Learning the Primary Colors of Dance

Amy LaViers '09

Final Report

30 April, 2009

Prof. Naomi Leonard

Prof. Luigi Martinelli

MAE 442

a senior thesis submitted to the Department of Mechanical and Aerospace Engineering, Princeton University in partial fulfillment of the requirements for the degree of Bachelor of Science and Engineering.

Abstract

This thesis lays exploratory groundwork in order to better engineer a system that can give structure to dancing. Dance is currently a relatively unstructured art form. At least in part, this is due to the complex medium through which it speaks - the human body. The body is a dynamic system with many degrees of freedom and thus, whose motion has a complex external mathematical parameterization. The hypothesis of this thesis is that there is a simpler, internal parameterization of this motion. That is, that our behavioral patterns reduce the space of possible human postures, simplifying the parameterization problem. With this guiding principle, such a parameterization is searched for in this thesis. Motion capture is employed to collect a detailed representation of human movement. A linear and nonlinear algorithm are applied to this data in hopes of finding a reduced representation. The results of this analysis recreates results from previous studies and raises questions for future works.

Acknowledgements

I would like to acknowledge the critical support and contribution to this project of: Prof. Naomi Leonard for advising this unconventional project, finding invaluable collaborators to the project, and being a great cheerleader; The School of Engineering and Applied Science (SEAS) and the Mechanical and Aerospace Engineering (MAE) department for funding the project through the SEAS and Marshall fund; Prof. Liz Bradley and Jonathan Olson at the University of Colorado for their critical collaboration and contribution to the ideas and content of the project; Prof. William Bialek and Greg Stephens for their provision of a prototypical project and guidance on the subjects of data analysis; Prof. Magnus Egerstedt for the use of his motion capture equipment and resources; Peter Kingston for two long days of his time and effort collecting this dataset with me; Vicon Motion Systems for their generous gift of trial versions of several of their motion capture software and software support from manager Jeffrey Ovadya and engineers Kristin Roepke and Steven Andrews without whom the necessary data manipulation and format conversions for the analysis would not have been possible; Neil D. Lawrence who was so kind to not only answer my emails from across the pond but also to update his software versions in attempts to provide me with greater data visualization capabilities for his GPLVM software; to Curtis Huttenhower for answering several frantic emails with calm, clear computer science know-how that saved me on numerous occasions; and to Prof. Fernando Perez-Cruz for taking the time not only to explain much of the math associated with the nonlinear algorithms to be discussed but also

for proofreading and providing invaluable feedback on Ch.3.

I would also like to extend special thanks for their wonderful help to: Jo Ann Kropilak-Love, Prof. Alexander Smits, and Emanuel Stockman for departmental support; Prof. Rebecca Lazier and The Lewis Center for the Arts for guiding and advising me on the subject matter of dance for the last four year; Chris Park for his vital Python knowledge; Jack Wang who in addition to having written one of the key papers this thesis cites, answered several of my specific questions via email; and to my family and friends who proofread parts of this thesis: Joe Rokicki '08, Kate Carroll '09, and Matt Tilghman '09. Also, thanks to my roommates Whitney Davis '09 and Natacha Jamar '09 for thanking me in their acknowledgements.

Some data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.

Contents

Chapter 1

How Do Dancers Represent Movement?

The notes that musicians use to compose scores are clear for a given instrument. Each instrument has its own range of notes, pitch, and timbre. The relationship between each note is understood in a clear mathematical sense, i.e. each note is a given number of steps and/or octaves apart. As a result, computers can understand and reproduce this relationship. That is, given a musical score, a computer can instruct an output device to play the corresponding tonal and rhythmic structures, and likewise, if played some music through an appropriate input device, software can construct the corresponding standard music notation.

In dance, the instrument is the human body, a much more complicated and less understood object. Its limits are continually pushed by dancers and choreographers resulting in ever changing styles and vocabularies (which are composed of a new, different set of "notes" or movements). Although choreographic theories and notations have been developed for danced movements (that is, movements which are motivated by rhythmic, spatial, kinesthetic, and expressive principles), transcribing a work onto paper or into animation

software is a laborious task that lacks an automated solution. This is just one example of how dance is deplete of quantification, even relative to other genres of art.

1.1 The Beginning of Dance Theory

Perhaps the most famous attempt to quantify dance comes from Rudolf Laban, an early 20th century dancer, choreographer and self proclaimed dance theoretician. Over the course of his life, he developed quantitative theories about the space surrounding a dancer, the effort and dynamic qualities with which a dancer moves, and the positions and movements which create "harmonious" choreography.

Figure 1.1 provides visual examples of the three dimensional geometries Laban used to specify the orientations of the body that comprise "harmonious movement sequences." These diagrams illustrate the shapes created by directions in which a dancer might reach and the planes that bisect the body in which motion takes place. Employing a comparison to music, he also created movement "scales" (Fig. 1.2) which, like their musical counterparts, he designed to teach dancers about the rela-

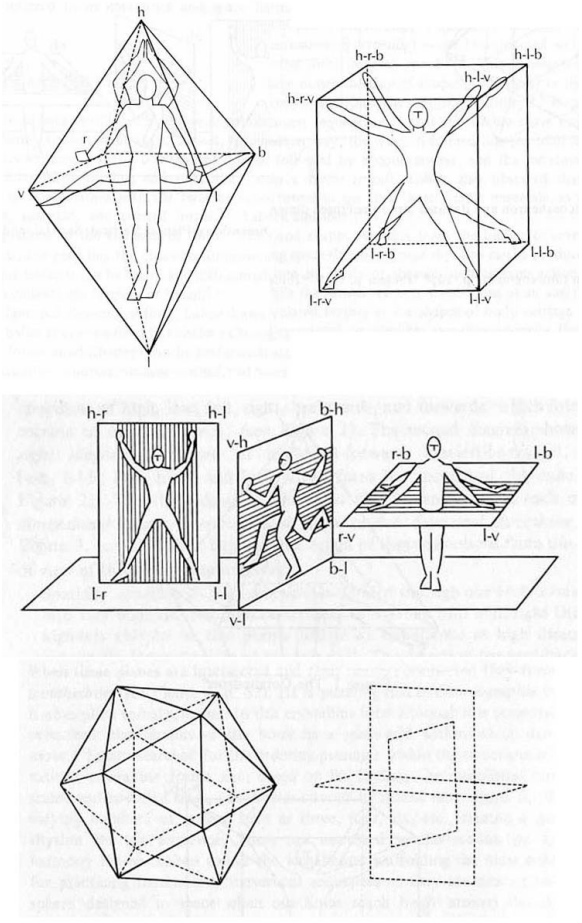


Figure 1.1: **Laban's division of space.** This figure illustrates Laban's concept of the body's geometry and part of his theory on how to best utilize the space surrounding a dancer's body. [19]

tionship of their body with space, heightening their awareness of the many permutations of this relationship. [19]

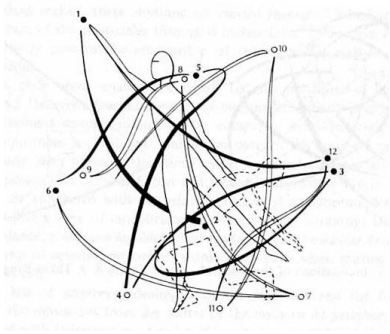


Figure 1.2: **Laban movement scales.** These exercises indicate "harmonious" movement trajectories. [19]

Laban's influence on the dance world has developed the language we use to describe dance. Ohio State University's Professor Emeritus Vera Maletic writes in [19] that the unifying element in Laban's work is his "confrontation with the absence of a contemporary dance terminology." As Maletic goes on to explain, Laban used his theories of movement to form principles of choreography which would dictate "symmetry, balance, and other forms of harmony." Thus, Laban used his studies to form guiding principles for himself and other choreographers.

In addition to his advances in description and theory, Laban created a codified system of dance notation based on these studies. This notation, shown in Fig. 1.3 requires many parameters for each pose in a choreographic sequence. Compare Laban's notation to the musical staff at the left of the figure: no where in his notation does something so ordered or sequential occur. While based in his extensive behavioral studies, Laban's script lacks the order and logic that we experience when we move. His notation uses the vertical columns on the page to segment the body; the various shadings are used to indicate how far off the ground a movement should happen; and decomposed rectangels are used to indicate the direction of travel for each step. The example of this notation demonstrates how complex

this paramterization can be made, despite the simplicity with which we take a step or reach an arm.

1.2 Computation and Dance

1.2.1 Merce Cunningham

Following Laban, choreographer Merce Cunningham's interest in similar tools resulted in the development of DanceForms[1] , software for recording dance movements. Having computerized software for choreography aided Cunningham in two obvious ways. One, he has made technology a part of his creative process since the 1960s [6]. In some ways, technology fits seamlessly into his philosophy which involves systematically removing typical elements such as music from his choreographic process and using chance and randomness into his work, resulting in cerebral dances that have been compared to thermodynamics and information theory [6]. Two, in the early 1990s when he began developing it, Cunningham was well into his seventies, and the tool would provide him with a way to continue choreographing as his body aged. However like Laban's scripted notation, the software has a high learning curve and is considered too tedious and labor intensive to use on a daily basis. The most prevalent form of dance preservation is film; these convenient archives can be created and interpreted without extensive training. Choreographers supplement the imperfections of this two dimensional record by taking videos from many angles.

