Software for Choreography:
Real-Time Analysis of Expressiveness in Dance Performance

A Junior Independent Work Report

Written by Adam Stasiw

Under Advisor Professor Rebecca Fiebrink

In partial fulfillment of the requirements for a Bachelor of Science in Engineering degree in the

Department of Computer Science, Princeton University

May 8th, 2012

Abstract

Integrating various technical systems into creative performances, especially dance pieces, is a compelling challenge that has become more prevalent over the past century. Since creative and technical teams must collaborate to produce a final work that is not only technologically possible but also creatively meaningful, one of the more common practices is to utilize motion capture technology to analyze the movements of the dancers themselves and use the resulting information to directly influence the performance. However, limitations to the current systems include the potential for high costs, intrusive or restrictive equipment that inhibits the freedom of dancers, and the fact that many techniques cannot be applied in real-time. As such, I here present a system developed with Microsoft Kinect, at a relatively cheaper $150, to dynamically analyze the qualities of dancers' movements without interfering with their range of motion. Various mathematical equations written in C# derive relevant features from the low-level joint position data provided, and this information is fed through the Wekinator machine learning suite [1] to train algorithms to classify the expressiveness of movements; since emotion is too subjective of a parameter, the quality of motion is quantified by the four Effort dimensions of Laban Movement Analysis: Weight, Space, Time, and Flow [2]. While the resulting classifications are now sent to a Max/MSP patch for visualization, the information is transmitted in the universal Open Sound Control (OSC) format [3], which could easily be routed to other systems, such as those that control the light and sound elements of dance performances.

1. Introduction

As technology has become ever more prevalent in our everyday lives over the last century, artists in various fields have begun to take note and integrate computational systems into their creative works. This is especially prevalent in the realm of performance, even in the most basic modern-day inventions of programmable light and sound boards that flesh out the reality of a piece on stage. Beyond these common elements, certain artists have even incorporated things such as video projections, animations, or even robots into their productions. This is especially common in the realm of dance, where various choreographers, most notably in more contemporary pieces, have brought technical elements into their works in order to further explore the relationship between the dancer and his environment. Some specific examples of this trend include the piece “Seraph” by Pilobolus [4][1], in which a single dancer interacts with two flying robots, or Chunky Move’s “Mortal Engine” [5], which incorporates a full multimedia landscape in which the dancers interact with light and sound in novel ways. Clearly, then, new possibilities for unique and dynamic choreographic works have emerged as a direct result of advancements in the realm of computation.

In particular, one technological application that has become increasingly popular is that of using motion capture techniques to acquire, analyze, and employ datasets of motion information from the performers themselves. The lower-level gesture information acquired can be used to control lights, sounds, videos, or nearly any other technological system. Typically, this approach requires dancers to wear special bodysuits equipped with sensors or reflectors to transmit movement data to the system [6] [7]. While these setups can and do provide interesting and useful information to artists, they are suboptimal because of their cost, which can be above the tens of thousands of dollars for a full motion capture studio, and the restrictions they can place

[1] See also the company’s webpage at www.pilobolus.com

on the dancers themselves, such as in systems that require performers to wear gesture-tracking body sensors. Additionally, these systems often cannot be used in real-time during the performance; instead, the analysis is often performed off-stage and then somehow incorporated into the show at a later time. As such, the existing approaches to utilizing motion capture data in relevant and interesting ways are limited by cost, restrictiveness, and time.

I therefore propose here a system that is cheap, easy-to-use, and dynamic for analyzing expressiveness in dance performance. As will be detailed in later sections (Section 4 – My Solution: The XBOX Kinect), my system costs less than $150, does not interfere with dancers or the creative process in any way, and can be used in real-time to send output parameters to various technical elements of the show. This is accomplished through an interpretative process that converts the low-level position data acquired into relevant mathematical descriptors (Section 5 – Motion Feature Equations), which can then be better correlated to expressiveness, as represented by the Effort dimensions of Laban Movement Analysis (Section 3 – Laban Movement Analysis). This system provides a simple and effective means (Section 6 – System Analysis) for producing meaningful artistic output from the very actions of the dancers in the piece. Through my contributions to the field, it is clear that even though my methods and technical handling of the motion capture information do not deviate from established techniques (Section 7 – Related Work), my system as a whole advances the subject area and provides new opportunities for creative works to incorporate technology into performance (Section 8 – Conclusions and Future Work).

2. Background

As discussed above, current motion capture systems possess limitations that restrict their usefulness in creative projects; specifically, technologies used for analyzing expressiveness in dance performance are often suboptimal or inadequate in terms of providing flexible, easy-to-use systems that can dynamically influence the content of a show. One of the more recent systems, discussed in Jessop's work [6] and used in Tod Machover's multimedia opera, "Death and the Powers" [8] [7], involves the use of body sensors that transmit gesture information in real time. Although these tools can therefore solve the problem of requiring analysis after the fact, they are often still expensive, and require being created or tailored specifically for the application at hand. Additionally, these sensors, and those used in other systems, typically have a limited range of relevant information that can be acquired; for example, Jessop's system can be trained to recognize gestures, but has no capabilities for determining the position of a body in space. Even beyond these limitations, such sensors can also interfere with the freedom and artistic capabilities of performers and choreographers, as wearing such equipment requires a dancer to adapt his movements to work with the technology.

Besides these issues of cost, complexity, and artistic limitations, we must also consider the different possibilities for how to actually evaluate the expressiveness of a performer. That is, as the system acquires movement data, in what ways should we analyze and manipulate that input information to derive meaningful approximations for the qualities of motion in the dance? Emotion, measured either directly (i.e. by classification of movements into categories such as Happy, Sad, Angry, etc.) or on a Valence-Arousal plot, has been used as a measurement in other works [9] as the output of expressiveness-tracking systems. However, since emotion is such a complex and subjective phenomenon, I believe that such a classification system would fail to perform as accurately as one using a different measurement. To reinforce this point, we can observe that even a single performance can produce widely varying responses among individual audience members, and so deriving a specific emotional descriptor for a dance or sequence of motions would be difficult for humans, let alone for any machine learning techniques. Furthermore, we must also consider the fact that it is possible to have an evocative and expressive series of movements that does not convey a particular emotion but is perhaps more abstract instead; for example, the choreography of Merce Cunningham often involves movement for its own sake, as he sees “all movement as dance” [10]. As such, we choose not to directly track emotion as the feature of expressiveness measured by our system.

