or derivative-free methods are optimization algorithms that solve a nonlinear programming problem using no derivative in the process. This is useful for problems where the derivative is either impossible to compute or unreliable. There are several non derivative methods, such as coordinate descent methods, and direct search methods. I will briefly review these methods. For more information readers are referred to Bertsekas (99).

The coordinate descent algorithm minimizes the cost at each iteration along one coordinate direction. The algorithm varies the order of coordinates chosen to guide the search direction. One of the advantages of this method is that it is well suited for parallel computation. The convergence properties for this method are very similar to that of gradient descent.

The direct search algorithm samples the objective function along some directions in the neighborhood of the current point in the iteration. These methods depend on the sampling rule and the choice of directions (Pardalos and Resende 02). The convergence properties of these methods are often unsatisfactory (Bertsekas 99).

Simulated annealing is another algorithm that is used to solve non-linear optimization problems. It was proposed by Metropolis (Metropolis 53). One of its major advantages is its ability to avoid getting trapped in local minima.

The algorithm selects a point $S'$ from the neighborhood of a current solution $S$. This point is then tested with the objective function. If it yields a better solution, then $S'$ becomes the new solution. If it doesn't yield a better solution, it still accepts the move with probability $e^{\frac{-D}{T}} > q, where\ 0 < q < 1$, where $q$ is generated randomly. The value of the variable $T$ decreases with time, thus the probability decreases as the search goes on. The process continues until a stopping condition is satisfied, i.e. cannot accept uphill moves and solution S is better than any of the neighbors (Pardalos and Resende 02).

## A.3 Constrained Non-linear Optimization

Two approaches exist to solve constrained non-linear optimization problems. The first approach is based on transforming the constrained problem into a number of unconstrained problems and use the algorithms discussed above for solving the unconstrained problems. The second approach is to transform the problem into simpler constrained problems of a linear or quadratic form and then used linear or quadratic constrained algorithms to solve them.

The problem can be outlined as follows:

$$\min \; f(x)\text{, some constraints, such as } g(x) \leq 0;\ h(x)=0$$

where $x \in \Re^n$, and $f: \Re^n \rightarrow \Re$ is the objective function. Thus, a feasibility region can be defined as follows:

$$F = \left\{ x \in \Re^n : g_i(x) < 0, i = 1,\ldots,k \right\}$$

## A.3.1 Exterior Penalty Methods

A penalty function $p(x)$ is introduced, such that:

$$p(x) = \begin{cases} 0 & x \in F \\ > 0 & otherwise \end{cases}$$

Thus, the optimization problem can be reduced to solving the following unconstrained nonlinear problem:

$$\min_{x \in \Re^n} \left[ f(x) + \frac{1}{\varepsilon} p(x) \right]$$

where $\varepsilon$ is the penalty parameter and it is given a value > 0. The algorithm then is described as follows:

Step 1. choose $\varepsilon$ such that $\varepsilon_{k+1} < \varepsilon_k$, and $\lim_{k \rightarrow \infty} \varepsilon_k = 0$

Step 2. choose a starting point $x_0$, k =0

Step 3. solve unconstrained problem: $\min_{x\in\Re^n}\left[f(x)+\frac{1}{\varepsilon}p(x)\right]$

Step 4. Check if $x_k$ solves the problem;

if so stop, otherwise go to step 5

Step 5. set $x_0 = x_k$, k=k+1, and goto step 3.

The function $p(x)$ varies depending on the penalty method used. One of the common penalty methods used is called Quadratic Penalty method, which defined $p(x)$ as follows:

$$P(x,\varepsilon)=f(x)+\frac{1}{\varepsilon}\left\{\ \left\|\max\{g(x),0\}\right\|^2+\left\|h(x)\right\|^2\right\}$$

where g(x) describe inequality constraints and h(x) describe constraints of the form h(x) = 0. The quadratic penalty method is simple. However, it has some disadvantages. First, as $\varepsilon$ becomes smaller, the Hessian matrix $\nabla^2 p(x)$ becomes ill conditioned, and thus the minimization problem becomes more difficult and slow. Therefore, this method is no longer used; it has been replaced by lagrangian methods, which are addressed below.

