This is the author's version of the work. The definitive version was published in Thanos Vasilakos and Magy Seif El-Nasr. DigitalBeing: an Ambient Intelligent Dance Space. World Congress on Computational Intelligence: Fuzz-IEEE, 2006. Vancouver, July 16-21, 2006. Downloaded from the SFU Library institutional repository.

DigitalBeing: an Ambient Intelligent Dance Space

Magy Seif El-Nasr, Thanos Vasilakos

Abstract— ***DigitalBeing* is an ambient intelligent system that aims to use stage lighting and lighting in projected imagery within a dance performance to portray dancer's arousal state. The dance space will be augmented with pressure sensors to track dancers' movements; dancers will also wear physiological sensors. Sensor data will be passed to a three layered architecture. Layer 1 is composed of a system that analyzes sensor data. Layer 2 is composed of two intelligent lighting systems that use the analyzed sensor information to adapt on-stage and virtual lighting to show dancer's arousal level. Layer 3 translates lighting changes to appropriate lighting board commands as well as rendering commands to render the projected imagery.**

I. INTRODUCTION

Through the years, dancers have been trained to use their bodies as a media to express their inner feelings while improvising with music. Theatre dance expanded the basic set of expressive gestures adding lighting and scenery as part of the artistic expression. In the late 1990s, many artists and researchers explored the use of technology in dance spaces. One of the most popular ideas is the use of computer generated images and animations projected on the backdrop of a dance performance [1-4]. Alternatively, we seek to expand the set of expressive gestures in dance through the use of sensors and intelligent systems that control physical stage lighting as well as virtual lighting within projected imagery. We call the resulting artifact *DigitalBeing*.

We see the *DigitalBeing* as an ambient intelligent environment. Ambient Intelligence (AmI) is a vision that integrates concepts ranging from ubiquitous computing to autonomous and intelligent systems to compose an environment where technology is invisible and embedded in everything around us acting autonomously on our behalf responding to our needs. In such environments [5] electronics will be integrated into clothing, furniture, cars, houses, offices, and public places. We seek to extend this vision to a dance space.

In particular, dancers will wear wireless physiological sensors that measure arousal through skin conductance and body temperature. We will also place pressure sensors over the entire physical dance floor to collect dancers' locations and movements. We will develop an intelligent system that analyzes the physiological sensor signals computing a continuous signal representing dancer's arousal state. This system will also analyze pressure signals to identify on-stage lights affecting the dancer. This information will then be distributed to two intelligent lighting systems that adjust on-stage lighting and lighting of projected 3D imagery, accordingly. The idea is to express the arousal state of the dancers through light color within the stage as well as in virtual imagery projected on the backdrop.

We believe that using this ambient intelligent system will allow dancers to use their environment as an expressive medium. This takes dance to a new direction that has not been explored before. The contribution of this paper is in the conceptual framework discussed through the architecture of the ambient intelligent system. The system itself is still under development.

The paper will be divided into the following sections. In the previous work section, we will review related work. The following section will discuss the space and equipments used, including sensors and lighting equipment. We will follow this discussion with a description of the architecture. We will conclude by discussing contributions of the work presented as well as outlining some future work.

II. PREVIOUS WORK

Numerous composers, choreographers, dancers, and theorists have explored the use of technology in theatre and dance. We do not intend to describe all the work that has been done in the realms of academic research, installations, or interactive productions here. However, we would like to discuss few examples that have influenced our work. Discussing these examples will help situate our work, uncover its uniqueness, and its purpose in extending current work and techniques.

One of the most influential and significant work that used animated figures for choreography is the work of Merce Cunningham. In his dance performance '*Trackers*', he used a computer system called 'Life Forms' devised by Tom Calvert [6] to choreograph his dance movements. '*Life Forms*' is a piece of software designed to provide several stylized animated characters that allow users to choreograph a scene or explore certain dance steps.

M. Seif El-Nasr, Assistant Professor, Penn State University, 316E IST Building, University Park, PA 16802 (corresponding author to provide phone:814-865-6166; e-mail: magy@ist.psu.edu).

T. Vasilkos, Professor, University of Thessaly, Greece. (e-mail: *vasilako@ath.forthnet.gr*).

In addition to using animation for choreography, Cunningham also developed a virtual dance installation in collaboration with Paul Kaiser and Shelley Eshkar. This work was presented at Siggraph 1998. It was composed of a mental landscape in which motion-captured hand-drawn figures performed intricate choreography in 3D [1].