We therefore require some other set of features that will allow our system to accurately convey a sense of the qualities of movement without relying directly on an interpretation of emotion. Mapping movements to a 2-dimensional plot of Valence and Arousal is one technique used to approximate emotion, but, as noted before, such mappings are often subjective and difficult to program; specifically, ascertaining such information is not ideal given the low-level movement information provided by motion capture systems, which generally only return joint positions. As such, I have found that a suitable and quantifiable mapping layer for this project involves the Effort dimensions of Laban Movement Analysis. These features, described in further detail in the next section, can easily model the expressiveness of movement directly from low-level inputs, and they allow the system to reliably and effectively classify gestures in a way that is consistent and significantly more objective.

3. Laban Movement Analysis

We have therefore seen that emotion is too subjective a feature to reliably calculate from the low-level information attained through motion capture techniques. As such, the system requires an output layer, comprised of features that can be efficiently and accurately derived from the acquired movement data, to be used in determining the expressiveness of gestures. Such features could ideally be used not only to classify the emotions or valence-arousal values of a dance but also to serve as meaningful representations of expressiveness in and of themselves.

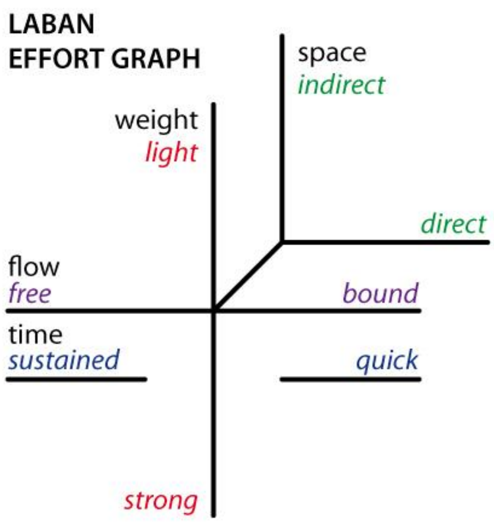

Figure 1 A 4-dimensional graph of the Laban Effort dimensions.
From R. Cottin on Wikimedia Commons.
Accessible at http://commons.wikimedia.org/wiki/File:Laban-effort-graph.jpg

The solution to this issue of describing the qualities of movement can be found in the teachings of Rudolf von Laban, a Hungarian dancer who developed a unified theory of movement that detailed ways in which dance could be described qualitatively [2]. This Laban Movement Analysis (LMA) has been used throughout the past century to help describe the properties of movement in all aspects of life, but it is especially useful in attempting to convey the qualities of motion in the context of performance. The expressiveness of dance and gesture can be especially well qualified using the idea of Effort. In LMA, Effort is a four-dimensional set of features, as shown in Figure 1, that describes the characteristics of movement; every gesture can then be described in terms of a value for the four Effort dimensions, Weight, Space, Time, and Flow. These values range from Strong to Light, Direct to Indirect, Sudden to Sustained, and Bound to Free, respectively.

Weight is a measurement of the “impact” of movements [11] and can also be thought of as a relationship to gravity [2] or “a measurement of how much energy is being put into the movement” [6]. For example, movements with Heavy Weight tend to have a lot of force and strength such that a lot of energy is invested in the movement and any impact made would be significant. On the other hand, Light Weight motions are more gentle and delicate, and may appear more airy and free given their softer impact.

Space, when considered in terms of Effort dimensions, can be thought of as a representation of the direction and path of a movement. That is, a motion that is Direct in Space will have a more linear feeling in that its focus is on moving towards a specific destination point in a clear and exact trajectory. Indirect movement will instead feel more angular and wavy in that the focus of motion is not to reach a certain point but rather to move through the surrounding area in a less precise path.

Time is a measurement of the duration of movement in LMA, and motions can be qualified as either Sudden or Sustained. Sudden movements are quick, jerky, and perhaps unexpected, as if there are abrupt shifts in the purpose of motions. Contrary to these rapid changes, Sustained gestures feel more held out and stretched in terms of their ease and fluidity, and there is a general sense of continuity in the legato nature of such movements.

Flow is perhaps the least intuitive of the four Effort dimensions, but can be understood quite easily in terms of the fluidity and tension of movements [2]. That is, Free movements tend to be carefree and loose, with a sense of relaxation and the possibility for various possible motions to follow the current gesture. Bound movements, however, involve a muscular tension along with a sense of being restricted, as if the potential for further motion is limited.

Now that we have described these four Effort dimensions, it is easy to see that any motion can be distinctly represented by a point in the 4-D space formed by considering each quality along an axis. This notion is supported by the fact that these dimensions can take on values in a continuous range between their extremes, and are not simply limited to, say, Heavy and Light. For example, a motion could be described as having a Weight quality that lies between these two extremes if it were neutral in terms of that dimension by being neither delicate (Light) nor forceful (Heavy) in its impact. However, since it is difficult to ascribe a real-valued classification to an Effort dimension (for example, consider how difficult and odd it would be to attempt a gesture of precisely 0.687 Weight on a scale of -1 to 1), we can justifiably make the simplifying assumption that movements are classified in a binary fashion (i.e., either Light or Heavy, but with no specific value for the Weight dimension).

It is important to note that it is rather difficult to isolate Effort dimensions from each other when considering actual movements. That is, every gesture or motion has a corresponding classification for each of the four dimensions, and so focusing on one specific trait while holding the other three neutral would be extremely challenging, if not altogether impossible. To demonstrate this point, try to imagine a movement that has Heavy Weight, but is altogether Spaceless, Timeless, and neutral in Flow; given how difficult it is to conceive of such an isolated motion, we can now feel confident in our assertion that the four Effort dimensions are closely interrelated.

However, even given this closely correlated nature of the four dimensions, it is possible to maintain one (or perhaps two) of these qualities as neutral and consider the range of gestures that are possible when the other dimensions are allowed to vary. In LMA, such sets of movements are defined as Drives, which are the collections of movements defined by the absence of a single Effort dimension. For example, Vision Drive is the realm of motions that are Weightless but that have specific qualities of Flow, Space, and Time; within this Drive, since actions are neutral in Weight, they tend to have an "external" feeling associated with a sense of "planning" or "envisioning" [2]. Aside from Vision, the other three Drives are Action, Spell, and Passion, which correspond to a lack of Flow, Time, or Space, respectively. Of particular interest is the

Action Drive, which contains the set of motions that are "task-oriented" given that they are well-defined in terms of Weight, Space, and Time without respect to Flow; that is, the movements in Action Drive can be considered the quintessential gestures that humans can perform without taking intent or feeling into consideration [2]. Some specific examples include the Punch action (Heavy, Direct, Sudden) and the Float action (Light, Indirect, Sustained), although there are a total of eight different gestures corresponding to each possible combination of classifications for the three Effort dimensions excluding Flow.[2]

4. My Solution: The XBOX Kinect

After considering all of the limitations of the current systems in the field of affective analysis of dance performance, I found a solution in Microsoft's XBOX Kinect. This tool is a cheap ($150) peripheral originally designed for skeletal tracking and gesture recognition for various video games, but the potential of this camera to be used for other applications was quickly realized by programmers and computer enthusiasts around the world. Various open source frameworks emerged for utilizing the information from the color camera, depth sensor, and skeletal tracking abilities of this device, and Microsoft soon released their own official Kinect SDK in response to its popularity. This SDK integrates with Microsoft Visual Studio 2010 on Windows 7, allowing for programs written in C#, C++, or Visual Basic to interact directly with the data acquired through the device. For my system, I decided to use C# as my programming language given its similarities to Java, with which I was already familiar. In order to establish the basics of getting input data from the Kinect camera, the majority of my code for connecting the Kinect and attaining new motion information was copied directly from the code provided by Dan Fernandez in his "Skeletal Tracking Fundamentals" program [12].