## A.3.2 The Logarithmic Barrier Method

This method is used to solve constraint nonlinear programming problems with inequality constraints only. The method introduces a barrier function $v(x)$, such that $v(x)\rightarrow\infty$ as x approaches the boundary defined by the feasibility region. Thus, the problem can be reduced to solving the following unconstrained problem:

$$V(x,\varepsilon)=f(x)+\varepsilon v(x)$$

Different functions can be used for v(x), the barrier function. One of the popular functions used is as follows:

$$\varepsilon \sum_{i=1}^{p} \log(-g_i(x))$$

The method suffers the same disadvantages as the penalty method. Therefore, this method is not used to solve general nonlinear programming problems. However, it has proven to be very useful for solving linear constraint problems as stated by (Pardalos and Resende 02).

## A.3.2 The Projected Gradient Method

Several methods were developed to solve constraint nonlinear programming problems using feasibility direction $(\bar{x}_k - x_k)$, where $\bar{x}_k$ is a feasible vector other than $x_k$. The conditional gradient method solves the problem by iterating through different values of $x_k$ until a local minima point is reached. Thus, the algorithm computes $x_{k+1} = x_k + \alpha_k(\bar{x}_k - x_k)$, where $\alpha_k$ is the step size, which is set using a line search algorithm, as described above. $\bar{x}_k$ is computed using the following formula: $\bar{x}_k = \arg \min_{x \in X} \nabla f(x_k)'(x - x_k)$. The convergence rate for this method, however, tends to be slow when the feasibility region is polyhedron.

Gradient projection method solves for $x_{k+1} = x_k + \alpha_k(\bar{x}_k - x_k)$. $\bar{x}_k$ is computed using the following formula: $\bar{x}_k = [x_k - s_k \nabla f(x_k)]^+$, where $[.]^+$ denotes projection, and $s_k$ is a scalar. Hence, $\bar{x}_k$ is computed by taking a step towards the negative gradient direction (as in steepest descent), which is then projected. A feasibility direction is then computed using this value. The algorithm then moves along the feasibility direction. This method is simple, but suffers the same disadvantages as steepest descent, i.e. it does not guarantee a global optimal solution and it may be slow depending on the form of the function and the step size used.

# APPENDIX B

# MIRAGE

## B.1. MIRAGE - THE INTERACTIVE STORY

I transformed the legend of the house of Argos into an interactive story, called *Mirage*. Ancient Greek dramatists have adapted the legend and dramatized a particular position, such as the position of Electra in Sophocles's *Electra*. Instead of taking one point of view, *Mirage* exposes the truth from different perspectives. The story introduces the user/participant as a new character called Archemedes, son of Aegisthus and Clytaemnestra. He was given away to a farmer until he can safely regain his throne. When his foster parents reveal his false identity, he goes to Argos to find out his true identity and regain his position as the prince. Since he was spirited away from the palace at an early age, he does not know the story behind the murders that made him prince.

At this point in the narrative, the participant takes the role of Archemedis, making choices in the story. As the story unfolds, he is confronted by Electra, Clytaemnestra, Aegisthus, and Orestes, learning their stories, their needs, and their fears. His actions define his position and subsequently define his relationships with others.

## B.3. CHARACTERS

### B.3.1 Archimedes (participant's character)

Archemedes was taken from his mother's hands when he was a baby. He was given to

a poor couple who raised and nurtured him. His parents decided to tell him the truth about his identity. They tell him that they are not his true parents, and that he is the son of Clytaemnestra and Aegisthus, the king and queen of Argos. They tell him that it was dangerous for him there; people tried to kill him seeing him as a competition for the throne. The king (his father) finally decided to take him away from the palace and give him to a couple of farmers to raise and nurture. Archimedes now finds himself determined to find out who his real parents are and the real reasons he was given away. Searching for who he really is, he ventures back to his birthplace to reclaim his throne, his family, and himself.