1.2.2 William Forsythe

Other notable efforts include those of present-day choreographer William Forsythe who describes movement and choreography in terms of three dimensional curves and translating shapes in his educational software package, *Improvisational Technologies: A Tool for the Analytical Dance Eye*, that visualizes these concepts with a mixed media of video and

[1] Originally titled LifeForms [6]

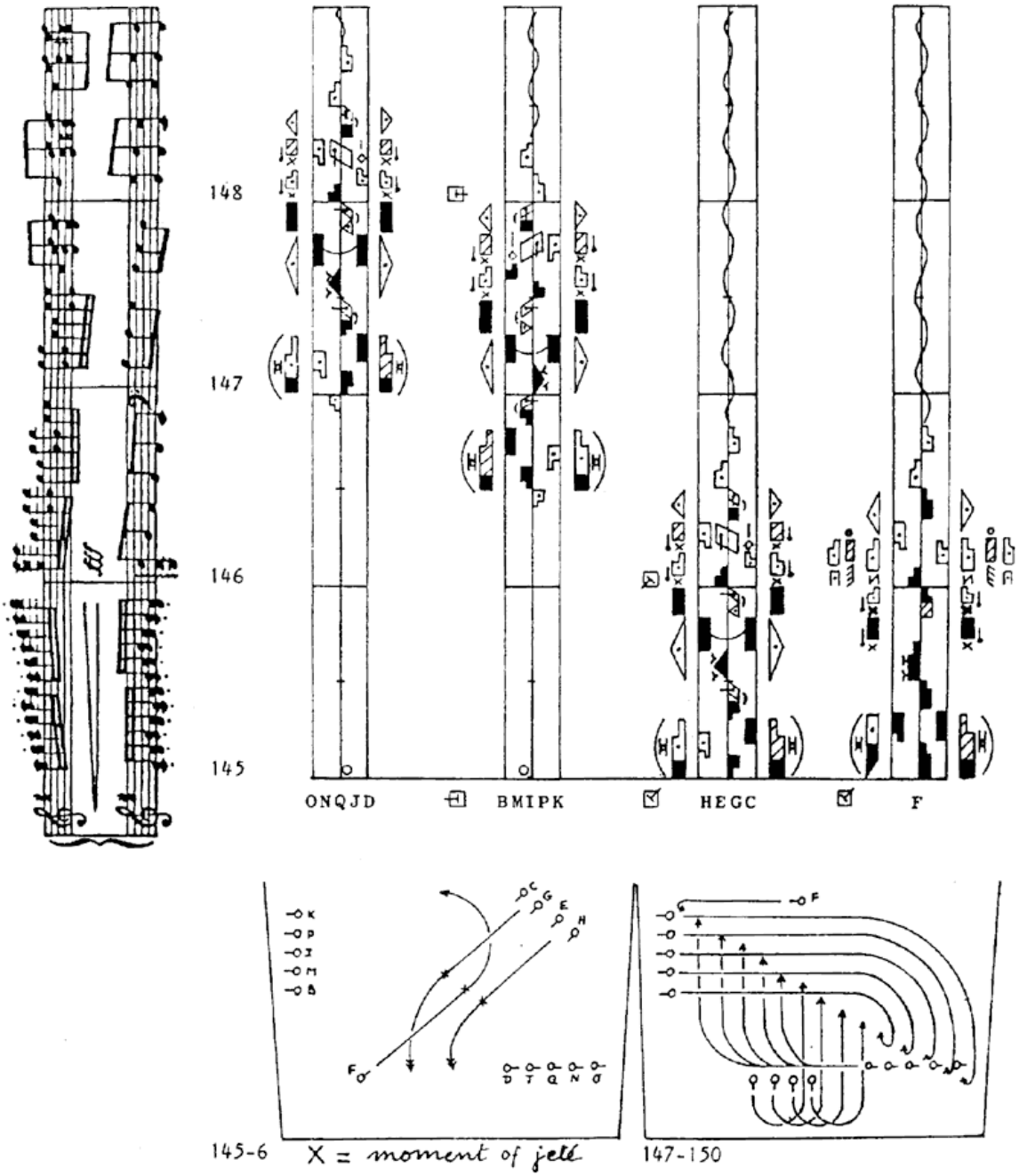


Figure 1.3: **Example of Laban notation.** This figure from Rudolf Laban's Schrifttanz (1928) illustrates the many parameters required to specify human movement in this famous notation. [11]

animations [8]. More recently Forsythe has collaborated with Maria Palazzi and Norah Zuniga Shaw at Ohio State University's Advanced Computing Center for the Arts and Design (ACCAD) in order to "[visualize] choreographic information in new ways" [9]. This project, titled *Synchronous Objects*, involved intense documentation in order to create detailed computer visualizations (and revisualizations) of William Forsythe's work "One Flat Thing, Reproduced." The spatial and temporal content of a piece of choreography should not be underestimated. Dance is a phenomenon of synchronous human movement that requires, not only careful planning by the choreographer, but also the intense attention and skill of the dancers. These simulations are more than the materialized shapes of Forsythe's geometrical vision. They also entail the real time cues off which the dancers create this dynamic work of art. While the many images from this project are breathtaking, one of which is shown in Fig. 1.4, it is the rehashing of the work, which the center's rigorous quantification allows for, that enables greater understanding of the piece. Through the project's media, it is like the piece has been put in slow motion so that the choreographer, his dancers, and we, the audience members, can better experience and understand the work.[9]

1.3 The Need for Representation

Just like a painter knows that each color in their palette is a precise combination of only three primary colors or musician working with eight octaves of sixty-four notes, with a more quantified description of dance, choreographers will benefit from the ability to identify similar "primary" movements and principles. Continuing to work towards better, more quantified representations of movement and choreography will add to movement theories and enhance the artistic expression in and understanding of choreographers' work. Additionally, the identification of the underlying patterns in movement and choreography, can also provide a basis for a more complete, more intuitive and unified system of dance and movement notation.

Figure 1.4: **A Computer Visualization of Choreography.** In this image, "Difference Forms," patterns in Forsythe's choreography (as seen from above) are illuminated using video processing. Photo courtesy of Synchronous Objects Project, The Ohio State University, and The Forsythe Company.

Chapter 2

How Do Engineers Represent Movement?

The ultimate goal of this project is the same as that of Laban, Cunningham, and Forsythe's endeavors: to provide artists a new, innovative tool. Like these dancers, this project will attempt to analyze and quantify the patterns in human movement, dance, and choreography, but here, the approach originates from a mathematical point of view. Thus, the first question is one of representation. Dancers have approached this question from an internal perspective using their own experiences with the body's physical mechanisms. Here, this perspective is an external one, and the experiences of dancer must be represented and collected for the computational mechanisms of a CPU.

To answer the question of representation, there are two major options: represent dance movement as a series of dynamic movements or as a series of static postures. This chapter identifies studies which employ each of these representations and will use them to demonstrate the choice of representation, and through their example, will more explicitly outline the concept of this project.

2.1 Dynamic Representation

Domitilla Del Vecchio and her collaborators [7] look for "primitives of motion" or movemes (first introduced by [4]) within the motion of a mouse pointer created as their test subjects draw specific shapes on a computer. They break up the motions each subject uses to complete the drawing tasks into discrete dynamical systems through "segmentation and classification." Specifically, each moveme is defined by a matrix A and vector b where the model for the moveme is a linear time invariant (LTI) system with a unit step input and full state output given in simplest form by the familiar form in Eqs. 2.1 and 2.2. That is, they make the reasonable assumption that the motions users employ to draw differ in the behaviors of their velocity and acceleration over time. They use the differences shown in Fig. 2.1 to specify separate systems, defined by a set of mathematical equations, for each. It is these systems with which they hope to generate an "alphabet" of movemes which, once found, can spell out any motion.

$$\dot{x} = Ax + b \tag{2.1}$$

$$x = y \tag{2.2}$$

Once they have identified points in their data where the next model or moveme should take over, termed switching times, they classify this data as being one of three different movemes which are specific to the task of drawing the objects outlined by their experiment. Then, their segmentation algorithm The results of this are shown in Fig. 2.2. They complete this task with an average error of only 10.5%.

This approach does succeed in creating a model for pointer motion; however, as the wrist actually moves in three dimensions when it controls the mouse, the motion of the pointer on screen is only an indirect, simplified measure of actual human movement. This contrived experimental set up may not be reasonable to expand to more complicated motions. Addi-

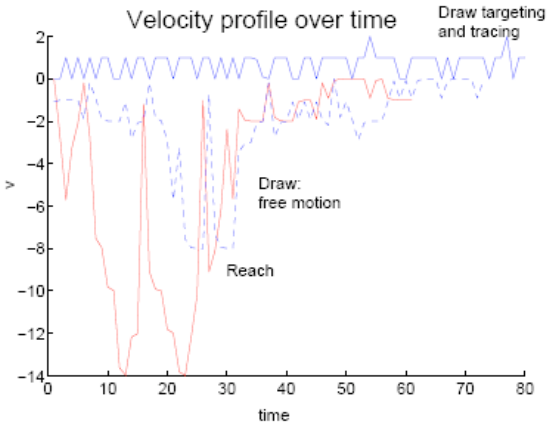


Figure 2.1: Draw and reach velocity profiles [7].

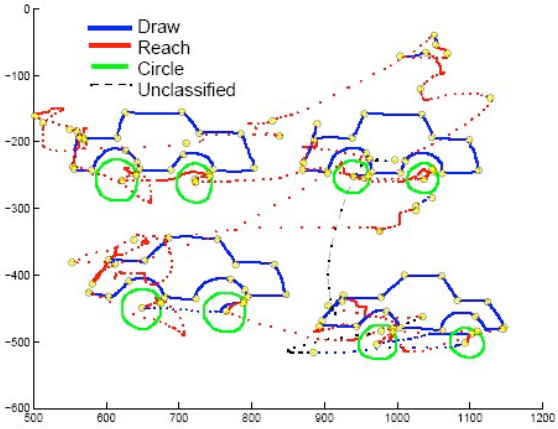


Figure 2.2: Draw, reach, and circle classification [7].

tionally, since each moveme, once identified, must be classified and a model proposed and verified, this classification process may also become less and less practical for more complex and/or 3-D movements.

2.2 Static Representation

Instead, I consider a different approach: using static postures instead of dynamic systems to represent the movement. This approach naturally leads to a different mathematical analysis of the data like the one that has been completed for the movements of the species *Caenorhabditis elegans* (Fig. 2.3) in a lab at Princeton University. Stephens and collaborators [26] discovered that essentially all the postures of these worms could be represented with a linear basis of only four postures. Each pose was represented as one hundred angles and was recorded as the worms were allowed to move freely. These poses create a more complicated data space than the one of simple Cartesian coordinates enforced in Del Vecchio's experiments, it also creates one that is more likely to tell something interesting about the worm's natural behavior.

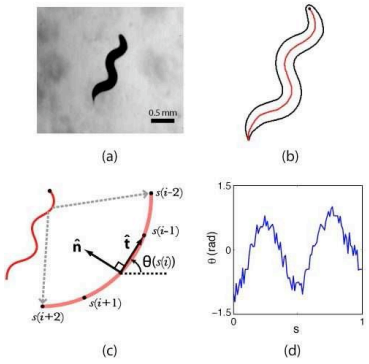


Figure 2.3: ***Caenorhabditis elegan* (C. Elegan).** This figure shows (a) the species itself as tracked by the researchers, (b) the image processing of the worm, (c) the mathematical representation parameterized by 100 joint angles, (d) and the final representation of the worm image as data to be analyzed [26]. Pay attention to this graph. It contains a worm represented in four ways that outline, in general, a parameterization of a dynamic system and foreshadow the data collection for this thesis.