---

[2] Interested readers are encouraged to find more information about LMA, Effort, and the various Drives from sources online such as http://www.lmaeffortbank.com [2]

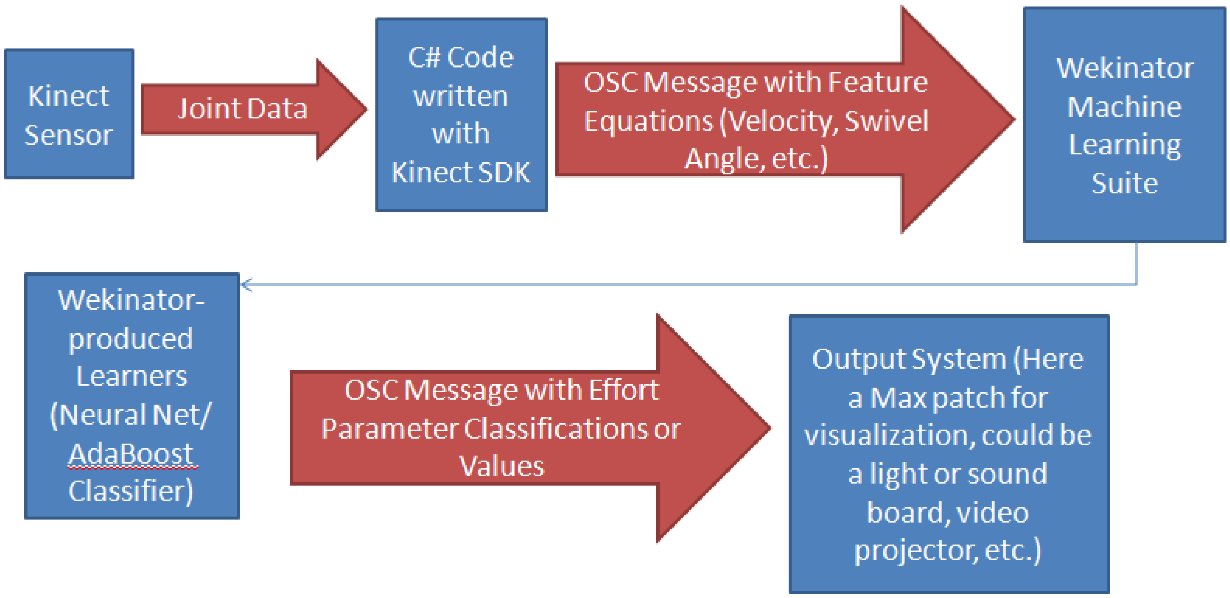

Figure 2 This architecture diagram represents the flow of information through the Kinect motion analysis system.

Using the APIs provided with the Kinect SDK, I designed a system, outlined in Figure 2, to derive meaningful information from the joint data provided by the motion capture camera. Since the Kinect peripheral automatically tracks the X, Y, and Z coordinates of 20 joints of the human skeleton accurately and efficiently, I was free to come up with ways to use that data instead of having to worry about how to track individual dancers. My approach was influenced by the ideas of Jessop, Zhao, Badler, and others in their various research papers [6] [11] [13], which are discussed in more detail in the Equations and Related Work sections; from their research, I designed my system to derive various mathematical features that could then be related to the LMA Effort qualities previously described. These models contain information ranging from the velocity and acceleration of joints to the swivel angles of both elbows.

Since expressiveness is a complicated facet of movement, I looked to find ways in which to automate the analysis process as much as possible. I therefore decided that while I could accurately reason about which mathematical features would be significant in the analysis, my ability to then determine *how* those features affected the Effort dimensions would be significantly worse than the ability of established computer learning algorithms to perform that task. That is, the code written with the Kinect SDK calculates the various mathematical components of motion, but those values are then sent to a separate machine learning suite, the Wekinator, for classification [1]. Having the Wekinator perform the regression algorithm on my dataset and classifying the Effort dimensions with an automatically generated neural network or discrete classifier removes the human error involved with trying to assign significance to various movement features, and enables the machine learning algorithm to make predictions and connections about the subtle complexities of movement that I would not be able to make on my own. Taking this approach therefore enabled me to calculate the significant features of movement and manipulate which were sent to the classifier in order to better test the performance and accuracy of the system, as described in the Analysis section, without needing to write my own neural network or AdaBoost code. Additionally, this automation in the learning process enables the system to be built and trained with a dataset of movement information collected in the lab (Section 6 – System Analysis) before being used dynamically in a performance setting. Ultimately, then, the constant flow of low-level Joint position information from the Kinect through the C# feature equations on to the regression algorithm created by the Wekinator enables our system to be used in real-time, as the Effort dimension classifications are made at the same time as the Kinect is acquiring the next frame of movement information from the dancers.

This approach is justified because of the inherent complexity of motion, especially with respect to expressiveness and the subtleties involved with the layered qualities of dance as described with Effort features. Since it is nearly impossible to have a movement that is purely performed in terms of one Effort dimension, as discussed previously, we can see that dance motions involve an interrelationship of various qualities; this then translates into the fact that each movement feature sent as input to the Wekinator can influence the classifications of multiple Effort dimensions in potentially complicated ways. That is, there is not a simple one-to-one correspondence between the value of a certain motion feature equation and the Effort classification of the movement that would result. For a specific example, we can see that something as simple as the acceleration of a joint can influence the Flow dimension, but that same acceleration value could also have an impact on the Weight classification, especially given the fact that we are considering Weight to be related to the amount of energy associated with a movement. Therefore, the relationship between the mathematical motion feature inputs and the Effort dimension outputs is more complicated than could be determined manually, and so using the Wekinator to automatically analyze the input data and discover connections between that input and the resulting output classifications is a justified and intelligent design decision.