The participant assumes the role of Archimedes after he enters the castle. The participant's goals may vary from trying to become the prince or ruling the kingdom to just playing along. I anticipate that the participant's objectives will fall into the following set of objectives: become a prince, become a king and rule the kingdom, become a hero and rescue the kingdom from the curse that hunted it for years, find out the truth about the kingdom, find out the story and stick to what he/she thinks is the right thing to do regardless of the consequences, survive, or just play along.

### B.3.2 Electra

Electra has been hunted by the past – hearing about her father’s murder, witnessing her mother’s betrayal, and witnessing her brother exiled and outlawed was enough to devastate her. For years she has been trying to regain strength to avenge the injustice that befell her, her brother, and her father. She has been reduced to a slave in her father’s castle watched on every step. She wants to see her mother suffer for her father’s death. She wants revenge. She wants to watch her bother regain his kingdom. She is waiting for Orestes, her brother, to return and help her kill her mother and Aegisthus, and regain what is rightfully his. She doesn’t know of the existence of Archimedes yet.

### B.3.3. Clytaemnestra's Objectives:

Clytaemnestra knows the risk she took when she killed her husband, but it was an inevitable evil that she had to do for he killed her first born child, sacrificed her to regain Helen. When he went to fight for Helen, she found Aegisthus and found love. He helped her get rid of her husband (the King), and take her revenge. She is now the queen and Aegisthus is the king. However, she is haunted by, and in danger from, the past. Against all odds, she is going to fight for Aegisthus, for her kingdom, and for her life.

### B.3.4. Aegisthus

Aegisthus vowed on his father's deathbed that he will take the kingdom back. It was rightfully his father's. After killing Agamemnon and regaining the kingdom he felt peace, only for one single moment. He, like Clytaemnestra, is hunted by the past and by his actions. He wants to show the world that he is a good king, for it is his rightful place to be the king. Electra and Orestes always stood in his way. He got rid of Orestes, but Electra still roams the castle. He wants to protect the kingdom, himself, and his family from the eternal curse that surrounds them.

### B.3.5. Orestes

Like Electra, Orestes is motivated by revenge and justice. He was driven away from his own kingdom as an outlaw seeking refuge with farmers and outlaws. Life in exile was not easy. He is determined to make his mother and Aegisthus pay for such torment. Returning to the castle and helping his sister is the only thing on his mind.

## B.4. The clock

Most dramatic narratives have a clock, which defines the closure and builds the dramatic tension (McKee 97). In *Mirage*, I constructed the story such that the dramatic tension increases with the anticipation of Orestes' arrival. However, the interpretation of Orestes' arrival is different depending on the participant's choices and goals. For example, here are three different interpretation of this event:

- Orestes is coming to kill the participant's mother and father. The participant has to make a choice of whether he is going to stand by and let this happen or help his parents.
- Orestes is coming to kill the participant (Archemedis), because he stands in his way of getting the throne.
- Orestes is coming to help the participant kill his parents. The truth hour is approaching; the participant has to make sure that this is really what he/she wants to do.

The dialogue and beats of the story are assembled dynamically to keep the clock and use it to intensify the dramatic tension.


**Figure B.1. Choice Point**

## B.5. The Interface

The participant interacts in *Mirage* through mouse clicks and choices. He/she can click on different objects and choose a verb and adverb from a drop down menu listing all appropriate actions and adverbs depending on the object and the story. The menu automatically appears after the participant clicks on the object. If the object is a character, then a list of possible actions or speech verbs and adverbs will be presented. The list is restricted to a maximum of 4 words per entry. Figure B.1 shows a screenshot from a scene of *Mirage* at the choice point. It shows a menu with four speech choices for Archemedis.