Stephens created a large data set of worm postures (on the order of 10^5 postures) using 2-D images to construct a description of the worm's centerline (assuming a rigid body around that centerline). Initially, the problem was described by 100 degrees of freedom (the 100 angles that composed each pose). The underlying simplification of the postures (and thus motion) of C. elegans was uncovered by performing a dimensionality reduction algorithm on the data set. These techniques will be discussed in more detail in the next chapter. In general these algorithms analyze the variance (deviation away from the average) of the data and find new axes or parameters that can more simply describe the resulting data set. In this case the four postures found can be viewed as analogous to the three primary colors, red, blue, and yellow/green: any linear combination of the basis postures can create almost all (95%) of the observed postures.

Subsequent analysis of three of the four postures, termed eigenworms, gives insight to the worms' behavior. It is first noted that two of the postures represented the means by which the worm travels either forward or backward. This oscillatory relationship can be seen in the ringed pattern formed when the probability density of the two modes is plotted. The third eigenworm is shown to correspond to the mechanism which the worm uses to turn its path. Stephens et al. provide an intuitive graph that shows when the instantaneous value of this third component in the worm's posture breaches some threshold. The plot shows that this threshold is exceded when the worm's trajectory makes a significant curve. Thus, the worm requires this third mode to change direction.

Finally, equations of motion in this new, lower dimensional space, are derived. Although these equations only consider two of the eigenworms, they show that for a thermal stimulus the worm's response which was thought to be random is actually quite predictable. With a more complete and quantitative description of the worm's movement, this response is seen to be correlated with the precise position of the worm at the time of the stimulus. This knowledge allowed the worm to be effectively steered around on its agar medium through specific application of thermal stimuli, showing that the basis that spans the voluntary or

random motion of the worms also applies to the directed case when a stimulus is present.

2.3 How will this thesis represent movement?

The research of Stephens et al. is interesting and useful because of the new behavioral information it provided about a species, which as a model organism for many biological experiments, was already very well studied. Additionally, the elegant concept of the experiment provides a simple model that is archetypal of this project. This model follows the steps outlined below:

1. **Establish a "high dimensional" data set.**

   Our data set will contain frames of human motion. This set, like the one of C. elegan postures, is clearly represented by many parameters. In the case of the C. elegans, the data set was described by 100 parameters. This number of parameters can be thought of as the dimensionality of the data set, and it typically scales with the complexity of the system under consideration. Clearly, for the case of human movement, the number of degrees of freedom is many times greater than for the simple C. elegans, but as a result, our ability to represent this motion is limited. We will consider a data set represented by 57 degrees of freedom (DOFs). These degrees are defined by the 47 points of measurement on the body that define 26 joint angles which each have associated DOFs.

2. **Choose computational methods to search for underlying simplicity.**

   The very same principle component analysis (PCA) used in [26] will be applied to a data set of human postures. As will be discussed in the next chapter, it can be expected that the true underlying structure of this many times more rich case of motion will not be revealed using the simplistic, linear assumptions made in PCA. While

in this chapter the boons of dynamic representation have been downplayed in favor of highlighting the differences in the two experimental setups discussed, the second computational method that we will consider does take advantage of the fact that our movement is constrained by the fact that each posture is succeeded in time by another. This method, which has been specialized for data of human movement, also removes the assumption of a linear relationship between each pose that Stephens et al. employ and instead applies probabilistic and dynamical ones.

3. **Collect data with some convenient form of representation.**

   Measuring the motion of humans is a well studied problem. Perhaps it began on the cave walls of the Neanderthals as they recorded images of themselves going about their daily activities. Now, it is now a thriving industry that quantifies the biomechanics involved in movie making, athletic gear development, and medical rehabilitation. This technology, generally called motion capture, has been developed into a fairly repeatable procedure of data collection and manipulation by companies such like Vicon Motion Systems, who made this data set possible.

4. **Process data with chosen methods.**

   The tedium associated with data analysis will be briefly outlined.

5. **Apply results to a greater, behavioral picture.**

   It is hoped that my results will begin to provide dancers and choreographers with some quantification of their art: of the mechanics of movement and of the creativity of choreography. The results presented here are a baby step in that direction. Their implication will be presented in the larger context of similar research that is currently on-going.

Chapter 3

Finding Patterns: Tools for Analysis

Why hypothesize that the behavioral space of an organism of a lower dimension than the space of all possible postures? Or, in other words, why say that the set of postures an organism naturally makes is smaller than the set of postures that it can physically make? The intuition here is easy to demonstrate: Reach out for an object in front of you. Now scratch your head. Similarities can be observed between the two actions: each involves the movement of the fingers, hand, fore and upper arm through distinct angles at each of the connecting joints.

Mathematically, this parameterization is complex. The actions involve at least thirteen joints (two for each finger, the wrist, elbow, and shoulder) some with multiple degrees of freedom (the elbow has one: it bends in one direction, but the shoulder has three: it can swing in two different perpendicular planes and can rotate). Additionally, classical dynamics will associate a position, velocity, and acceleration for each appendage or section of the arm.

Do the actions again. Did you think about anything like the representations just dis-

cussed? No, because behaviorally, the actions are encoded in the brain as completely separate, holistic entities. We, as humans do not control individual appendages and joint angles. Equivalently, the neural map of actions is not organized (or parameterized, we might say) in terms of the human body's many parts and associated joint angles but is instead organized behaviorally. This principle can be experienced as in the above experiment and has also been evidenced by neuroscientist Michael Graziano [10] in his work with Macaque monkeys. Furthermore, this principle has been applied in the dance world through a set of exercises designed for teaching students to access movement outside their current repertoire by Laban's student Irmgard Bartenieff [1] [1] .

In the technical world of data analysis and machine learning, this neural map can be described as a possible "underlying manifold" or set of "latent variables."[2] These are used to describe structure that, although unseen, governs a given data set. Data, as collected by scientists and engineers in terms of two convenient parameters, will very often seem like complete, nonsensical chaos - especially for poorly understood systems. This is because often, the most convenient parameters for measurement may not actually govern the system and thus, will not reveal its true simplicity. For example, in the case above of human motion, joint angles can easily be measured while neural organization is a subject that confounds and motivates an entire field of research. Being able to recognize this simpler parameterization (or, indeed, any other parameterization - perhaps without a neural correlate) would allow for more concise representation and provides deeper insight into the system under consideration. The technical formalization of this search for simplicity is outlined in this chapter.

The ways in which this representation can be searched for are diverse. A subtle difference between them that should be highlighted is whether the method is meant for reduction or explanation [14]. In fact the term that is often used, "dimensionality reduction," is really

[1] In Laban's own words: "Don't think of back bending but think 'Your head leas into backward arc towards the floor....!'" [1]. Like Laban, Rebecca Lazier, a professor in Princeton's Dance department, emphasizes that students should think of an action to perform not of the body part(s) that they want to move.

[2] It can be said that the latent variables lie on this manifold.

a misnomer for this thesis. Often a reduction in the data's representation is desired; for example, PCA can be used for image compression. However, it is actually study of the structure itself that motivates this thesis. That is, here, the question: "What are the set of rules which govern human movement and make complex motions so simple to execute?" is favored over a related one: "What is the smallest representation of this motion sequence required to create a viable reproduction?" If indeed it is the case that PCA captures most of the information with only a few, primary terms, then this method will serve to explain much of the structure; this was the case for the C. elegans [26]. It is also feasible that once human movement is better understood (that is, better mathematically parameterized), reduction through these techniques will provide the basis for a viable dance notation.

Two algorithms have been chosen to analyze the motion capture data found online at Carnegie Mellon University's database and collected specifically for this thesis at Georgia Institute of Technology. One is PCA (as used in [26]) which can be visualized as the switch between a two dimensional scatter plot and a one dimensional correlation (defined by a line of best fit), and is outlined in Section 3.1. The second builds on the basic idea of PCA, adding flexibility as it searches for nonlinear relationships and incorporating components of probability and dynamics to improve its prediction. While a full derivation is outside the scope of this thesis, Sec. 3.2 presents each layer in turn, along with its historical context, building up to the concept of the final algorithm, Gaussian Process Dynamical Modeling (GPDM) [31, 32].

3.1 Linear Tools

As described above, high school algebra's simple example of a line of best fit, is a good example that intuitively demonstrates many of concepts involved in any dimensionality reduction technique. Principal component analysis (PCA) is the multi-dimensional extension of this and can be described visually with sample data that depends on two variables as in

[25]. The following is a brief summary that outlines PCA. (Also see Appendix A.)

1. Subtract the mean, μ, of the data set, **D**. Figure 3.1(a) shows the modified set (**D'**) with a new mean of zero.

2. Calculate the covariance matrix C of the data. For this two-dimensional example, this matrix is a 2×2 matrix with the covariances of each pair of variables as indexed by the rows and columns of the matrix (Eq. 3.2). Covariance is defined in Eq. 3.1.

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - X)(Y_i - Y)}{n-1} \tag{3.1}$$

$$C = \begin{bmatrix} cov(X,X) & cov(X,Y) \\ cov(Y,X) & cov(Y,Y) \end{bmatrix} \tag{3.2}$$

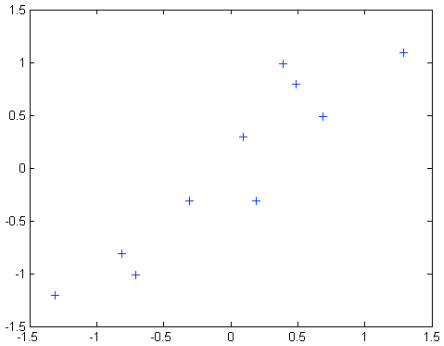
3. Calculate the covariance matrix's eigenvectors and eigenvalues. Eigenvalues indicate the magnitude of some change that a linear mapping (which can be represented as a matrix) exerts on a vector or array of data. Eigenvectors indicate the direction of that change. For example, if the mapping is a rotation, the eigenvalues will be 1 and the corresponding vector will be the axis of rotation. For a 2×2 matrix there will be two vectors with associated values. Here, these two vectors, $\mathbf{e1} = [e1_1\ e1_2]$ and $\mathbf{e2} = [e2_1\ e2_2]$ now form a new, orthogonal coordinate system as shown in Fig. 3.1(b) [25].

4. The first eigenvector (that is, the one with the highest eigenvalues) is what looks like the traditional line of best fit[3] in the plot to the left. The second eigenvector is perpendicular to that and encodes each point's deviation from this principal component. The data can now be expressed in terms of these eigenvectors. This transformed data (Fig. 3.1(c)), **T**, is defined according to $[Transformed\ Data] = [Transposed\ Eigenvectors]x[Transposed\ Mean\ Adjusted\ Data]$. This is made explicit in Eq. 3.3.