It is also significant to note that both the transfer of motion features from the C# code to the Wekinator and the resulting output of LMA Effort classifications take place through Open Sound Control (OSC) messages [3]. These messages comprise a standard format for data interchanges that enable the Kinect data and computed features to be sent from C# through the Ventuz OSC wrapper to the Wekinator, which then outputs the Effort dimension values in OSC messages; in my system, the code to translate the C# motion features into OSC messages for use by the Wekinator was copied directly from the source code written by Jordan Rogers-Smith [14]. For this project, I used Max/MSP, a programming language designed for use in the fields of music and the arts, to visualize the resulting LMA Effort classifications (Section 6 – System Analysis).

While the resulting Max/MSP patch used for demonstration in this system is useful, various other systems could be manipulated with the resulting Effort data. For example, these output messages could be picked up by countless other systems, such as those that control lighting or sound in onstage performances, for dynamic artistic opportunities involving the real-time manipulation of performance elements based on the expressiveness of the performers themselves.[3]

5. Motion Feature Equations

In order to more accurately determine values for the LMA Effort dimensions corresponding to a dancer's movements, I needed to find various equations that could convert the low-level information about joint positions into more descriptive models of motion. I found such a collection of equations in Zhao's paper [11], wherein he effectively represents facets of movement with mathematical features that can be better correlated with the LMA Effort dimensions. Using his equations, which I copied directly to implement in my C# code, I was then able to use the basic skeletal tracking data acquired from the Kinect to calculate useful features of the dancer's movements. This section therefore borrows heavily from Zhao's work, both in terms of the equations themselves and of the descriptions of the formulas and how they correspond to Effort dimensions.

Intuitively, velocity and acceleration of the 20 joints are more relevant to the qualities of gesture than simply the location of a dancer's body in space, and so these first and second derivatives of position were approximated as described below. In addition, the third derivative of position, the jerk, is calculated both for its use in the torsion equation detailed further on in this section and for its usefulness as an input feature in and of itself.

[3] A particularly significant piece of related work is "Death and the Powers," [8] [7] which is described in further depth in the Related Work section.

Let $Position_{i,t}$ represent the X-coordinate of the position of Joint i at the time corresponding to frame t. Then let $\Delta t = \frac{1}{Frames\ per\ Second} = \frac{1}{30}$ represent the time interval between two frames captured by the Kinect skeletal tracker. This leads to the following equations for the velocities and accelerations of joints (analogous equations exist for the Y- and Z-coordinates):

$$Velocity_{i,t} = \frac{(Position_{i,t} - \ Position_{i,t-1})}{\Delta t}$$

$$Acceleration_{i,t} = \frac{(Velocity_{i,t} - \ Velocity_{i,t-1})}{\Delta t}$$

$$Jerk_{i,t} = \frac{(Acceleration_{i,t} - \ Acceleration_{i,t-1})}{\Delta t}$$

While the position of joints is still important to consider,[4] these velocity, acceleration, and jerk equations are more useful in determining the Effort dimensions since there is more direct correlation between these values and the qualities of movement. For example, if a dancer is moving abruptly and changing the direction of his joints in a rapid way, he will be moving with Sudden Time; this Effort classification is therefore closely related to the acceleration and jerk of his skeleton. Zhao also mentions that having a large number of accelerations and variations in velocity corresponds to a Free Flow, since Bound Flow movements would be less likely to be changing in speed or direction during a gesture. [11]

In addition to these more basic descriptors of the dancer's movements, Zhao also presents some more complicated feature equations, such as the curvature and torsion of a joint's trajectory and the swivel angle of his elbows [11]. The first of these equations, the curvature, measures the change of the tangent of a joint's trajectory over time. A related concept, the torsion, measures the extent to which a joint twists along its path. They are defined as follows, using the 3-

[4] For example, as mentioned in Zhao's paper [11], the height of a dancer's sternum can be indicative of the Weight dimension, as having a lower sternum tends to correlate to a heavier sense of movement.

dimensional vectors of velocity ($\hat{v}$), acceleration ($\hat{a}$), and jerk ($\dot{a}$) of joints in each of the X, Y, and Z axes.[5]

$$Curvature_i = \hat{v}_i \times \hat{a}_i = \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ \dot{x} & \dot{y} & \dot{z} \\ \ddot{x} & \ddot{y} & \ddot{z} \end{vmatrix}$$

$$Torsion_i = \frac{\begin{vmatrix} v_i(x) & v_i(y) & v_i(z) \\ a_i(x) & a_i(y) & a_i(z) \\ \dot{a}_i(x) & \dot{a}_i(y) & \dot{a}_i(z) \end{vmatrix}}{\|\hat{v} \times \hat{a}\|^2}$$

Since these equations are related to the trajectory of movement and the degree to which motions curve and twist along a path, it is evident that they correspond to the Space dimension; that is, a gesture with a high degree of curvature or torsion is much more likely to be Indirect than Direct given that the movement is not as linear in terms of direction towards a destination. Curvature and torsion can also relate to the Weight of a movement [11], given that a motion with more twisting along its trajectory is likely to be less forceful and less focused, thereby presenting a Light Weight gesture with less of an impact.

Perhaps the most complicated of the equations included in this system, the swivel angle of the dancer's elbows, is a useful mathematical feature that helps to quantify, among other things, the Space and Flow of movements. This correlation makes sense because both Indirect and Free movements will tend to have a greater fluidity and range of motion in the elbow as compared to Direct or Bound gestures, which are more likely to involve rigidity in the elbows and fewer changes in the swivel angles of the arms.

[5] Equations copied directly from Zhao's paper [11], with descriptions and explanations of relevance to Effort dimensions paraphrased from that work.

Intuitively, the swivel angle of the elbow corresponds to the location of the elbow relative to the vector connecting the shoulder and wrist joints. We define this angle more precisely with the following diagram, taken from Zhao's paper [11].