Besides the use of animated characters in a virtual performance, several performers have explored the use of animation within a real-life dance performance. For example, projected graphics has been used on backdrops in the San Francisco ballet '*Pixellage*' [2]. In one of the scenes they used a virtual animated ball (projected on the screen behind the dancers) which dancers threw to each other. Another ballet performance called '*The Catherine Wheel*' [3] used an animated character to represent the spiritual figure of Saint Catherine. By using an animated character, artists can easily represent the spiritual nature of the character as opposed to using real life effects or make-up.

Another example of the mix between virtual and real characters is in the work of Meador et al. [4]. They developed a collaborative production that mixes the use of virtual and real dancers within a dance stage. They used three different projectors within a dance performance; one of these projectors was used to project a virtual character that interacted with the dancers on stage [4].

This work was influenced by the work of Dan Saltz who directed *The Tempest* 2000 produced by the Interactive Performance Lab Group at University of Georgia [http://dpa.ntu.ac.uk/dpa_ search/result.php3?Project=136]. In this production of *The Tempest*, they projected the character *Ariel* as a virtual character. Motion capture equipment was used to animate the character in real-time. The use of a synthetic character for Ariel added to his magical quality, thus enhancing the overall performance.

Troika Ranch is a Dance company that explored the use of technology in dance. They developed a motion capture system called *MidiDancer*, which uses several cameras to capture performers' motions. They used this technology in several productions using projection video and motion capture to enhance the medium of expression for dance [http://www.troikaranch.org/].

Ulyate and Bianciardi showed their work on the Interactive Dance Club in Siggraph 1998 [7, 8]. The interactive dance club is composed of several pieces where they experimented with several setups and sensors, including infra-red, pressure, and cameras. They divided the dance floor into different zones which induced different interactivity paradigms. For example, in one zone they had a set of parallel light beams that detect when beams are broken. By breaking beams of light, participants can trigger 4-16 notes of musical phrases.

Similarly, Todd Winkler explored the use of gestures as a media for music composition [9-11]. He focused on the use of dance and space to compose electronic music. He used the *Very Nervous System* (VNS) [12-14] which is a system composed of one or two cameras that detect speed and location of dancers in a 3D stage. Using this as a data point, he then explored the use of algorithms to generate music [10]. He explored several methods of mapping the output data from VNS to musical parameters, such as frequency, pitch, and timber. He presented two productions in the late 1998 showing his system at work [11].

Several artists have explored the reversal problem of how to visualize music in a 2D or 3D projection used in a dance performance. Currently WinAmp and Windows Media Player both include built in algorithms that map music to 2D space using a frequency spectrum extracted from the music file [15, 16]. DiPaola and Arya explored the ability to extract emotional parameters from music and mapping them into a synthetic face [17]. Wagner and Carroll developed a 3D music visualization system called *DeepWave* [18]. *DeepWave* analyzes music files extracting frequency, pitch, vocals, etc. and maps them to shape, color, texture, and animations in 3D space. Through experimentation they found that vocals are best mapped to color and transparency, percussion to size and shape, and guitars and keyboards to animation. *DeepWave* also allows users to author skins and input 3D scenes and textures.

Beyond projection as a way to influence the dance space, Louis-Philippe Demers have explored the use of adjusting physical stage lighting within an art installation [19, 20]. He describes a system that uses several sensors including, pas sensors, video sensors, optical and infrared sensors, sonar sensors, and 3D ultrasound devices to predict blocking and gather gesture information. Using these as inputs, he developed a system that changes brightness or intensity, color pattern, and angles of on-stage lights. He showed this system in several projects, including *The Shadow Project* [21] and *Lost Referential* [22]. Our work follows the same idea. However, we will change light direction, color, and intensity based on a formal model of lighting design developed based on theatre and film design theories [23, 24]. We are also adapting the lighting to the arousal state of the dancer which was not explored in Demers' work.