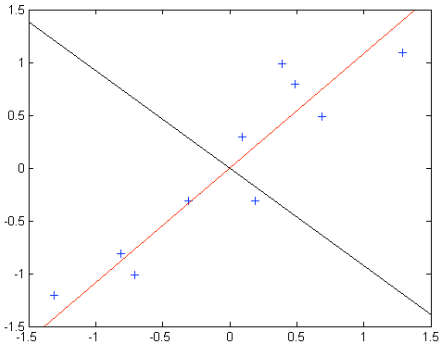
$$\begin{bmatrix} t_{xx} & t_{xy} \\ t_{yx} & t_{yy} \end{bmatrix} = \begin{bmatrix} e1_1 & e1_2 \\ e2_1 & e2_2 \end{bmatrix} \times \begin{bmatrix} d'_{xx} & d'_{yx} \\ d'_{xy} & d'_{yy} \end{bmatrix} \tag{3.3}$$

5. Here there is a choice of how many eigenvectors to keep (for n-dimensional data there will be n eigenvectors) for data reconstruction ($[Transposed\ Original\ Data] =$

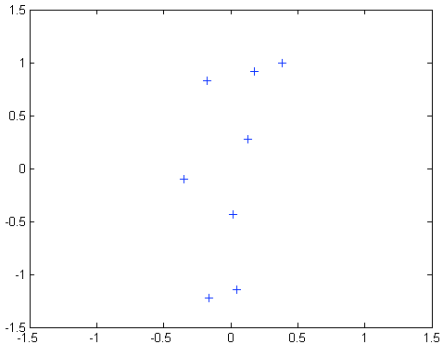
[3] One important difference is the assumption of a dependent and independent variable that a line of best fit makes but is absent in PCA. Here, if we switched which data was called **y** with the data called **x**, this vector would not change. [20]



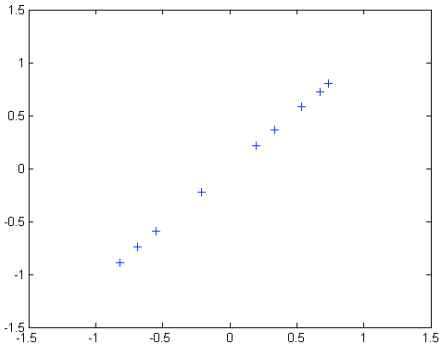
(a) Mean Subtracted Data Set



(b) Principal Axes



(c) Transformed Data Set



(d) Reduced Data Set

Figure 3.1: **PCA Demonstration.** Refer to these for illustration of text. Since these are purely for proof of concept, the data analyzed here is arbitrary and the plots are shown without dimensions labeling the axes. See Appendix A for MATLAB code.

$[Transposed\ Eigenvectors]x[Transposed\ Transformed\ Data] + Mean$ shown in Eq. 3.4). If both eigenvectors are chosen to represent the data, the transformed set is identical to the original set. If only one eigenvector is chosen, the transformed data becomes lower dimensional. Figure 3.1(d) shows that an expression of the data with only the principal component (the eigenvector with the highest eigenvalue) is a pretty good representation for this correlated original data set.

$$\begin{bmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{bmatrix} = \begin{bmatrix} e1_1 & e1_2 \\ 0 & 0 \end{bmatrix} \times \begin{bmatrix} t_{xx} & t_{xy} \\ t_{yx} & t_{yy} \end{bmatrix} + \mu \tag{3.4}$$

This process is used in many applications that find simplified representations of complex data sets. For example, in the process of creating facial recognition software, an image of a face is represented as a single row vector of pixel values (each row of n pixels from the nxn image is concatenated onto the row vector). Then many images (now represented as row vectors) of faces are stacked into a matrix which is decomposed by this process. Now, given enough facial images, the data set can be significantly reduced from its original $n \times n$ dependence. This reduction allows for the implementation of computation in the process of facial recognition. Likewise, a similar application where the data set is, say, the choreography of a given choreographer, could result in an efficient digital record of that artist's work.

3.2 Nonlinear and Probabilistic Tools

Several nonlinear techniques were considered for use in this thesis including: local linear embedding (LLE) [23], Isomap [27], and one that uses rate distortion theory to find its optimal manifold [5]. The general concept for the mechanism of these techniques is illustrated in Fig. 3.2.

Isomap groups data into the"neighborhoods" similar to those referenced in Fig. 3.2 by finding the points which are the very closest to each other (proximity data), while, as the name suggests, LLE finds many, locally linear fits between the data in order to group it. It

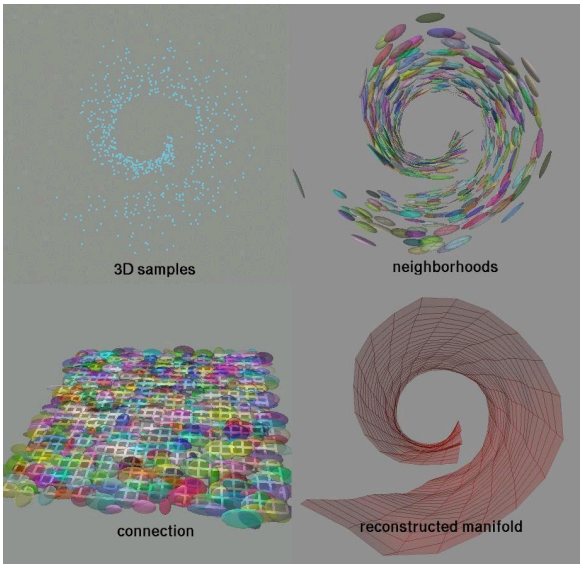


Figure 3.2: **Nonlinear Dimensionality Reduction.** This outline illustrates the general concept of nonlinear dimensionality reduction which use proximity data, that is, information about each data point's nearest neighbor, to construct a nonlinear manifold. [3]

is quite appreciable from the graphic example of the spiraled, 3-D data set that a simple linear fit would by-pass much of the structure and simplicity in this data; however, the small groupings or "neighborhoods" created by these techniques do not. This allows each algorithm to construct connections either by finding shortest distance between each cluster or by weighting each cluster in order to create a manifold which reflects the true structure of the data. On the other hand, Denis Chigirev and William Bialek's information theory approach creates two quantities one which represents the amount of information required to construct the manifold and the other which quantifies the amount of distortion a given manifold will impart on the data. It is through balancing this trade off that this algorithm creates its data-fitting manifold.

The above algorithms provide a sampling of typical nonlinear dimensionality reduction techniques. However, they consider that each point is generated independently. This is not the case for any data with a temporal dependence - like the case of human motion. The chosen algorithm, Gaussian Process Dynamic Modeling (GPDM), does consider this

dependence. Additionally, it has proven success in revealing underlying simplicity for the human motions of walking and swinging a golf club [32].

3.2.1 Latent Variable Models

GPDM derives from information theory and, at heart, is a latent variable model (LVM).[4] In his textbook, *Information Theory, Inference, and Learning Algorithms*, David Mackay compares these models to error-correcting codes - a common application in this field. These codes interpret a signal or data set that has been encoded with extra information in order to preserve its ability to be reconstructed even if some data is lost during data transfer (say, over old telephone lines or a busy internet server station). For example, the most simple method for error detection is the addition of an extra "1" or "0" (a bit) to a string of seven bits (to make a byte of data) that indicates some information about the previous bits. This bit is called a parity bit and may indicate whether the number of bits that are a "1" should be even or odd. The simple error-checking code would then check to see that this number and the parity bit were consistent.

Let's refer to the original data set as some vector $\mathbf{x} = [x_1, x_2, ..., x_n]$, a new data set (to be broadcast via some information transfer medium) as $\mathbf{y} = [y_1, y_2, ..., y_k]$, and the mapping between them as a matrix $\mathbf{G}$. Notice that the number of elements in $\mathbf{x}$ and $\mathbf{y}$ are different and that $\mathbf{G}$ lets us switch between them, that is, $\mathbf{x} \in R^n$, $\mathbf{y} \in R^k$, and $\mathbf{G} : R^n \rightarrow R^k$. We can write this relation in equation form as in Eq. 3.5 [18]. Generally, we expect $n < k$ so that the redundancy inherent in expressing $\mathbf{x}$ (of dimension R^n) in the space of $\mathbf{y}$ (of larger dimension, R^k) provides the error-correcting codes with the extra information necessary to reassemble $\mathbf{x}$ if any information is misinterpreted. Thus, from the perspective of an error-correcting code, $\mathbf{x}$ is a pattern, or variable, that is *latent* or hidden in the data ($\mathbf{y}$). Unlike these codes where the redundancy of $\mathbf{y}$ is used to this advantage, in the case of human

[4]The discussion in this entire section derives from three main sources, [18, 15, 20], and only the more specific quotes are identified hereafter.

motion and dance, the redundancy of $\mathbf{y}$ is viewed to be inhibiting to our understanding.

$$\mathbf{y} = \mathbf{G}\mathbf{x} \tag{3.5}$$

In the case of LVMs, we know $\mathbf{y}$ *only* - these are the k number of observations for the system - the mapping that created them is an unknown. LVMs provide a guess or model as to what the latent variables $\mathbf{x}$ and mapping $\mathbf{G}$ are. Think of this in terms of famous war time encoding algorithms, such as those implemented by the Enigma Machine, where much time and many efforts went in to finding $\mathbf{x}$ and $\mathbf{G}$ from $\mathbf{y}$. Compare $\mathbf{x}$ to communications between war time enemies, $\mathbf{G}$ to an encoding algorithm used to keep these communications secret, and $\mathbf{y}$ to intercepted (but encoded) transmissions. These models extend linear PCA discussed in the previous section to a more flexible method by incorporating a nonlinear mapping between the real world data ($\mathbf{y} \in R^k$) and the space, of dimension R^n, in which $\mathbf{x}$ is more naturally expressed and $\mathbf{G}$ is applied.

Here the analysis makes a big, interesting assumption, and before we make it, we will take a paragraph to clarify. There is a distinction between two ways in which the mathematics of probability can be applied. A classical application of probability is to describe "frequencies of outcomes in random experiments" [18] such as the concrete outcome of flipping a coin or rolling a dice. However, another approach is to describe "degrees of belief in propositions that do not involve random variables" [18]. This generalization is known as a Bayesian viewpoint and is a viewpoint which the discussion and tools in this thesis takes. Thus, probability distributions will describe assumptions and inferences made about the data and its characteristics (which are very different than the outcomes of a defined system like a flipping coin).