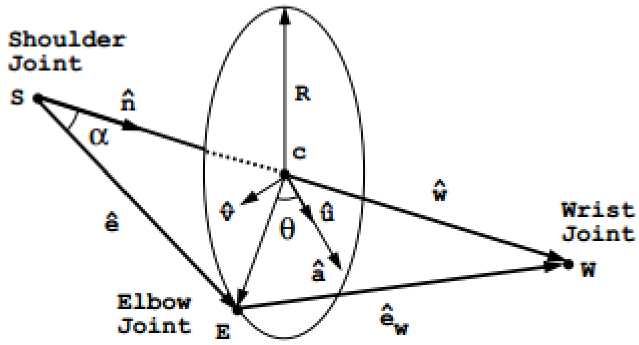

Figure 3 This diagram displays the vectors and the swivel plane that are related to the swivel angle, θ

Let $\vec{S}$ represent the swivel plane, which is defined by the normal vector $\hat{n}$ along the path from the shoulder joint to the wrist joint and containing the elbow joint, at position E. We then define the vector $\hat{a}$ to be the projection of $-\hat{k}$, the unit vector pointing along the negative direction of the Z-axis (that is, towards the floor), onto $\vec{S}$ [15]:

$$\hat{a} = \frac{-\hat{k} + (\hat{k} \cdot \hat{n})\hat{n}}{\|-\hat{k} + (\hat{k} \cdot \hat{n})\hat{n}\|}$$

The vector $\hat{b}$ is then defined as the vector from c, the center of $\vec{S}$, to the elbow position at E. If we then let $\hat{e}$ represent the vector from the shoulder to the elbow and $\hat{c}$ represent the vector from the shoulder to the center of the swivel plane, we can compute $\hat{b}$ according to the following:

$$\hat{b} = \hat{e} - \hat{c} = \hat{e} - \cos(\alpha)\|\hat{e}\|\hat{n}$$

In the proceeding equation, α is the angle between the shoulder-to-wrist vector, $\hat{n}$, and the shoulder-to-elbow vector, $\hat{e}$. Now, if we let $\hat{e}_w$ represent the vector from the elbow joint to the wrist joint, we have the following trigonometric equations copied from Zhao's paper to compute $\cos \alpha$:

$$\cos \alpha = \frac{\widehat{w}^T \widehat{w} + \hat{e}^T \hat{e} - \hat{e}_w^T \hat{e}_w}{2\|\widehat{w}\|\|\hat{e}\|}$$

Then, once this value has been used to compute $\hat{b}$, we use trigonometry and the definitions of the cross and dot products to derive the following equation for θ, the swivel angle:

$$\begin{cases} \sin(\theta) = \dfrac{\|\hat{a} \times \hat{b}\|}{\|\hat{a}\|\|\hat{b}\|} \\ \cos(\theta) = \dfrac{\hat{a} \cdot \hat{b}}{\|\hat{a}\|\|\hat{b}\|} \end{cases}$$

$$\theta = \operatorname{atan}(\tan \theta) = \operatorname{atan}\left(\frac{\sin \theta}{\cos \theta}\right) = \operatorname{atan2}(\|\hat{a} \times \hat{b}\|, \hat{a} \cdot \hat{b})$$

Although all of these equations have been discussed in terms of the more obvious connections to Effort dimensions [11], it is also the case that there are correlations between these mathematical equations describing lower-level movement characteristics and the LMA Effort dimensions that the learning algorithms can discern beyond our more intuitive understanding. As mentioned previously, this potentially complex system of relationships explains why I chose to use the Wekinator for the analysis of mathematical data in order to classify movements. Where human error and oversight could skew the results or miss connections between these equations and the low-level motions from which they are derived, the neural networks and classifiers of the machine learning suite are able to automatically and empirically relate this data to the higher-level LMA dimensions. This, in turn, allows us to observe the patterns and subtleties that arise in the movement classification without having to worry about performing analysis by hand. Such an approach also leads to more accurate classification models.

Ultimately, then, since the system performs a continuous process of motion capture, feature extraction, and classification through the Wekinator's algorithms, we have built a comprehensive software package that analyzes the expressiveness of a dancer's movements while they are being performed; the Effort dimensions of his actions are classified in a cheap, non-restrictive way that is done during the process of movement, rather than having to be done after the fact in a lab. To reiterate, this use of motion feature equations and the automation of the machine learning process are important in building a software system that can dynamically analyze the qualities of movement performed in real time.

6. System Analysis

Before explaining the details of machine learning tools and the input dataset used for training, I will first summarize my methods for analyzing the system and the corresponding results found. Once I had trained one neural network for each of the four Effort dimensions in the Wekinator using my input dataset, I tested the performance of the system by executing various different movements in front of the Kinect and judging how well the classifications displayed in the Max/MSP patch actually matched my intended qualities of movement. In a first run, the results were suboptimal, as it seemed that the system was only accurate when I was performing gestures similar to the ones used in training. This therefore led me to exclude joint position from the collection of input features; this decision is justifiable because the Effort classifications provided should be independent of location in space. After this change to the dataset and the subsequent retraining of the algorithms, the system had much better performance. Specifically, the Effort classifications provided more accurately matched the desired values, and the system performed with minimal latency. These results and the analysis process will now be discussed in more detail, followed by a summary and discussion of my findings.

Using the Wekinator machine learning suite, training the system to classify movements based on the LMA Effort dimensions became a relatively simple task. As described previously in the architectural diagram for the system (Figure 2), the raw data from the Kinect was analyzed in the C# code according to the equations presented above; then, these input features were sent in OSC messages (one per frame) to the Wekinator. Each message contained 222 input features, corresponding to the velocity, acceleration, and jerk of each of the three coordinates (X, Y, and Z) of each of the 20 joints (3 * 20 * 3 = 180), the curvature and velocity of each of the 20 joints (2 * 20 = 40), and the swivel angles of both elbows (180 + 40 + 2 = 222).

Then, once the machine learning tool was configured to accept these messages as its input features, I provided examples to the Wekinator by dancing with certain Effort qualities. Specifically, I set the tool to track one Effort dimension at a time and set that feature’s value to either 1 or -1. I then performed a short sequence of movements that embodied the given qualities (choosing distinct pieces of music for each of the 8 rounds helped to focus my motions appropriately) and recorded those examples into the dataset. For the four characteristics, I chose the value of -1 to correspond to Light, Indirect, Sustained, and Bound and the value of 1 to represent Heavy, Direct, Sudden, and Free for the dimensions of Weight, Space, Time, and Flow, respectively. 3747 examples were recorded over the session, with roughly 400-500 examples per classification corresponding to about 15 seconds of consecutive dance time. Each of these recorded sessions included a relatively diverse variety of movements within the given realm of motion constrained by the specifically set classification value. This then helped the system to be able to learn from a varied sampling of gestures with a specific Effort classification.

Once all of these examples were recorded and saved into a dataset, the Wekinator’s training algorithm automatically generated and refined one neural network for each of the four dimensions. In the learning process, only the examples that directly pertained to a given Effort dimension were included; that is, the examples corresponding to when I danced with Heavy Weight were only used in training the Weight neural network and were ignored in training the other three networks. Training was taking a prohibitively long time, however, which made sense given that the dataset contained 3747 examples. Since this learning process was therefore too slow, I followed the advice presented on the Wekinator's homepage[6] and trained each network for 175 epochs with the learning rate decreased to 0.03 from 0.3 (this enabled the network to be trained quickly but with a decreasing error rate per epoch such that the algorithm did not settle into a local, rather than global, extreme value in the training process[7]). This gave the following RMS error values for the accuracy of the model on the training set:

Dimension	**RMS Error on Training Set (Including Position)**
Weight	0.05
Space	0.1
Time	0.11
Flow	0.18

Table 1 Error rates for the four trained neural networks for each of the Effort dimensions

Once the neural networks were trained on the example data, their outputs were sent through an OSC message to a Max/MSP patch, shown in Figure 3, designed to separate the Effort values and display them on a range from -1 to 1. It is important to note that the actual outputs of the neural networks were not constrained to the range [-1, 1], and so scaling was performed within the patch to fit the information into the [-1, 1] interval represented by the Max/MSP slider object. In addition to these real-valued sliders, the output was also visualized with a text message classifying each dimension into one of its two possible values, as determined by a threshold value of 0 such that values below 0 were rounded to -1 and those above 0 were rounded to 1.