I. The Dance Space

We envision a dance space similar to a proscenium theatre stage. Stage lights will be rigged on posts above the dancers. We will use a backdrop to project a 3D world sharing the theme of the dance performance. We will ask dancers to wear physiological sensors to track their internal state. We will also implant pressure sensors in the space to track dancers' positions and movements. The space will include a 3D surround sound system to play music and ambient sounds related to the projected virtual world. Dancers will be free to move around in the space. Sensor information will be transmitted wirelessly through a local network to a computer that analyzes the information and alters on-stage lighting as well as the projected 3D world in terms of its lighting to express dancers' arousal state.

A. Physical Equipment

For on-stage lighting, we will use physical lights that accept commands to rotate in 3D space and change color. There are several intelligent dynamic light models in the market today. These lights can be programmed to rotate. They also include a color wheel that allows artists to load up to 20 color gels. Through a light board artists can give commands to specific lights to direct them to change color at any given moment. The transition from one color to another will not be as smooth as in a virtual world, however. This is due to the fact that in virtual environments color is described using three digits representing RBG values. In a theatre production the color is represented by color gels with predetermined RBG values. Therefore, color mixing will be inflexible and difficult compared to a virtual world. Preparation and planning is needed at the pre-production stage.

Dancers will wear wireless physiological sensors; we opt to use SenseWear. SenseWear® PRO_2 Armband is a wearable body monitor that enables continuous collection of low-level physiological data: heat flux, skin temperature, near body temperature, and galvanic skin response. This device also includes a HealthManager web-enabled service that combines wearable computing, wireless connectivity, and online services, which is perfect for our purposes to allow dancers to freely move about. Using this device we will collect and transmit physiological data continuously.

In addition, we will use pressure sensors on the floor. For this purpose, we will adopt the design discussed in Srinivasan et al.'s work [25]. They designed a pressure sensor floor mat specifically for dancing. The floor will be constructed of several light-weight high resolution pressure sensor mats covering the entire stage. The actual number of sensors will range from 20,800 to approximately 100,000 sensors depending on the size of the stage. The sensors will be clustered together. They will send signals to a host computer that assembles and fuses this information to determine lights affecting the dancer.

B. Artistic Content

As with any performance, artistic content is important. Artistic content for this particular piece include: the 3D virtual environment to be projected within the performance, the actual music pieces composed for the performance, and the sound effects used in the performance. We also believe that artists should have control on the artistic change projected by the lighting. For this purpose, artists will direct the intelligent systems to compose changes that match the artistic vision. All intelligent systems developed in this paper will include tools that allow artists to author or dictate artistic goals that compose his/her artistic vision.

II. Architecture

The architecture is composed of several subsystems (shown in figure 1). The Sensor Analysis System analyzes two sensor signals: physiological sensor signals to identify the dancer's arousal state, and pressure sensor signals to identify lights relevant to dancer's positions. The arousal state will be stored in a structure called *Dancer Arousal State* represented in XML. The lights relevant to dancer's positions will be stored as a list of light IDs that are continuously changing as the dancer moves.