To further exploit the coin analogy, instead of predicting whether the toss is heads or tails, we are predicting which president is depicted on the "heads" side of the coin. This is not really a random variable but is a function of whether, say, a quarter or 50-cent piece is

being tossed. Why is this view point helpful? It allows a model of, say, coin-specific usage for coin-tossing to be established. Factors that might affect this model are heft of the coin and prevalence of the coin in coin purses across the population. Thus, a Bayesian viewpoint creates scientific models of physical systems. These models are different than those of, for one, classical mechanics where a tried and true "law" like $\mathbf{F} = m\mathbf{a}$ is applied. However, they do the same thing: allow us to formalize how we see the system in order to make predictions about what will happen next or how a system will behave.

Now, back to the previous example where $\mathbf{y}$, $\mathbf{x}$, and $\mathbf{G}$ represent the knowns and unknowns of a model. Consider that LVMs can be viewed as a probability problem where instead of predicting a given set of outcomes, the hope is to predict the unobserved variables that have caused a given set of outcomes. That is, instead of describing what will happen when a coin is tossed, the solution should describe the fact that two sides of a coin "probably" caused the roughly 50/50 split between heads and tails in the data.

Mackay explains this situation is an "inverse probability" problem where the probability of unobserved variables is computed given the occurrence of the observed ones. Bayes theorem (Eq. 3.6) describes the conditional probability of $\mathbf{G}$, given the data $\mathbf{y}$ and expresses the inverse probability, referred to as a posterior, that applies to these problems. The posterior describes the total inference we are making about the data $\mathbf{y}$. While the equation itself is not Bayesian in nature, the way LVMs interpret and apply it, is. Thus, the ideas or beliefs we have about the nature and distribution of the latent variables will define the posterior that is applied. It is impossible to quantify (i.e. by flipping a coin over and over) whether each term in this posterior is correct, but it will provide a concrete, mathematical expression of these ideas. This expression is our model, and if it is good, it will predict what the system does.

$$P(\mathbf{G}|\mathbf{y}) = \frac{P(\mathbf{y}|\mathbf{G})P(\mathbf{G})}{P(\mathbf{y})} \Rightarrow posterior = \frac{likelihood\ x\ prior}{evidence} \tag{3.6}$$

In order to illustrate how LVMs construct this expression, consider each term in Eq. 3.6. The evidence, $P(\mathbf{y})$, simply describes the probability of the data $\mathbf{y}$ and is defined by the various degrees of freedom of the system that generated $\mathbf{y}$. The likelihood, $P(\mathbf{y}|\mathbf{G})$, describes the likelihood that a given $\mathbf{G}$ generated the observed data, $\mathbf{y}$, (in other words, the probability of $\mathbf{y}$ given $\mathbf{G}$). LVMs maximize this likelihood according to Eq. 3.7. The iterative optimization outlined in Eqs. 3.8 - 3.10 computes the model and updates (with Eq. 3.10) during each iteration, according to the gradient outlined in Eq. 3.7. The prior, $P(\mathbf{G})$, is the marginal probability of $\mathbf{G}$ (the probability of $\mathbf{G}$ where the value of $\mathbf{y}$ is not constrained). The prior will be exploited in the next section and is fashioned after our hypothesis - an opinion or belief, if you will - about the data and its characteristics.

$$\frac{\partial}{\partial \mathbf{G}_{ji}^{-1}} \ln P(\mathbf{x}|\mathbf{G}) = G_{ji} + x_j z_i \tag{3.7}$$

where z_i is a dummy variable for the iteration according to Eq. 3.9 [18].

$$\mathbf{a} = \mathbf{G}^{-1}\mathbf{x} \ (Step\ 1) \tag{3.8}$$

$$z_i = \phi_i(a_i) \ (Step\ 2) \tag{3.9}$$

$$\Delta\mathbf{G}^{-1} \propto [(\mathbf{G}^{-1})^T]^{-1} + \mathbf{z}\mathbf{x}^T \ (Step\ 3) \tag{3.10}$$

where $\mathbf{a}$ and $\mathbf{z}$ are a dummy vectors used to store values that update $\mathbf{G}^{-1}$ during the iteration and a_i and z_i are their elements, respectively [18].

Think of $\mathbf{G}$ as something that completely defines $\mathbf{x}$ and the totally new space in which $\mathbf{x}$ lives. Once $\mathbf{G}$ is calculated, $\mathbf{x}$ is just a matrix inversion away (Eq. 3.5), and it is this inverse, $\mathbf{G}^{-1}$[5] , that the optimization solves. Remember that $\mathbf{y}$ lives in our, real world and that the elements of $\mathbf{x}$ are latent variables whose true shape and dimension is distorted by our method of measurement. These variables lie on some manifold; the range of options for

[5]Here $\mathbf{G}^{-1}$ is taken as some matrix capabile of recovering $\mathbf{G}$. It is not meant in the literal, mathematical sense of an inverse as this may not be the most prudent method of matrix inversion. [20]

this manifold are defined by $\mathbf{G}$. In order to best untangle this distortion, the optimization allows for any function to define the relation between the real world and the latent space via Eq. 3.9 where the choice for ϕ_i is going to be a nonlinear function where the group of functions, $\phi = [\phi_1,\ \phi_2, ..., \phi_n]$, are known as basis functions. Based on this mapping, the solution for $\mathbf{G}^{-1}$ is updated during each cycle of the iteration in Eq. 3.10. The result of using this nonlinear mapping, is the allowance for great flexibility for the nature of the space containing $\mathbf{x}$.

In this section, we have seen that LVMs provide a general framework for mining, so to speak, a data set for underlying patterns. This framework is much more flexible than the one of PCA and combined with the assumptions added in the next sections establishes a reasonable algorithm for applying to a data set of human motion. Through an iterative optimization, LVMs maximize the likelihood that their solution created the data and maintain the flexibility of defining any nonlinear mapping between the data and the manifold containing the latent variables.

LVMs have a direct link to PCA in the context of the sequence of advances that resulted in GPDM. In their 1999 paper, Michael Tipping and Christopher Bishop [28] apply the framework of LVMs to traditional PCA. Creating so called probabilistic PCA, Tipping and Bishop extend the application of classical PCA to that of a predictive model that is able to provide principle axes for a data set even if it is missing some entries [28]. Building on the work of Tipping and Bishop, Neil D. Lawrence creates an alternate latent variable model representation of PCA[6] [14, 15]. He then extends this alternate model to include the tool of Gaussian process creating a Gaussian-process latent variable model (GP-LVM).

[6]Lawrence does the opposite of Tipping and Bishop. He optimizes the latent variables and marginalizes over the likelihood and calls this dual probabilistic PCA (DPPCA) [15].

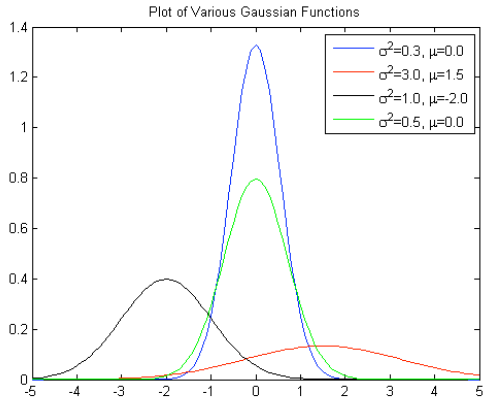


Figure 3.3: **Gaussian Distributions.** A look at the code that generated it (Appendix A) shows that each distribution is given by two parameters: a mean μ and variance σ^2.

3.2.2 Gaussian Process-Latent Variable Models

One of the most commonly used probability distributions is the Gaussian distribution. As shown in Fig. 3.3, they are completely described by two parameters: their mean, μ, and variance, σ^2. Indeed, this distribution has proven to be a useful tool in lots of situations. For example, professors use this to establish how they will distribute the As and Fs they give to their students. Inherent in this policy is the assumption that this tool correctly describes the distribution of student quality. Likewise, Lawrence applies this reasoning to LVMs. Assuming that each variable in a given space follows this distribution, Lawrence shows that the single Gaussian distribution becomes a Gaussian process that now defines a distribution over the functions that may govern the values of these variables. In order to implement this, Lawrence applies a prior (Eq. 3.11) that modifies the probability of the inference made about the data. That is, it modifies the posterior (Eq. 3.6) which is the equation that basically guides this analysis. [15]

$$p\left(\mathbf{G}^{-1}\right) = \prod_{i=1}^{k} N\left(\mathbf{g}_i|\mathbf{0},\mathbf{I}\right) \tag{3.11}$$

where $\mathbf{g}_i$ are the rows of $\mathbf{G}^{-1}$ and $N(...|\mathbf{0},\mathbf{I})$ indicates a Gaussian distribution with $\mu = 0$ and $\sigma^2 = 1$.

Now, μ and σ^2 are functions as well. That is, with the application of this prior, the parameters that describe the Gaussian process become part of the optimization. Marginalization over $\mathbf{G}^{-1}$ (reducing what is a probability of $\mathbf{G}^{-1}$ to being a probability of the parameters on which $\mathbf{G}^{-1}$ depends) results in Eqs. 3.12 and 3.13. Here, the mean will be taken to be zero, and the variance (which is now a covariance function) is given by Eq. 3.15. Once $\mathbf{G}^{-1}$ is marginalized, the likelihood of the latent variables (given by Eq. 3.14) can be optimized according to a manner similar to the one presented in Eq. 3.7. The model must optimize this covariance function in the same way that PCA orients its principle axes in accordance with its optimization of the covariance of the data. In fact, this formalization given in Eq. 3.15 generalizes the implicit assumptions Tipping and Bishop made in [28] when they created probabilistic PCA [15].

$$p(\mathbf{y}|\mathbf{x},\beta) = \prod_{i=1}^{k} p\left(y_i|\mathbf{x},\beta\right) \tag{3.12}$$

$$p\left(y_i|\mathbf{x},\beta\right) = N(y_i|0,\mathbf{x}\mathbf{x}^T + \beta^{-1}\mathbf{I}) \tag{3.13}$$

$$L = -\frac{DN}{2}\ln 2\pi - \frac{D}{2}kn|\mathbf{K}| - \frac{1}{2}tr(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) \tag{3.14}$$

$$k(x_i,x_j) = x_i^T x_j + \beta^{-1}\delta_{ij} \Rightarrow \mathbf{K} = \mathbf{x}\mathbf{x}^T + \beta^{-1}\mathbf{I} \tag{3.15}$$

where β describes the variance of a Gaussian distribution that describes the noise of the model (and it added in the second term via superposition)[15]. This term embodies Lawrence's assumption that the data's distribution (in the real world) will be deviated from the true structure according to a Gaussian distribution.