[6] http://wiki.cs.princeton.edu/index.php/ChucK/Wekinator/Instructions
[7] Help from http://www.willamette.edu/~gorr/classes/cs449/momrate.html with understanding the learning rate for neural networks.

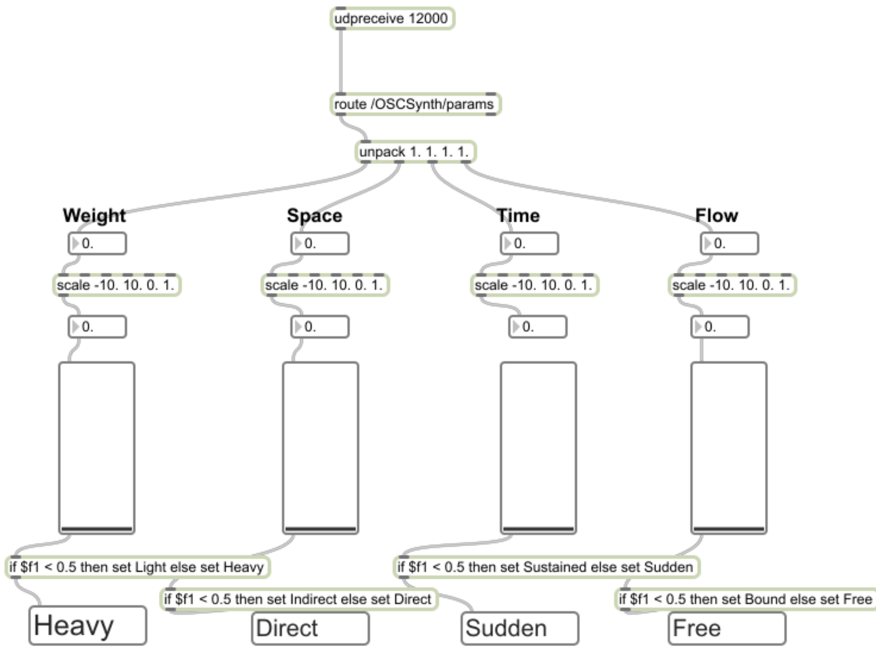

Figure 4 This Max/MSP patch is used to visualize and classify the Effort dimensions as sent by a real-valued neural network learner from the Wekinator.

When I then performed a new series of movements in front of the system to test its performance, I found that the output classifications matched the Effort dimensions that I was trying to convey fairly well in certain situations. I did, however, notice some discrepancies and difficulties in attempting to get the system to recognize the qualities of my motions in the general case, especially when the new movements I was performing were not spatially similar to the training movements; specifically, the system did not make the distinctions between different classifications as clear as they should have been.

Because of this, I decided to run the same dataset through the Wekinator but with discrete output features taking on either one of the two values with no real-valued output in between the two extremes. This transformed the problem into one of discrete classification using an AdaBoost learning algorithm with decision trees as the weak learner. When I analyzed the accuracy of this scheme in the Wekinator, I received the following results:

Dimension	Accuracy of Model on Training Set	Cross-Validation Accuracy (2 Folds)	Cross-Validation Accuracy (5 Folds)
Weight	99.72%	100%	99.72%
Space	100%	99.49%	100%
Time	100%	99.78%	99.89%
Flow	100%	99.32%	99.66%

Table 2 Accuracy of AdaBoost classifier on dataset including position

These values show significant accuracy, and so it would seem that the Wekinator provides better results for my system when run with a discrete classifier rather than a real-valued neural network. However, this result is not entirely true, as the training dataset did not include examples for which the classification value for an Effort dimension was 0 (corresponding to a neutral classification). The inclusion of these values would likely help to stabilize some of the results from the neural networks, but would not be useful in training the discrete classifiers. Additionally, the cross-validation accuracies presented in Table 2 are misleading since even with these very high accuracy values, the system did not perform that well in properly classifying novel gestures. This then suggests that the learning problem presented to the Wekinator's algorithms was different and easier than the actual problem I was intending it to solve. To clarify, this means that although the Wekinator's algorithms had high cross-validation accuracies, this was with respect to a problem that did not accurately represent the greater

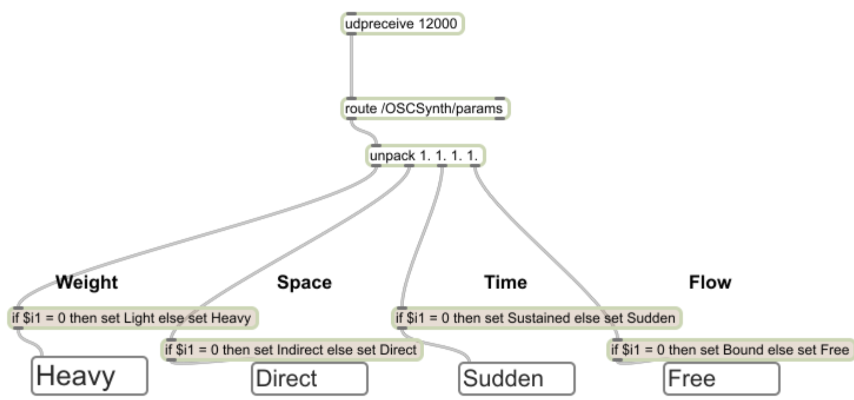

Figure 5 This Max patch is used in the discrete case, for which the Wekinator generates four AdaBoost-enhanced decision tree classifiers to assign a single label to the Effort dimension of a movement.

problem of classifying the expressiveness of dance movements.[8] The system error was especially noticeable when movements were in different spatial locations than those included in the training dataset; the system was therefore more prone to making errors in assigning Effort dimensions to movements that did not correspond (at least loosely) to the training gestures.