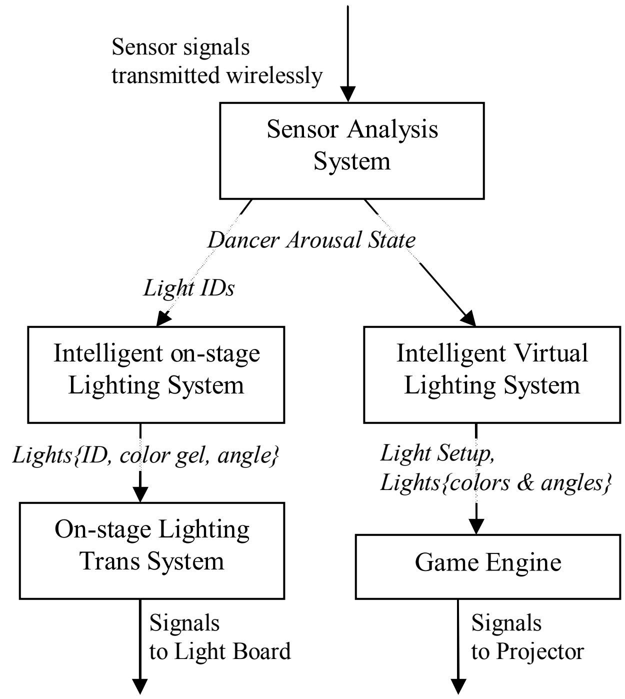

Fig. 1. Architecture of the System

The *Dancer Arousal State* will be passed to two systems: *Intelligent Virtual Lighting System* and *Intelligent on-stage Lighting System*. Both systems are based on our earlier work on Expressive Lighting Engine (ELE) [23, 24]. The *Intelligent Virtual Lighting System* uses virtual space info (provided by the artist at a preproduction stage), artistic constraints, as well as the *Dancer Arousal State* to compute a light setup if none exists. A light setup consists of the number of lights to use in the virtual environment and their placements. Once a light setup is calculated, the *intelligent virtual lighting system* determines color and angle changes for each light currently specified in the light setup. These changes will be determined to match the current arousal state of the dancer based on theatre and film lighting design theory [26-29], as will be discussed below. The light setup, colors, and angles will be given to a game engine to render the frame, which is then projected on the cyclorama.

Similarly, the *intelligent on-stage lighting system* determines colors and angles for stage lights given dancers' locations, *Dancer Arousal State*, and artistic constraints. It, however, determines color based on color gels specified in its database (which contains all gels uploaded by artist in preproduction and their color wheel location). It also does not generate a light setup. Instead, it categories lights on stage as: focus lights which are lights affecting dancers given by the list of light IDs (output of Sensor Analysis System), and non-focus lights, all lights not in the list of light IDs. Based on this difference, it computes, for each physical on-stage light, a color from the color wheel and an angle rotation. This information is then translated to light board hex code by the *On-stage Lighting Trans System* for the physical lights. For virtual lights, we will translate intelligent virtual lighting system outputs to rendering commands which are interfaced with a rendering engine.

Separating the lighting systems from their implementation (e.g. rendering engine and the translation to lighting board instruction) is important because it enables us to plug in other intelligent lighting systems without disturbing the implementations of the other layers, thus enabling modularity and reuse.

A. *Sensor Analysis System*

We will collect GSR and Body Temperature. These signals are continuous numerical values that capture arousal. We will pass these signals through a filter and will synchronize their readings and sampling rates. The output of this system is a signal identifying arousal in time stored in XML (*Dance Arousal State*).

Lights are rigged above the stage. Identifying dancer's position and movement is important to allow dynamic lighting control as discussed above. Instead of computing the lights affecting dancers automatically, we will manually map lights to specific mat numbers. Therefore, given a signal from a specific mat, the system can easily determine the lights affecting the mat. Using this method, we can determine which lights affect the dancers at any particular moment in time. This may be a crude technique; however, it will suffice as a first attempt to prototype this system.

The output of this system is a list of light IDs of lights affecting dancers at a specific moment in time. We anticipate this particular output to be continuously changing. Therefore, output from this system will be buffered and fed into the next layer for processing as a process within the next layer becomes available.

B. *Intelligent Virtual Lighting System*

This system is based on our earlier work on ELE [23, 24]. The *intelligent virtual lighting system* extends ELE to include a temporal modulation of lighting based on an arousal signal. Before I discuss the intelligent Virtual lighting system, I will briefly discuss ELE.

ELE, Expressive Lighting Engine, is an automatic intelligent lighting control system developed based on cinematic and theatrical lighting design theories; it is designed to automatically select the number of lights, their positions, colors, and angles. To accomplish this task, ELE uses lighting design rules formulated based on a study of film and theatre lighting. These rules are represented mathematically in an optimization function. The use of optimization is important to balance conflicting lighting-design goals. While adapting the lighting to the interaction, ELE also maintains visual continuity and style.

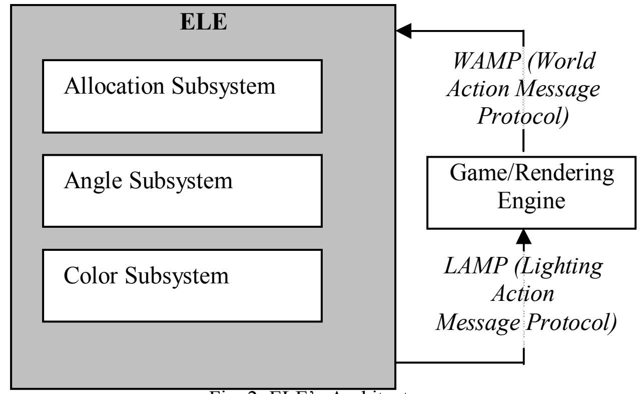

Fig. 2. ELE's Architecture

ELE as a black box is illustrated in figure 2. As shown, ELE takes in several parameters, represented as an XML structure called WAMP. These parameters are as follows:

- Stage layout or scene graph
- Locations of characters
- Local props that emit light, e.g. windows, torches, lamps
- Stylistic parameters including: low-key/high-key, overall contrast level, overall palette, specific ideal saturation, warmth, intensity or hue values for particular areas in the level or scene
- Dramatic intensity of the scene

ELE then emits an XML-based structure called LAMP, which includes the following:

- Number of lights to be used.
- For each of these lights:
  - type of instrument (e.g., spot light or point light)
  - color in RGB color space
  - attenuation
  - position as a 3D point
  - orientation including the facing and up vectors
  - range
  - masking parameters
  - Depending on the light instrument used, the Penumbra and Umbra angles.

These parameters are given to a rendering engine to render the frame.

As shown in the figure, to configure the lighting in the scene ELE is divided into three subsystems: allocation subsystem used to select the number of lights and their relative location based on the areas in the scene, angle subsystem which selects angles for each light, and color subsystem which selects colors for each light. I will discuss these subsystems briefly below.

Using theatrical and cinematic lighting design theories, ELE uses stage layout or scene graph information as well as artistic stylistic constraints to device a light layout. It divides the scene into *n* different cylindrical areas. It then categorizes these areas as: focus, describes the focus of the scene, non-focus, areas surrounding the focus area, and background areas. This is important because lighting designers often use light to bring out the focus, increase depth by varying brightness or color of lights in different areas, or increase contrast (determined by colors of lights lighting focus and non-focus areas). ELE determines where to direct viewers' attention (or the focus) given the characters in the frame.

By taking artistic style directions considering what the artist cares about, e.g. depth, motivation, contrast, etc., ELE optimizes a multi-objective function to determine the number of lights to use for each area. The function is as follows:

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

where p is light configuration, λ_v is the importance of visibility, λ_d is the importance of depth, λ_m is the importance of modeling, and λ_{vc} is the importance of visual continuity, and where $V(p)$ is visibility given p, $D(p)$ is depth given p, $M(p)$ is modeling given p, and $VC(p)$ is visual continuity given p.

We formulated a greedy algorithm that allocates 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; and
3. repeat step 2 until the number of lights assigned is less than or equal to the maximum.

Oftentimes, artists want their lighting design to reflect realistic directions. This desire can be encoded as an artistic direction that ELE then uses to determine angle of light. In determining the angle of light, ELE also takes into account the quality of light and their influence in projecting depth, modeling, and mood. ELE uses a non-linear optimization system based on hill climbing to select an angle for each key light that minimizes the following function:

$$\lambda_v (1 - V(k,s)) + \lambda_{-} \left| k - k^{-} \right| + \lambda_m \left| k - m \right| + \lambda_l \min_i \left| k - l_i \right|,$$

where k and s are defined as the key light azimuth angle relative to the camera and the subject angle relative to the key light, respectively, as shown in figure 3, k^{-} is the key light azimuth angle from the previous frame, λ_{-} is the cost of changing the key light angle over time (to enforce visual continuity), λ_m is the cost of deviation from the mood azimuth angle, m is the mood azimuth angle suggested by the artist, λ_l is the cost of azimuth angle deviation from a practical source direction, l_i is the azimuth angle of light emitted by the practical source i, and λ_v is the cost of deviation from an orientation of light that establishes best visibility.

Based on Millerson's [30] documented rules we formulated the following equation to evaluate the visibility and modeling of a given key light azimuth angle:

$$V(k,s) = \sin(k)\cos(s).$$

ELE uses rules based on Millerson's [30] guidelines to select fill and backlight azimuth angles depending on the value of the key light angle. According to Millerson's guidelines [30], fill light azimuth and elevation angles are calculated to be the mirror image of the key light angle. We define backlight azimuth angle as:

$$b = (k + \pi) \bmod 2\pi.$$

The interaction between colors assigned for each area in a scene composes the contrast and feeling of the entire image. Using the ideal values and their associated costs, ELE uses non-linear optimization to search through a nine-dimensional space of RGB values. It differentiates among focus colors, non-focus colors, and background areas to select a color for each individual light in the scene. It evaluates this color by using a multi-objective cost function, where each objective evaluates the color against the lighting-design goals, including establishing depth, conforming to color style and constraints, paralleling dramatic tension, adhering to desired hue, saturation, and lightness, and maintaining visual continuity. The cost function is defined as follows:

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

where

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

where c^t is a vector of light colors for focus f, non-focus n, and background b, and areas at frame t. Color c_i^t is represented in RGB color space; S(c) denotes the saturation of color c; H(c) denotes the hue of color c; L(c) denotes lightness of color c (in RGB color space).

ELE uses CIEDE2000, a well-known formula for measuring color difference [31, 32] as follows:

$$E = \sqrt{\left(\frac{\Delta L}{k_L S_L} \right)^2 + \left(\frac{\Delta C}{k_C S_C} \right)^2 + \left(\frac{\Delta H}{k_H S_H} \right)^2 + \Delta R},$$

where $\Delta R = R_T f(\Delta C \Delta H)$ and ΔL, ΔC, and 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.

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

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

where B are the indices for background lights; NB are the indices for non-background lights; and E is the color difference defined above.

Based on the results collected by Katra and Wooten described in [33], we used a multiple, linear regression method to formulate color warmth in RGB color space, as follows:

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

The optimization problem discussed above is a constraint-based optimization problem, where the color, c, is constrained to a specific space of values defined by style (e.g., realistic style restricts the values of saturation or hue). ELE uses a boundary method to bind the feasible solutions using a barrier function $v(x)$, such that $v(x) \to \infty$ as x approaches the boundary defined by the feasibility region. ELE uses the following formula for $v(x)$:

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

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.

The *intelligent virtual lighting system* extends this work to allow manipulation of lighting in time as a function of arousal. The general idea is to derive a dynamic technique that can set artistic stylistic constraints and vary dramatic intensity as subject to the arousal input from the sensors.

Based on a study of film and theater techniques [26, 27, 34-36], we formulated four major patterns for matching lighting changes to arousal. The decision of which pattern to use is left up to the artist. These patterns are as follows:

1. Arousal mapped linearly to brightness contrast between focus and non-focus areas, i.e. difference in brightness between colors of lights lighting focus areas and others lighting non-focus areas.
2. Arousal mapped linearly to warm/cool color contrast between focus and non-focus areas, i.e. difference in warm and cool colors of lights lighting focus areas and others lighting non-focus areas.
3. Arousal mapped linearly to saturation of colors.
4. Arousal mapped linearly to warmth of colors.

The output of this system is a light setup, if one did not exist. The light setup consists of number of lights and their layout. In addition, the system also determines angles and colors for each light in the light setup. This information is continuously changing in time as dancers move or their arousal state changes. Therefore, this continuous information will be passed to the rendering engine at a rate slower than 30 per second.

C. Rendering Engine

We have implemented a system that translates the *intelligent virtual lighting system* commands to three engines: Unreal Tournament, Ogre 3D, and WildTangent. The system accepts lighting commands from the *intelligent virtual lighting system* and invokes different methods in the engines that set the lights, their positions, their angles, and colors.

Figure 3 and 4 show screenshots where we fabricated an arousal signal; we used pattern 1 and 3 from the list above, results of which are displayed in figures 3 and 4, respectively. Figure 3 shows a simple 3D room rendered using WildTangent, where we linearly varied brightness contrast between the focal point (center of the room) and surrounding areas. The figure shows three screenshots of the room taken at different points during the transition. Figure 4 shows a first person shooter rendered using Unreal Tournament, where we mapped saturation level to the number of enemies in the scene (i.e. if number of enemies are high the saturation is high and vise versa). The figure shows four screenshots taken at different points within the game.

Fig. 3. Linearly increasing brightness contrast (where center of room is the focus)

Fig. 4. Linearly increasing Saturation

D. Intelligent on-stage Lighting System

Using actual stage lighting restricts changes in several ways. Continuous changes can only be made to brightness through the use of dimmers and to angle through rotation of the lighting devices. Since we are using a color wheel, continuous control of color is not feasible. Therefore, color changes will be predefined and made on certain cues or at specific thresholds. Actual color changes can no longer be made in terms of fluid changes in RGB color space as is the output of ELE. The color wheel contains twenty slots where artists can install different color gels. Therefore, the colors will be chosen apriori and will also be documented in the system.