In his nonlinear interpretation, Lawrence makes use of radial basis functions (RBFs) for ϕ. A radial function is a function for which $\phi(x-c) = \|\phi(x-c)\|$ (where c is any constant center point) holds. The application of these RBFs assumes that data close together in the real world should also be close together in the latent space [30]. Since the functions assumed to be governed by the Gaussian process are no longer linear, this also results in a new covariance function (Eq. 3.16).

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_{rbf} \exp\left[-\frac{\gamma}{2}(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)\right] + \theta_{bias} + \theta_{white}\delta_{ij} \tag{3.16}$$

where γ, θ_{rbf} θ_{bias}, and θ_{white} are parameters that will now be optimized with $\mathbf{x}$ during the LVM's optimization.

Lawrence has applied a prior that incorporates the assumption that the distribution of possible manifolds is described by a Gaussian distribution. This description provides us with some concrete rules that, once one latent data is defined with some rough estimate, guide the LVM's optimization accordingly. This optimization is initialized using PCA. In subsequent papers Lawrence modifies his prior (Eq. 3.11) to constrain the data to be sequential in the latent space [16]. The next section will discuss how Jack Wang and his collaborators' choice of prior differentiates their algorithm from GP-LVMs.

3.2.3 Gaussian Process Dynamical Models

Each pose in the data set has a certain subset of poses that are likely to follow it. Furthermore, once a pose occurs, there is a probability distribution that describes what the next pose will be. Exploiting this reasoning is a good idea and is the addition Wang and his collaborators made to Lawrence's GP-LVM [31, 32]. This reasoning can be visualized as a field placed over the latent space in which any given location is constrained to point to another [17]. This is plotted for a two dimensional latent space in Fig. 3.4.

Wang and collaborators apply their dynamic prior by beginning with a latent variable

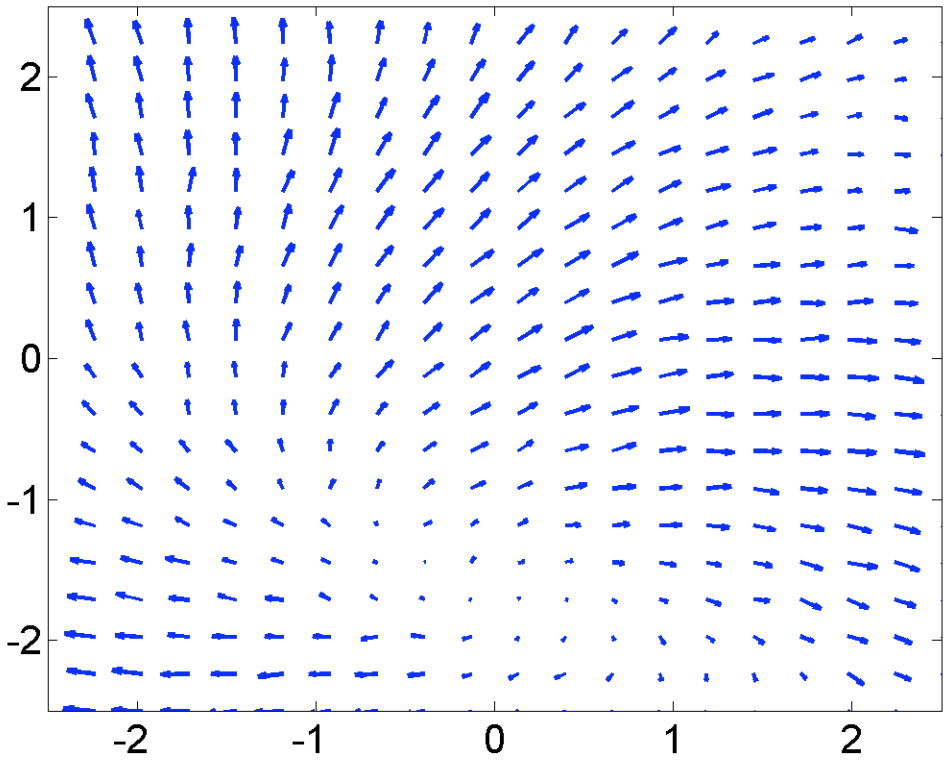


Figure 3.4: **Dynamical Constraint for the Latent Space.** This constraint creates a model that accounts for the temporal relationship of the data points. Figure courtesy of Neil D. Lawrence.

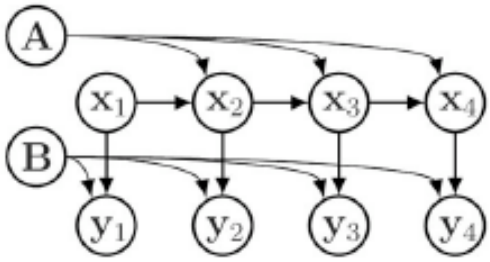


Figure 3.5: Mapping for GPDM. This visualizes the parameterization of $\mathbf{x}$ and $\mathbf{y}$ by their corresponding matrices $\mathbf{A}$ and $\mathbf{B}$ and also the generation of $\mathbf{y}$ from $\mathbf{x}$ [32].

mapping that applies first order dynamics to the model Eq. 3.17 - 3.18. This mapping is visualized in Fig. 3.5. The combination of this new dynamical mapping, as well as priors and covariance functions similar to those used in GP-LVMs results in a new prior that can be applied to the latent variable model that includes both the aspects of Gaussian process over the function space as well as dynamical constraints that accounts for the time dependence of the data. Practically speaking, this just results in more parameters (termed hyperparameters) to be optimized.

$$x_t = f(x_{t-1}; \mathbf{A}) + n_x \tag{3.17}$$

$$y_t = g(x_t; \mathbf{B}) + n_y \tag{3.18}$$

where n_x and n_y are time dependent additive Gaussian noise, f and g are linear combinations of nonlinear functions parameterized by weights given in $\mathbf{A}$ and $\mathbf{B}$, i.e. $f(x_i; \mathbf{A}) = \sum_i a_i \phi_i(x_i)$ and $g(x_i; \mathbf{B}) = \sum_j b_j \psi_i(x_i)$ [32].

The derivations and exact equations that produce this new prior have been omitted, but do not forget that this is still a latent variable model. The additions to our error-correcting code are somewhat superficial. This final algorithm simply makes room for greater control over the assumptions about each term in the probability that governs the inference about the data (Eq. 3.6). Now, the nonlinear mapping functions, which are unknown, are presumed to

follow the rules of a Gaussian process. This provides a convenient mathematical framework for specifying the type of functions. Specifically, this framework is manifest as the covariance function (which can be comprised of linear, polynomial, or radial (or some combination there of) basis functions) and dynamical mapping functions ϕ and ψ. This controls the mapping between the real world where $\mathbf{y}$ was measured and the space containing a manifold on which some latent variables $\mathbf{x}$ lie most naturally and can therefore be recognized. Additionally, the probability describing the distribution of these manifolds (the prior) is taken to be Gaussian in order to best suit LVMs to the case of human motion.

Chapter 4

Motion Capture

In order to take this method from the simple case of C. elegans to the many times more complex one of a dancer, let us look to the representation of three dimensional dance postures in other computational applications for a complete, yet computationally reasonable, description in order to recast dancers' internal intuition of movement. Perhaps the most glaring use of 3-D representation is in the business of computer animation. The desire to better animate characters such as Gollum in the series of blockbuster hits The Lord of the Rings has, in part, driven the development of so called motion capture technology. This technology allows real-life motion (often human) to be incorporated with computer animation allowing animators to create more realistic depictions. Sophisticated motion capture capabilities have been developed on university campuses, often in conjunction with industry leaders. As a result, this technology is becoming a more accessible and robust method of data collection. For example, Walt Disney Company has opened up a global research lab at Carnegie Mellon University that will work in conjunction with their Graphics Lab [24].

Figure 4.1: **Example of a VICON motion capture camera setup.**

4.1 Hardware

Key to this project is the contacts established at labs that specialize in this research. Professor Magnus Egerstedt at Georgia Institute of Technology and Professor Dimitris Metaxas at Rutgers University's Center for Computational Biomedicine Imagine and Modeling (CBIM) run two such motion capture labs that have advised this thesis. These labs use VICON optical motion capture systems such as the one pictured in Fig. 4.1. Between four and twelve of these cameras strobe a designated area with infrared light in order to segregate motion detection to a wavelength different from those that might otherwise be shared with possible interferences such as equipment signals and lighting conditions in the room. Reflective markers are placed on a test subject, and those reflections are collected by the cameras' sensors. [12]

4.2 Software

Arizona State University researchers Stjepan Rajko and Gang Qian describe the next steps as three distinct processes. First the system must pick out these reflective points in each frame - this step is called tracking. Once the path of each marker is recorded, the system begins labeling each marker using some pre-defined model for the test subject. A model appropriate for human motion is as literal (though much simpler) as an anatomy textbook;

```
# AST/ASF file generated using VICON BodyLanguage
# ------------------------------------------------
:version 1.10
:name VICON
:units
  mass 1.0
  length 0.45
  angle deg
:documentation
   .ast/.asf automatically generated from VICON data using
   VICON BodyBuilder and BodyLanguage model FoxedUp or BRILLIANT.MOD
:root
   order TX TY TZ RX RY RZ
   axis XYZ
   position 0 0 0
   orientation 0 0 0
:bonedata
  begin
     id 1
     name lhipjoint
     direction 0.673618 -0.716094 0.182886
     length 2.63917
     axis 0 0 0  XYZ
  end
.
.
.
  begin
     id 30
     name rthumb
     direction -0.707107 -6.34891e-011 0.707107
     length 0.768342
     axis -90 -45 -2.85299e-015   XYZ
    dof rx rz
    limits (-45.0 45.0)
           (-45.0 45.0)
  end
:hierarchy
  begin
    root lhipjoint rhipjoint lowerback
    lhipjoint lfemur
.
.
.
    rwrist rhand rthumb
    rhand rfingers
  end
```

Figure 4.2: **Acclaim Skeleton File (.ASF) Sample.** This figure demonstrates the .ASF file format. Example taken from CMU motion capture database.

the system applies user defined labels such as “upper back,” “right thigh,” and “left wrist.” Finally, a representation of the data must be produced during pose estimation. The specifics of this depend on file format, but it in some way combines the model and the recorded motion. A representation could be in the form of joint angles of the human skeleton. [21]

The model for labeling by VICON systems is defined by an Acclaim Skeleton (.ASF) File (Fig. 4.2. Paired with this file and completing the data set is the Acclaim Motion Capture (.AMC) file (Fig. 4.3. The sections of the .ASF file are dubbed: **:version**, **:name**, **:documentation**, **:units**, **:root**, **:bonedata**, and **:hierarchy**. An example file is shown in Fig. 4.2. The first sections are quite self-explanatory, but the later sections contain the essence of how the data is represented and thus, are worth a brief discussion.