## B.6. Interactivity Model

The participant's choices have a major impact on the story and the characters' actions and position in respect to the participant. The participant is presented with several choices involving relationships, justice, redemption, and family ties. The story system estimates the participant's intension given his action and the story situation. It then forms or adjust the participant's estimated goals and character. The system determines the character of the participant based on a four-dimensional character space: violence, self-interest, cowardice, and reluctant heroism. Furthermore, the system adapts the story by selecting story events and character behaviors that matches the estimated character.

# APPENDIX C

# RULES USED BY ELE

ELE uses many rules to select ideal values and cost parameters for the cost functions described in Chapter 5. These rules are implemented in *Java*. However, the authored rules are implemented in a language that sits on-top of *Lisp*. For the sake of simplicity, the rules described here are written in pseudo-code rather than raw *Java*.

The rules are divided into the following categories:

- Rules that select color parameters
- Rules that select angle parameters
- Rules that select defaults for light types, penumbra and umbra angles

## C.1. Selecting Color Parameters

ELE adjusts contrast depending on the dramatic intensity, the style of contrast established by the author, and author's previous patterns. ELE uses the following rules to modulate contrast:

```
if (changedDramaticIntensity())
```

```
{
        currentDramaticIntensity = getValueDramaticIntensity();
        ContrastType = getCurrentContrastType();


        SetIdealContrast(currentDramaticIntensity, ContrastType);
        // style is high in realism,
        // low in dramatic and expressionist
        if  ( (style.Realistic > 60) &&
              (style.Dramatic ≤ 30 ) &&
              (style.Expressionistic ≤ 30) )
                  setContrastCost (10);
        // style is high in realism,
        // average in dramatic and expressionist
        else if  ( (style.Realistic > 60) &&
                    (style.Dramatic ≤ 60) &&
                    (style.Expressionistic ≤ 60) )
                        setContrastCost (30);
             // style is high in realism,
             // high in dramatic and expressionist
             else if  ( (style.Realistic > 60) &&
```

```
                      (style.Dramatic > 60) &&
                      (style.Expressionistic > 60) )
                          setContrastCost (50);


// style is average in realism,
// low in dramatic and expressionist
If ( (style.Realistic ≤ 60) && (style.Realistic > 30) &&
           (style.Dramatic ≤ 30) &&
            (style.Expressionistic ≤ 30) )
                setContrastCost (30);
// style is average in realism,
// average in dramatic and expressionist
else if  ( (style.Realistic ≤ 60) && (style.Realistic > 30) &&
           (style.Dramatic ≤ 60) &&
            (style.Expressionistic ≤ 60) )
                setContrastCost (50);
      // style is average in realism,
      // high in dramatic and expressionist
      else if  ( (style.Realistic ≤ 60) &&
                (style.Realistic > 30) &&
```

```
                    (style.Dramatic > 60) &&
                    (style.Expressionistic > 60) )
                              setContrastCost (70);
// style is low in realism,
// low in dramatic and expressionist
if ( (style.Realistic ≤ 30) &&
     (style.Dramatic ≤ 30) &&
     (style.Expressionistic ≤ 30) )
          setContrastCost (40);
// style is low in realism,
// average in dramatic and expressionist
else if  ( (style.Realistic ≤ 30) &&
           (style.Dramatic ≤ 60) &&
            (style.Expressionistic ≤ 60) )
                setContrastCost (70);
     // style is low in realism,
     // high in dramatic and expressionist
     else if  ( (style.Realistic ≤ 30) &&
                (style.Dramatic > 60) &&
                 (style.Expressionistic > 60) )
```

```
setContrastCost (100);
```