Given the wide spread of potential outputs for the neural networks, the difficulty in properly scaling these results to a meaningful range, and the lack of neutral-valued examples in the dataset, I found that the Wekinator provided better results for my system when run with a discrete classifier rather than a real-valued neural network. This in turn supported the simplifying assumption that while performed movements exist on a continuous range for the Effort characteristics, representing these features is more accurate when the Laban analysis assigns either one label or the other to a given Effort dimension. For example, while a gesture executed by a dancer may lie somewhere between Heavy and Light Weight, it is significantly easier to assign a binary classification, rather than a numeric value of, say, 0.765, to that Weight dimension of motion. Additionally, this simplification leads to results of higher accuracy since no human error is introduced through a manual choosing of threshold value for the outputs of the neural networks. The new, simpler Max/MSP patch for the discrete classifier, shown in Figure 4, displays the appropriate valuation for each Effort dimension.

Therefore, in order to maximize the accuracy and descriptiveness of the system, I decided to choose the discrete AdaBoost classifiers (using decision trees as the weak learner) as the preferred learning algorithm, but with a dataset of the same examples excluding position data. This setup increased the performance of the system, and my subsequent tests of the algorithm with new movements showed a significant increase in the responsiveness and accuracy of the

---

[8] The reasoning for this apparent discrepancy between extraordinarily high cross-validation accuracies and the actual performance of the system as measured by my experience was derived from conversations with Professor Fiebrink.

resulting classifications. I believe that this improvement is caused by the fact that the learner was no longer erroneously associating the spatial location of my body with the expressiveness of the motions themselves, which should be independent of their position in space. The resulting accuracies are shown below in Table 3.

Dimension	Accuracy of Trained Model on Training Set	Cross-Validation Accuracy (2 Folds)	Cross-Validation Accuracy (5 Folds)
Weight	100%	75.81%	77.03%
Space	100%	67.79%	76.73%
Time	100%	77.4%	79.37%
Flow	100%	69.51%	71.62%

Table 3 Error rates for the four trained AdaBoost classifiers (excluding position) for each of the Effort dimensions

Although the cross-validation accuracy values are not as high as those presented before in Table 2, the performance of the system when tested with new gestures significantly improved. Specifically, the classifiers no longer make errors based on an erroneous assignment of significance to the position of the dancer's skeleton in space; as such, the system has an increased accuracy and efficiency across the vast array of movements that the human body can execute. Since the actual accuracy of the system has therefore improved in practice, we can see that the exclusion of position data has changed the machine learning problem to be more difficult but more applicable to the real problem of LMA Effort classification.

Now that we have established the system to have improved performance in classifying new gestures, we consider the applicability and success of our system, especially with respect to the intended possibilities of using this package in dynamic dance performances. Clearly, from the accuracy values presented in Table 3, the machine learning algorithms produced by the Wekinator have very high performance relative to the presented dataset. However, there is still error in the performance of the system when used on new movements. This is understandable given the inherent complexity of classifying the expressiveness of gestures in dance performance, but the accuracy of the system is certainly adequate for applications in the realm of artistic works. To support this claim, we consider the fact that any applications of such a system to control the technical elements of a dance piece would most likely result in changes to factors like the color of lights or the instruments in a soundscape over a span of time rather than in an instantaneous fashion. This system would therefore function well, as even though there is some noise and inaccuracy in the classifications in the short term, the algorithms do a good job of assigning values to the Effort dimensions with respect to a larger time interval. As such, despite the errors that are present in the usage of the system and that are visible in the Max/MSP patch used for display in this study, the system is more than adequate for use in dance performances. Ultimately, then, since the goal of this project was to create a system capable of dynamically analyzing the expressiveness of movements for applications to multimedia performances, we have succeeded in this attempt.

7. Related Work

Now that we have discussed the details of our system, we can turn to the background research and related work that served as the foundation for this project. To start, we consider some of the systems proposed in other papers. For example, Jessop's work established a system using body sensors that could track dancers' movements in real-time [6]. Her transformation of this information into Laban Effort quality mappings that could then be used to control various technical aspects of performance formed the central inspiration for my own work. The system presented here, however, has the advantages of being non-intrusive and considerably less expensive. Additionally, her computations involved her own interpretation of which features of movement were significant, followed by a projection of various values she deemed important onto the range of [-1, 1] for the Effort dimensions [6], whereas the Wekinator's automated training in my system helps to reduce any errors associated with the subjectivity and potential fallibility of the manual design of Effort classifiers. While her technique does produce good results, my system improves upon her methods because it considers an aggregate of several different features passed into the machine learning suite such that the complex interrelationships of movement qualities can be noticed by the computer in ways that humans cannot accurately predict or detect.

As discussed in the Equations section, Zhao's paper had a significant influence on the mechanics of my system, and the equations presented in his research are those that I chose to use as feature inputs for my own system [11]. His work is a detailed account of feature generation and extraction based on LMA and the Effort dimensions, and while some of his research is beyond the scope of this paper, reading through his material helped to establish a solid understanding of the details of motion capture analysis systems. Similarly to Jessop's work [6], Zhao's system required a neural network to be built specifically for his research; while this approach does work with valid results, it also introduces error and complexity into the process that can be simplified and improved by using an automated learning suite like the Wekinator.

Ultimately, the motivation for this project was to create a simple, easy-to-use software system that could analyze the expressiveness of movement in real time with the intention of using this information in various artistic and educational processes. I was especially inspired by Tod Machover's multimedia opera, "Death and the Powers" [7] [8], which seamlessly integrated technology into a theatrical performance in such a way as to make the computer systems not only unobtrusive partners to the human actors but also fundamental and compelling components of the production itself. Beyond the manipulations of light and sound based on Jessop's work with the Disembodied Performance system [6], the production also established a narrative in which robotic performers and other computer-controlled elements were integral to the story and the interactions of the actors. While the scale of this opera involves much more than just the movement analysis covered in this project, the work presented by the MIT Media Lab in the production demonstrates that computer science and art truly can be interrelated in ways that enhance the creative opportunities of the artists and highlight the advancements in technological systems.

8. Conclusions and Future Work

Based on the captivating multimedia experiences and academic presentations that I found during the research process, my system was therefore designed to be easily incorporated into various artistic contexts as a multi-purpose tool for controlling audiovisual aspects of a larger piece based on the real-time movements of the performers themselves. I discovered that while emotion and some qualities of movement are subjective, Laban Movement Analysis and the Effort dimensions enable a more objective and quantifiable assessment of the expressiveness presented in dance performance.

Since the generation of movement material of a given expression is subjective and personal process, it is difficult to accurately derive an explanation of the emotional content of a dance performance directly from the movement data itself. However, it is precisely this subjectivity and individualism in performance that makes dance such an interesting and compelling art form, and so I find it very significant that this software system is able to analyze the expressiveness of dancers without hindering their artistic capabilities. Additionally, this system has proven to be an accurate and efficient analyzer of the qualities of motion in real-time, which allows for dynamic applications of the technology to multimedia presentations with an inexpensive and easy-to-use system.