The root section defines the parent of the models’ hierarchy. Each element of **:bonedata**, which can be thought of as the code equivalent of a physical bone, is eventually

```
#!OML:ASF F:\VICON\USERDATA\INSTALL\rory3\rory3.ASF
:FULLY-SPECIFIED
:DEGREES
1
root 4.40047 17.8934 -21.0986 -0.943965 -8.37963 -7.42612
lowerback 11.505 1.60479 4.40928
upperback 0.47251 2.84449 2.26157
thorax -5.8636 1.30424 -0.569129
lowerneck -15.9456 -3.55911 -2.36067
upperneck 19.9076 -4.57025 1.03589
head 8.94923 -2.21289 0.178777
rclavicle -3.22404e-015 1.78906e-014
rhumerus -22.1457 -10.4191 -82.5993
rradius 31.043
rwrist -1.10026
rhand -18.32 23.6535
rfingers 7.12502
rthumb 7.96145 -6.17659
lclavicle -3.22404e-015 1.78906e-014
lhumerus -35.9023 4.66766 83.7329
lradius 27.8853
lwrist 5.29873
lhand -24.5852 29.6298
lfingers 7.12502
lthumb 1.91219 59.6009
rfemur 3.63674 10.1003 34.5122
rtibia 25.6934
rfoot -20.9622 -4.01225
rtoes 5.06152
lfemur -27.9936 -4.46438 -9.04925
ltibia 22.5608
lfoot -6.36063 8.24507
ltoes 1.09177
2
root 4.40608 17.8851 -20.9321 -0.860525 -8.41227 -7.49779
.
.
.|
```

Figure 4.3: **Acclaim Motion Capture File (.AMC) Sample.** This figure demonstrates the .AMC file format. Example taken from CMU motion capture database.

contained within the order and convention defined by these two sections. The hierarchy uses two basic mechanisms, translation and rotation, to arrange bone locations and connections. These are defined for the parent by **position** and **orientation**. The **:bonedata** element allows for intuitive definition of the remaining model. Each bone is given what essentially amounts to a defining vector (**length** and **direction** define how each bone branches off the model) as well as a global orientation vector (**axis**) and degrees of freedom (**dof**) with corresponding **limits** to prevent unrealistic interpretations of motions (when desired). These bones are locked into the hierarchy as shown in Fig. 4.2. Thus, the connectivity of the test subject model is defined according to the organization of the human body and thus, preserves the relationship between the body's joints and appendages. Figure 4.3 shows an .AMC file, which provides the animation information from the motion recording. The file is simply delineated by frame numbers, and within each frame are coordinates that provide animation for each channel, where every degree of freedom per bone is a channel. [13]

Equation 4.1 casts this problem to the more general mathematical framework of 3.1 and 3.2. Each frame of animation is characterized by the matrix given in 4.1, and thus, the Acclaim file format, with which the data for this thesis is stored, has cast the representation of each frame of human movement as a $R^{(numberofjoints)x(numberofdofs)}$ dimensional problem. Furthermore, Eq. 4.1 can be expressed as a matrix that is indexed by frame number and degree of freedom. This is given in Eq. 4.2. Each of these matrices is another way to express the data in the .AMC files and will be employed by the analysis techniques in the next chapter.

$$\mathbf{f} = \sum_{i}^{n} frame_i = \sum_{i}^{n} \begin{bmatrix} root_i^{tx} & root_i^{ty} & root_i^{tz} \\ lft\_femur_i^{rx} & lft\_femur_i^{ry} & lft\_femur_i^{rx} \\ lft\_tibia_i^{rx} & lft\_tibia_i^{ry} & lft\_tibia_i^{rx} \\ lft\_foot_i^{rx} & lft\_foot_i^{ry} & lft\_foot_i^{rx} \\ rt\_femur_i^{rx} & rt\_femur_i^{ry} & rt\_femur_i^{rx} \\ \vdots & \vdots & \vdots \\ neck_i^{rx} & neck_i^{ry} & neck_i^{rx} \\ head_i^{rx} & head_i^{ry} & head_i^{rx} \end{bmatrix} \tag{4.1}$$

where n is the number of frames of motion and the convention for each term is $joint_{frame\ \#}^{dof}$.

$$\mathbf{F} = \begin{bmatrix} root_1^{tx} & root_1^{ty} & root_1^{tz} & lft\_femur_1^{rx} & \cdots & head_1^{ry} & head_1^{rx} \\ root_2^{tx} & root_2^{ty} & root_2^{tz} & lft\_femur_2^{rx} & \cdots head_2^{ry} & head_2^{rx} & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ root_n^{tx} & root_n^{ty} & root_n^{tz} & lft\_femur_n^{rx} & \cdots & head_n^{ry} & head_n^{rx} \end{bmatrix} \tag{4.2}$$

where n is the number of frames and the convention for each term is $joint_{frame\ \#}^{dof}$.

Chapter 5

Data and Analysis

Now it is time to apply the dimensionality reduction techniques outlined in Ch. 3 to the collection method and software implementation of Ch. 4. This chapter presents the methods for data collection and analysis. It outlines the software used for both and touches on the file formats involved. The many tedious steps should not confuse the basic ideas of the previous chapters. In this chapter the organization of the .ASF/.AMC file format is simply funneled to and from the appropriate formats that connect this organization convention to those of the data collection and data analysis (linear and nonlinear) software. In effect, the process outlined in this chapter allows for a discourse between the software used. This discourse is required in order to allow each software element to access the same empirical information, and since each program comes from different realms of computational solutions, they typically employs their own data formats. Likewise, the specific commands used in each program will be touched on, but as these commands are simply artifacts of the software implementation, making them explicit will do little to add meaning to our discussion, particularly as they may change with the next update of the software.

5.1 Data Collection

Data was collected January 20-21, 2009 at Georgia Institute of Technology in Prof. Magnus Egerstedt's robotics lab. Previously, the motion capture facility was used to track small robots within an enclosed area. The capture volume was about 5x10x6 feet in dimension. This volume was created by the 8 Vicon M2 cameras (1280x940 pixels, recording at 100 frames/sec) and was limited by the orientation of these cameras in the rectangular facility (i.e. the capture volume would be greater with space for a circular arrangement of the cameras).

A pre-made Vicon skeleton template (iQ_ACTORExample_V5.vst) was used to model the geometrical constraint of the human body. This skeleton is represented in software much like the .ASF file.[1] This software representation outlines where on the body the computer expects to find markers. According to this outline, 47 reflective markers were placed on the dancer's body as shown in Fig. 5.1.

The first day and a half of collection was spent calibrating the placement of these reflective markers and ensuring that the software was correctly interpreting them and producing sensible models for motion. The software used for this calibration process and the subsequent data collection was Vicon Vision IQ[2] . In order to calibrate an anatomically accurate skeleton for this subject, the dancer stood in a "T-pose" (Fig. 5.2). The T-pose (accompanied with small perturbations) ensures that the cameras can clearly see and track each reflective marker. Next, the dancer moved around to demonstrate the range of motion of each joint. For example, these movements included a comprehensive series of arm flailing, leg kicking, back bending, and spine wiggling. A thorough range of motion measurement allows the skeleton to remain calibrated over the full range of postures the dancer might take. The calibrated skeleton (in the form of a .VSK file) will be applied to all of the

[1] The difference is superficial. .VST files support Vicon motion and skeleton files .V/.VSK as opposed to the Acclaim format.

[2] Version 2.5, build 275, rev 10924

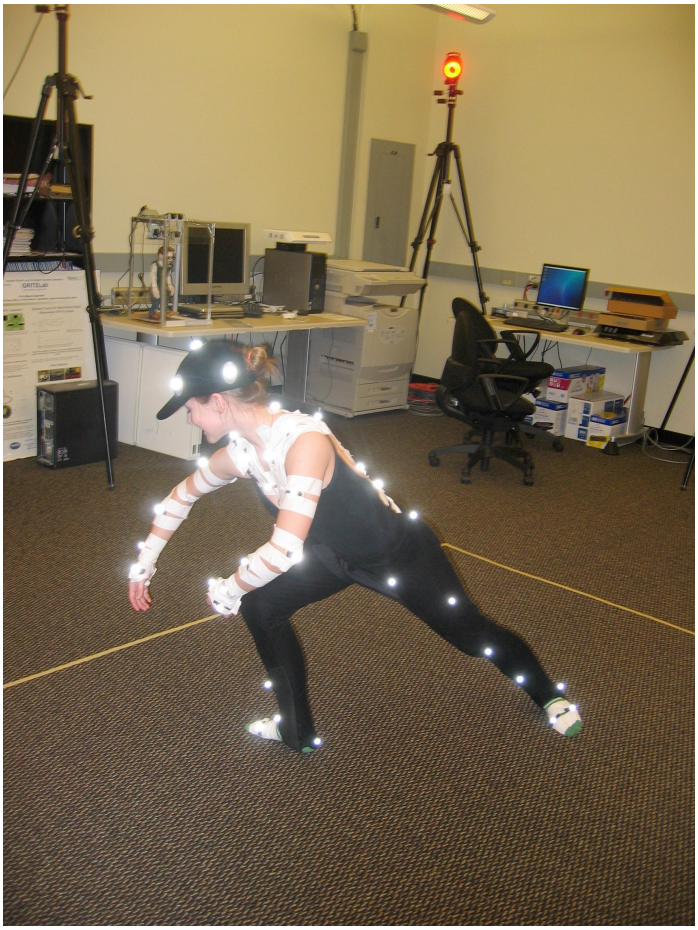


Figure 5.1: **Dancer used for data collection.** Here the reflective markers glow from the camera flash. Photo courtesy of Amir Rahmani.

recorded motion sequences for that subject according to the stages outlined in Fig. 5.2. The motion sequences that the dancer performed are listed below. Refer to Sec. 5.2 for a discussion on the anticipated use of each sequence.

Assortment - 4:53min A variety of pedestrian motions. For example, walking, waving, bowing, etc.

Ballet 1 - 1:14min Classical ballet sequence where each step is performed individually and then strung together: tomb, pas de bourre, glissade, and pas de chat.

Ballet 2 - 3:37min Classical ballet sequence where each step is performed individually and then strung together (and executed twice): tomb, pas de bourre, jeté, contretemps, tombé, pas de bourrée, fifth position, passé, fourth position (preparation), pirouette, balancé en avant, balancé en arrire, tombé, pas de bourrée, and grand jeté.