The *Intelligent on-stage lighting system* reuses several components from the *intelligent virtual lighting system*, such as the color and angle systems. In addition to these two systems, the intelligent on-stage lighting system stores values of colors installed on the color wheels of each light with their Saturation, Hue, Brightness, and Warmth color values. In addition, it also stores the light setup given the rigged lights and their IDs.

Given the light IDs of lights affecting the dancers, the *intelligent on-stage lighting system* computes angles of each light affecting the dancer using the same angle sub-system as the one used in the *intelligent virtual lighting system*. Thus, using film rules to model the dancers with light. The other lights on the stage are set to a default angle that creates a wash on the stage.

Reusing the color subsystem, the intelligent on-stage lighting system computes RGB color values of each light category. The *intelligent on-stage lighting system* categorizes lights affecting the dancer as focus lights and other lights as non-focus lights. Given this categorization and the RGB color values and the list of lights and their color gels, we will develop another sub-system to determine the best color gel that matches the RGB color computed. This algorithm will loop through all color gels (20) calculating the color difference between desired (r_d, g_d, b_d) or (H_d, S_d, L_d) color and RGB color in the color wheel at index *i*, denoted by (r_i, g_i. b_i) or (H_i, S_i, L_i). The *i* with the least difference will be selected. In this case, the search space is constant, 20 color gels. To calculate the difference between the colors, we will use the color difference formula outlined [31, 32] as follows:

$$E = \sqrt{\left(\frac{\Delta L}{k_L S_L}\right)^2 + \left(\frac{\Delta C}{k_C S_C}\right)^2 + \left(\frac{\Delta H}{k_H S_H}\right)^2 + \Delta R},$$

where $\Delta R = R_T f(\Delta C \Delta H)$ and ΔL, ΔC, and 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.

The output of this system is a list of light IDs; for each light ID, an angle and a color gel location. This output is then passes through the *On-stage Lighting Trans system* which translates these commands to light board Hex code.

III. Conclusions and Future Work

In this paper, we have discussed a new ambient intelligent environment we are developing, targeting an adaptive interactive dance space that expresses dancers' inner feelings through manipulation of stage lighting as well as lighting of a virtual world projected around the dancers. The contribution of the paper is in the architecture presented which describes an ambient intelligent environment composed of several layers. Layer 1 is composed of a sensor analysis system that analyzes and synthesizes sensor information. Layer 2 is composed of two intelligent adaptive lighting systems that use the analyzed sensor information to adapt on-stage and virtual lighting to portray a temporal progression showing dancer's tension level. Layer 3 translates high-level lighting changes to appropriate lighting board commands as well as rendering commands for the software engine used to render the projected 3D imagery.

As it can be seen, the described system is work in progress. Future work includes integrating the sensor analysis system as well as implementing and testing the physical lighting component and integrating it to the system. Once this is complete, we plan to evaluate the use of this ambient intelligent environment within a dance floor with actual dancers, aiming at evaluating its aesthetic ability and its utility in exploring different forms of expression for interactive dance.

References

[1] M. Cunningham, P. Kaiser, and S. Eshkar, "Hand-drawn Spaces," presented at Siggraph 1998, Special Sessions, 1998.

[2] F. Crow and C. Csuri, "Music and Dance Join a Fine Artist and a Paint Machine," *IEEE Computer Graphics and Application*, pp. 11-13, 1985.

[3] J. Gruen, "Dancevision," in *Dance Magazine*, vol. 57, 1983, pp. 78-79.

[4] S. Meador, T. J. Rogers, K. O'Neal, E. Kurt, and C. Cunningham, "Mixing Dance Realities: Collaborative Development of Live-Motion Capture in a Performing Arts Environment," *ACM Computers in Entertainment*, vol. 2, 2004.

[5] A. Vasilakos and W. Pedrycz, *Ambient Intelligence, WIreless, Networking, Ubiquitous Computing*. MA, USA: Artech House Press, 2006.

[6] T. Calvert and S. Mah, "Life Forms: an Application of Computer Graphics to Support Dance Choreography," presented at Siggraph 96 Visual Proceedings: The art and interdisciplinary programs of Siggraph, New Orleans, LO, 1996.