The cost of the changing the contrast depends on the author's style. Style is composed values between 0-100 describing how much the artists care about realistic, expressionistic, and dramatic lighting considerations. Style is defined and maintained by ELE depending on author's settings. ELE uses the following formulae to modulate the style:

$$Realism = \frac{\sum_{n=t-k}^{t} \left( \lambda_{vc}^{n} + \lambda_{p}^{n} \right)}{k} \quad (1.31)$$

where $\lambda_{vc}$ and $\lambda_p$ is the cost how much the author cares about visual continuity and that the angle of light conforms to the general direction motivated by the practical sources, respectively. $t$ is the previous frame. Thus, the estimated realism style depends on the previous $k$ frames.

$$Dramatic = \frac{\sum_{n=t-k}^{t} \left( \lambda_{v}^{n} + \lambda_{c}^{n} \right)}{k} \quad (1.32)$$

where $\lambda_v$ and $\lambda_c$ is the cost how much the author cares about contrast and visibility of the action/characters, respectively.

$$Expressionism = \frac{\sum_{n=t-k}^{t} \left( \lambda_{e}^{n} + \lambda_{d}^{n} + \lambda_{m}^{n} \right)}{k} \quad (1.33)$$

where $\lambda_e$, $\lambda_d$, and $\lambda_m$ is the cost how much the author cares about mood, depth, and modeling, respectively.

## C.2. Modulating Visual Continuity

In addition, to maintaining and modulating the contrast ideal and cost, ELE also maintains and modulates the cost value associated with the visual continuity parameter. The cost of visual continuity is calculated as the average of visual continuity cost used in the previous $k$ frames, as follows:

$$\lambda_{vc}^{t+1} = \frac{\sum_{n=t-k}^{t} \lambda_{vc}^{n}}{k}$$

## C.2. Selecting Angle and Allocation Parameters

ELE selects costs for visibility, $\lambda_v$, modeling, $\lambda_m$, and motivation, $\lambda_p$, depending on the current dramatic situation and the style defined above. ELE uses the following rules to set these parameters:

```
// high realism

If  ( (    (CurrentShotType == Closeup ) ||

           (CurrentShotType == MediumClosup) ||

           (CurrentShotType == Medium) ||

           (CurrentShotType == Full) ) &&

            (Style.Realism > 60) )
```

```
    set λp= 100;
// average realism
If  ( (    (CurrentShotType == Closeup ) ||
           (CurrentShotType == MediumClosup) ||
           (CurrentShotType == Medium) ||
           (CurrentShotType == Full) ) &&
           (Style.Realism ≤ 60) &&
           (Style.Realism > 30) )
    set λp= 50;
// low realism
If  ( (    (CurrentShotType == Closeup ) ||
           (CurrentShotType == MediumClosup) ||
           (CurrentShotType == Medium) ||
           (CurrentShotType == Full) ) &&
           (Style.Realism ≤ 30) )
    set λp= 20;
// high dramatic
If  ( (    (CurrentShotType == Closeup ) ||
           (CurrentShotType == MediumClosup) ||
           (CurrentShotType == Medium) ||
```

```
            (CurrentShotType == Full) ) &&
            (Style.Dramatic > 60)  &&
            (EmphasizeVisibility[13](FocusCharacter()) )
      set λv= 100;
// average dramatic
If  ( (    (CurrentShotType == Closeup ) ||
            (CurrentShotType == MediumClosup) ||
            (CurrentShotType == Medium) ||
            (CurrentShotType == Full) ) &&
            (Style.Dramatic ≤ 60) &&
            (Style.Dramatic > 30) &&
            (EmphasizeVisibility (FocusCharacter()) )
            )
      set λv= 50;
// low dramatic
If  ( (    (CurrentShotType == Closeup ) ||
            (CurrentShotType == MediumClosup) ||
            (CurrentShotType == Medium) ||
```