The system created here does, however, have its shortcomings and areas for improvement. For example, as discussed in the System Analysis section, the Wekinator classifiers still have some inaccuracies and noise when new movements are performed in front of the Kinect. This signifies that the system tends to fail or have less accuracy when presented with gestures differ too significantly from those that were in the training dataset. Since it is impossible to encode every possible movement into the dataset, this limitation of the system is understandable, and could be improved if there were a much larger dataset available for training. Additionally, even with the difficulties of the system, its performance is more than accurate enough for use in actual dance pieces. That is, the classifiers do a sufficiently good job that the resulting output could easily be translated into inputs for controlling technical elements of multimedia performances.

As for future work for improving and refining the system and its accuracy, I would first focus on developing new movement feature equations that could better model expressiveness and the Effort dimensions. While the mathematical models that I used here were descriptive, it is clear that the relationships between the dancer's movements and the qualities of his gestures are more complex than can be simply modeled by velocity, acceleration, and the other input features used in this project. In addition to finding new and more descriptive equations, I would also spend time experimenting with which input values were sent to the Wekinator in order to improve the accuracy of the machine learning algorithms. Specifically, it would be interesting to consider building classifiers when the dataset is restricted to, say, data about a single joint (like the right wrist, for example), or only the minimum and maximum values of each mathematical feature equation over all joints in the skeleton. I believe that, with more time and data, the accuracy of this system could be refined even further for better performance as a standalone Effort classifier.

Given the success of this system in terms of being a useful source LMA Effort classifications, which can be used for dynamically manipulating the technical elements of dance performances based on the simultaneous movements of the dancers on stage, it is also feasible to consider extending this system into other LMA qualities, such as Body or Shape, to track some of the different descriptors of movement that are not as directly related to expressiveness as Effort. Another potential extension of the system would involve training the classifiers with movement information from multiple dancers instead of just relying on my own experiences and gestural tendencies. This would enable a broader base of motions for training, which could then improve the overall accuracy of the system. Without this expanded base, though, using the system on other dancers is still possible; it is simple to set up the Kinect along with my software package and observe the Effort classifications that the system outputs. The accuracy of these classifications could then be analyzed through subjective experiences as well as through cross-validation accuracy values, and such considerations could lead to more accurate system overall. This would be significant in applying the system to actual dance performances, as most pieces will involve multiple dancers, each with their own unique styles and trends of movement.

Ultimately, then, this research with LMA, the Wekinator, and the XBOX Kinect has proven fruitful in providing advancements for the integration of technology and the arts. From this development, then, future work would involve integrating this system into actual performances, as by sending the resulting Effort dimensions in OSC messages to lighting boards, sound generators, or other hardware systems (a humanoid dance-bot, perhaps?) to create and manipulate rich environments for artistic projects, controlled automatically in real-time.

Clearly, then, this system has simplified and improved the existing processes of motion capture and Laban Movement Analysis to make these useful and interesting tools more readily available to the average artist. It is my sincere hope that such technologies will continue to grow more prevalent in the dance community, as the possibilities for engaging and remarkable performances continue to expand as we discover new ways to integrate the creative and the scientific.

I pledge my honor that this Independent Work paper represents
my own work in accordance with University regulations.

-Adam Stasiw

References

[1] R. Fiebrink, D. Trueman and P. R. Cook, "A Meta-Instrument for Interactive, On-the-fly Machine Learning," in *New Interfaces for Musical Expression*, Pittsburgh, 2009.

[2] R. Konie, "LMA Effort Bank," 2011. [Online]. Available: www.lmaeffortbank.com.

[3] M. Wright and A. Freed, "Open Sound Control: A New Protocol for Communicating with Sound Synthesizers," in *International Computer Music Conference*, Thessaloniki, 1997.

[4] A. Abazorius, "Dancing Machines," 14 November 2011. [Online]. Available: http://www.csail.mit.edu/node/1640.

[5] "Mortal Engine," Chunky Move, [Online]. Available: http://chunkymove.com.au/Our-Works/Current-Productions/Mortal-Engine.aspx.

[6] E. N. Jessop, "A Gestural Media Framework: Tools for Expressive Gesture Recognition and Mapping in Rehearsal and Performance," Cambridge, 2010.

[7] E. Jessop, P. A. Torpey and B. Bloomberg, "Music and Technology in Death and the Powers," in *NIME*, Oslo, 2011.

[8] T. Machover. [Online]. Available: http://opera.media.mit.edu/projects/deathandthepowers/.

[9] H. Park, J.-I. Park, U.-M. Kim and W. Woo, "Emotion Recognition from Dance Image Sequences using Contour Approximation," *Stuctural, Syntactic, and Statistical Pattern Recognition,* pp. 547-555, August 2004.

[10] R. Montagne, "Merce Cunningham: Dance at the Edge," NPR: The Long View, 29 December 2006. [Online]. Available: http://www.npr.org/templates/story/story.php?storyId=6692356.

[11] L. Zhao, "Synthesis and Acquisition of Laban Movement Analysis Qualitative Parameters for Communicative Gestures," Philadelphia, 2001.

[12] D. Fernandez, "Skeletal Tracking Fundamentals," February 2012. [Online]. Available: http://channel9.msdn.com/Series/KinectQuickstart/Skeletal-Tracking-Fundamentals.

[13] L. Zhao and N. I. Badler, "Acquiring and Validating Motion Qualities from Live Limb Gestures," *Graphical Models,* vol. 67, no. 1, pp. 1-16, January 2005.

[14] J. Rogers-Smith, "Max/MSP to Kinect via OSC for Official Microsoft SDK," 25 August 2011. [Online]. Available: http://jordanrs.blogspot.com/2011/08/maxmsp-to-kinect-via-osc-for-offical.html.

[15] N. I. Badler and D. Tolani, "Real-Time Inverse Kinematics of the Human Arm," Philadelphia, 1996.

[16] J. Rett, J. Dias and J.-M. Ahuactzin, "Bayesian Reasoning for Laban Movement Analysis used in Human Machine Interaction," *International Journal of Reasoning-based Intelligent Systems,* vol. 2, no. 1, pp. 13-35, 2010.

[17] A. Kapur, A. Kapur, N. Virji-Babul, G. Tzanetakis and P. F. Driessen, "Gesture-Based Affective Computing on Motion Capture Data," in *International Conference on Affective Computing and Intelligent Interaction*, Beijing, 2005.

[18] D. Bouchard and N. Badler, "Semantic Segmentation of Motion Capture using Laban Movement Analysis," *Intelligent Virtual Agents,* pp. 37-44, 2007.