[7] R. Ulyate and D. Bianciardi, "Interactive Dance Club," presented at Siggraph 98, Orlando, Florida, 1998.

[8] R. Ulyate and D. Bianciardi, "The Interactive Dance Club: Avoiding Chaos in a Multi Participant Environment," presented at CHI'01 Workshop New Interfaces for Musical Expression (NIME'01), 2001.

[9] T. Winkler, "Making Motion Musical: Gesture Mapping Strategies for Interactive Computer Music," presented at Proceedings of International Computer Music Conference, 1995.

[10] T. Winkler, "Creating Interactive Dance with the Very Nervous System," presented at Proceedings of Connecticut College Symposium on Arts and Technology, 1997.

[11] T. Winkler, "Motion-Sensing Music: Artistic and Technical Challenges in Two Works for Dance," presented at Proceedings of the International Computer Music Conference, 1998.

[12] D. Rokeby, "Very Nervous System," 1986.

[13] D. Rokeby, "Very Nervous System," Rokeby, David, 1986-1990.

[14] D. Cooper, "Very Nervous System," in *Wired*, 1995.
[15] "Windows Media Player."
[16] "WinAmp."
[17] S. DiPaola and A. Arya, "Affective Communication Remapping in MusicFace System," presented at Proceedings of European Conference on Electronic Imaging and the Visual Arts, London, England, 2004.
[18] M. G. Wagner and S. Carroll, "DeepWave: Visualizing Music with VRML," presented at Proceedings of the Seventh International Conference on Virtual Systems and Multimedia (VSMM '01), 2001.
[19] L. P. Demers, "Interactive and Live Accompaniment Light for Dance," presented at Dance and Technology Conference, Simon Fraser University, Vancouver, 1993.
[20] L. P. Demers and P. Jean, "New Control Approaches on Lighting," presented at Shadow Light '97, Flemish Opera House, 1997.
[21] J. Crawford, T. Schiphorst, M. Gotfritt, and L. P. Demers, "The Shadow Project," presented at Symposium on Arts and Technology, 1993.
[22] L. P. Demers and B. Vorn, "Lost Referential." New York, USA: The Great Hall of New York Hall of Science, 1998.
[23] M. Seif El-Nasr, "Intelligent Lighting for Game Environments," *Journal of Game Development*, vol. 1, 2005.
[24] M. Seif El-Nasr and I. Horswill, "Automating Lighting Design for Interactive Entertainment," *ACM Computers in Entertainment*, vol. 2, 2004.
[25] P. Srinivasan, D. Birchfield, G. Qian, and A. Kidane, "A Pressure Sensing Floor for Interactive Media Applications," presented at ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE), Valencia, Spain, 2005.
[26] J. M. Gillette, *Designing with Light*, 3rd. ed. Mountain View, CA: Mayfield, 1998.
[27] J. Alton, *Painting with Light*. Berkeley: University of California Press, 1995.
[28] J. Birn, "Digital Lighting & Rendering," G. Maestri, Ed. Indianapolis: New Riders, 2000.
[29] B. Block, *The Visual Story: Seeing the Structure of Film, TV, and New Media*. New York: Focal Press, 2001.
[30] G. Millerson, *The Technique of Lighting for Telivision and Film*, 3rd ed. Oxford: Focus Press, 1991.
[31] B. H. Hill, Roger, T. , and Vorhagen, F. W., "Comparative Analysis of the Quantization of Color Spaces on the Basis of the CIELAB Color-Difference Formula," *ACM Transactions on Graphics*, vol. 16, pp. 109-154, 1997.
[32] M. R. Luo, Cul, G., and Rigg, B., "The Development of CIE 2000 Colour Difference Formula: CIEDE2000," vol. 2000: CIELAB, 2000.
[33] E. a. W. Katra, B. R., "Preceived lightness/darkness and warmth/coolness in chromatic experience," 1995.
[34] D. Campbell, *Technical Theatre for Non-technical People*: Allworth Press, 1999.
[35] S. Calahan, "Storytelling through lighting: a computer graphics perspective," presented at Siggraph Course Notes, 1996.
[36] B. Brown, *Motion Picture and Video Lighting*. Boston: Focal Press, 1996.