[13] Emphasize visibility is a visual goal that is set using authored parameters. This goal is fired depending on the story situation, for example a character, x, is conveying information through gestures or facial expressions.

```
        (CurrentShotType == Full) ) &&
        (Style.Dramatic ≤ 30) )
    set λv= 20;
// Expressionism is High, goal is to emphasize evilness
If  ( (  (CurrentShotType == Closeup ) ||
         (CurrentShotType == MediumClosup) ||
         (CurrentShotType == Medium) ||
         (CurrentShotType == Full) ) &&
         (EmphasizeEvil14(focusCharacter()) &&
        (Style.Expressionism> 60))
    set λe= 100; //mood
    set moodAngle = UnderLight;
// Expressionism is High, goal is to emphasize mystery
If  ( (  (CurrentShotType == Closeup ) ||
         (CurrentShotType == MediumClosup) ||
         (CurrentShotType == Medium) ||
         (CurrentShotType == Full) ) &&
         (EmphasizeMystery15(focusCharacter()) &&
```

[14] Emphasize Evil is a visual goal that is set using authored parameters. This goal is fired depending on the story situation, for example a character, x, is evil if she/he is conveying evil intensions toward the participant.

```
        (Style.Expressionism> 60))
    set λe= 100; //mood
    set moodAngle = RimLight;
```

## C.3. Selecting Light Types and Penumbra and Umbra Angles

ELE selects light types for lights lighting the set based on the light sources.

For each light, *l*, lighting set piece, *s*:

```
If  ( (s.lightsource == Candle) || (s.lightsource == Torch) )
        l.lightType = point;
else l.lightType = spotlight;
```

ELE defines several spot light types, and sets umbra and penumbra values for these types based on film and theatrical instruments. For example, ERS40 and PARW are both types of lights used in theatre whose penumbra/umbra angles are 40/15, and 100/80, respectively.

ELE then associates an instrument to a virtual light depending on the area type, and the light source (if the virtual light is simulating a light source). Here are some rules that ELE uses to associate instruments/beam and field angles to virtual lights in the scene:

```
For each light, l
    If  ( l.assocArea().getAreaType == ForegroundArea)
```

[15] Emphasize Mystery is a visual goal that is set using authored parameters. This goal is fired depending on the story situation, for example a character, x, is hiding something from the participant.

```
    l.setinstrument(PARW);

if (l.assocArea().getAreaType== CharacterArea)

    l.setinstrument(ERS40);
```

# GLOSSARY- DEFINITIONS OF TERMS

**KEY LIGHT** Key light constitutes the main light source for the scene or for the subject lighted. It establishes direction and represents the source of light motivation in the scene. In addition, it determines the placements of shadows in the scene.

**FILL LIGHT** Fill light is often used to fill in the shadows created by the key or any unlit areas in the scene. There can be more than one fill light in a scene.

**BACK LIGHT** Back light is used to light the back of the subject. It is most often used to separate the subject from the background.

**RIM LIGHT** A rim light is a light placed behind the subject, but shifted to the side so that it skims along the side of the face. It is used to create shape, define texture, and separate the subject from the background.

**AMBIENT LIGHT**

Ambient light is an overall base directionless light. It is a method of lighting where every object in the scene is lit equally using a constant intensity, and where the light has no direction. In other words, objects are given luminance values, which is proportional to the intensity values (Möller and Haines 99).

**SET LIGHTS**

Set lights are lights used to light the architecture or the set of the scene. In film, they are most often called background lights.

**HARD LIGHT**

Hard light sources are small. They tend to be bright and produce dark and sharp-edged shadows.

**SOFT LIGHT**

Soft light sources are large sources and thus tend to wrap around the subject forming grayish shadows that have no distinct edges. The intensity of a soft light is lower than a hard light. Additionally, soft lights are directionless and they tend to be gentle.

**PRACTICAL SOURCES**

Local light sources found in the set of a scene, e.g.

torches or windows.

**LIGHT SETUP** A light setup consists of information including the number of lights used in a scene, their positions, orientations, and colors.

**CHARACTER MODELING** Character modeling, defined in glossary, is a term used by lighting designers for the effect of adding more lights on the character to establish depth and texture.