Ballet 3 - 3min Classical ballet sequence where each step is performed individually and then strung together: développé croisé devant, développé écarté, battement croisé devant, battement effacé derrire, plié, penchée, pas de bourrée, tendu, port de bras, and an arabesque turn.

Ballet 4 - 3:11min Classical ballet movements: battement effacé devant, fouetté, battement effacé derrire, tour jeté (the idea behind these four movements is that the first three compose the fourth), and six double pirouettes.

Ballet 5 - 5:35min Classical ballet barre movements: plié, grand plié, relevé, and jumps with port de bras in each position (1-6).

Improvisation - corrupted Ten minutes of improvisation in the style of contemporary or modern dance[3] .

Choreography - 1:42 and 1:31min This choreographed sequence was recorded twice. Here, the term "Choreography" indicates not just that the steps were predetermined (as all of the steps for each sequence except the improvisation were predetermined) but that there was some artistic (not purely aesthetic) viewpoint behind the ordering of the steps. It was choreographed by the subject and graduate student Peter Kingston and was inspired from the Improvisation sequence and features large, sweeping movements that draw the dancer off center.

[3]This sequence's data was essentially corrupted when a skeleton and its associated motion (and associated skeleton labels and joint connectivities, etc) were saved as twice, making one file that was too large to be opened by Vicon's software.

Once the infrared cameras have recorded the motion of the markers, there are still several critical steps required to produce usable data. However, the physical measurement is complete, and these steps happen in a digital language. The data begins as the moving points in space recorded by the infrared cameras (Fig. 5.2(a)). Next the calibrated skeleton is applied to this raw data; the skeleton matches up with the a specific frame that the user labels "T-pose" (Fig. 5.2(b). Up until this point the motion is said to be "unsolved" because it is not expressed in terms of any constraints (a skeleton). In this raw format (file extension '.c3d'), the file sizes are prohibitively large (on the order of Gigabytes). After "solving" the data, it can be expressed in terms of the skeleton and the file size will shrink to the order of Megabytes (for the sequence lengths in this collected data set). In order to solve the data, Vicon's Blade[4] software fits the raw data to the calibrated skeleton (Fig. 5.2(c)) and can now export this to a simplified file format (an .ASF/.AMC file pair). As shown in Fig. 5.2(d), the data is now represented as rigid links or "bones" which are defined by the skeleton and the visualization in Vicon Blade reflects this change as well.

Additionally, two sequences, listed below, from Carnegie Mellon University's motion capture database[5] will be analyzed.

Walking - 0:02min Subject # 2. Three steps beginning with the left foot.

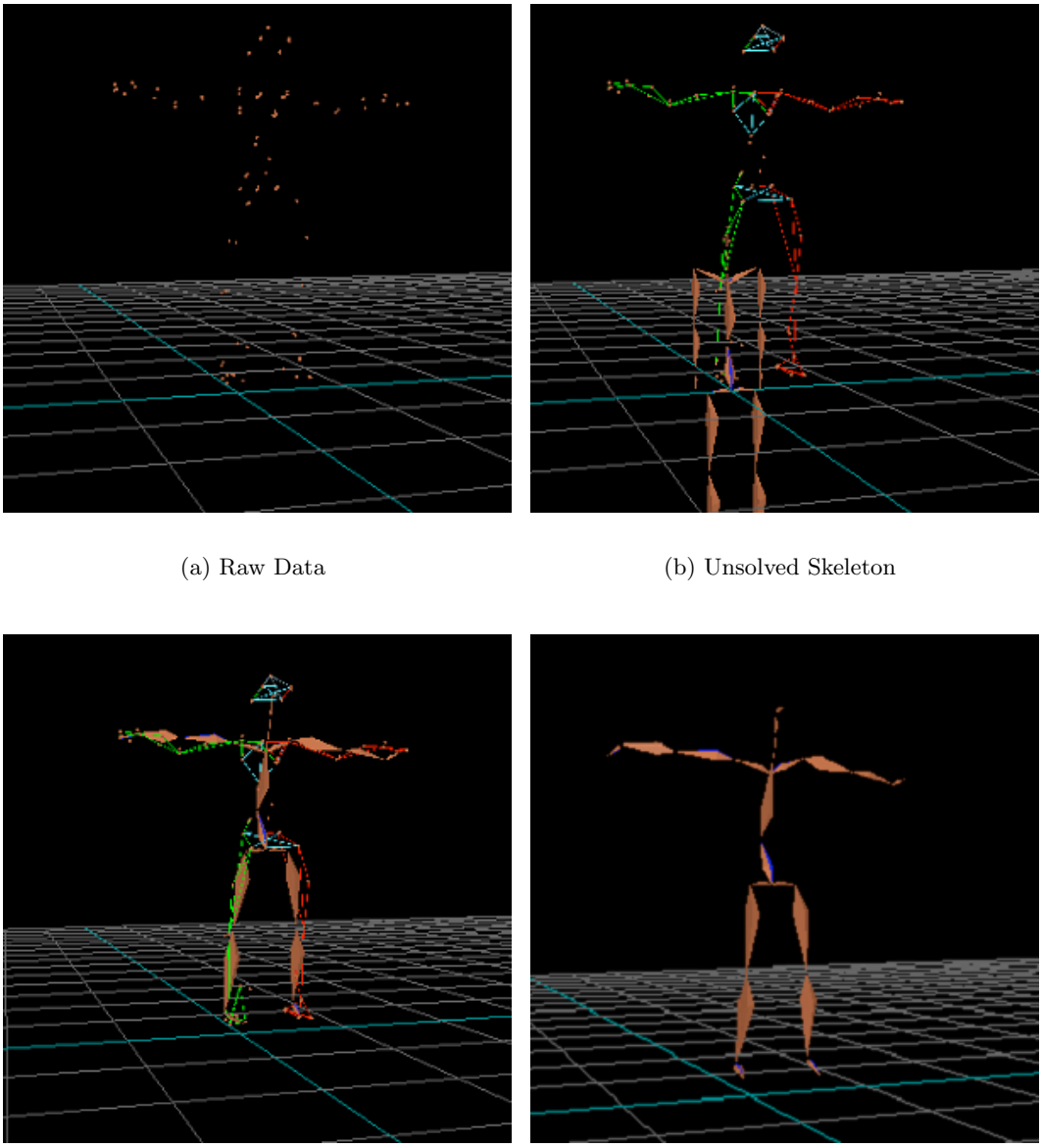
Arabesque - 0:04min Subject # 5. An arabesque and a step into a back combré.

5.2 Experimental Design

Now that each sequence has been explicitly outlined, the logic behind them will be discussed. An over-arching principle was simply to get as much movement recorded as possible with the short two days available. As two sets of researchers were using the movement some sequences were more specifically tailored to projects other than this thesis. For instance, (1)

[4] Version 1.5.195

[5]



(a) Raw Data

(b) Unsolved Skeleton

(c) Solved Skeleton

(d) .ASF/.AMC Representation

Figure 5.2: **Motion capture data processing.** These figures show the data as it progresses through the digital steps involved in collection.

the pedestrian movements were collected with the hope of applying authentic acceleration profiles to a puppet with limited range of motion, (2) and at the beginning of each classical ballet sequence the steps were performed individually to provide the "answer" to a researcher interested in finding a way to segment movements into chunks that make sense even to the dancer who performed them.

Nevertheless, these sequences will still be analyzed (hopefully interesting results) in the next chapter. The ballet sequences captured are ones typical of classroom excercises (however, these exercises often have their roots in performance) and are governed by the aesthetic principles of classical ballet: long, graceful lines, movements which often begin and end in a specific pose, and a typically frontal presentation of the body, to name a few.

The motions captured specifically for this thesis were the improvisation and subsequently choreographed sequences, and although one of these failed to be analyzed due to file size errors, discussion of the intended experimental design is warranted.

Improvisation is a tool used by dancers to find new movement patterns through an organic process (rather than through the environment of a classroom where it is often superficial imitation that is the channel for learning). Additionally, it is often employed as a starting place, a "word vomit," if you will, for choreography. During improvisation, inspiration for the movement is gathered from the environment through all the senses, namely, visual, olfactory, tactile, etc., and is filtered by the subject's mind to generate impulses from which movement issues. In some sense, the goal of improvisation is for the dancer to let go of high level thinking and become as submerged in the physicality and experience of moving as possible.

This is perhaps the closest to the behavioral state of simple organisms, such as the C. Elegans, that humans ever reach. It is a state in which the subject is not thinking "What are my plans tonight?" or "I really need to get a new pair of shoes..." but is simply moving. This thesis proposes, then, that motion capture of improvisation is the closest reproduction of the experimental setup of Stephens et al. for the case of humans. Thus, it is in this sort

of behavioral context, where the dancer moves in an unconstrained way, that this thesis explores, despite the loss of this data.

Going further, there is a question of what it is about the movement and its structure that changes when artists choreograph or set these movements in stone. With a truly parameterized improvisation (that say, had a basis of four or so postures like the C. Elegans), the structure of choreography could be explored using a behavioral lens. This analysis would begin to answer questions like "What changes does our creativity implement that makes the captivating works of William Forsythe, Mark Morris, or Twyla Tharp?" on a quantitative and scientific basis.

5.3 Linear Analysis

An undergraduate at the University of Colorado, Jonathan Olson (advised by Prof. Elizabeth Bradley) has developed code that performs the same analysis described in Sec. 3.1 on motion capture data in this .ASF/.AMC format. Olson's code, written in the Python language, forms matrices from the joint angles of each of the n frames of motion. Then, the code performs PCA on each large matrix (given in Eq. 4.2) and, finally, converts the reduced data back to an .AMC file. Even though this matrix is larger in size than the example presented in Sec. 3.1, it still follows the same steps, starting with computing the covariance between the terms of $\mathbf{F}$. The user can specify how many and which components to keep for playback, and this reduced .AMC file is reanimated with the original skeleton.

5.4 Nonlinear Analysis

The GPDM algorithm is applied using software from one of its key developers, Neil D. Lawrence. Lawrence has made his C++ code and supporting MATLAB toolboxes available online along with demonstrative examples. His GP-LVM C++ software[6] includes as an

[6] Version 0.12

optional functionality Wang et al.'s dynamical prior. The data size limitations of this software are about 1000 points with a maximum 1000 iterations for optimization.

The GP-LVM C++ software was compiled, installed, and run using a Cygwin command line. Cygwin is a Unix-like environment for Microsoft Windows that supports the GNU Compiler Collection (GCC) and various libraries (Fortran/g77 and for linear algebra: LAPACK and BLAS) utilized by the code. Once installed, this software requires data in the support vector machine light (.SVML) format. Anton Schwaighofer's MATLAB tool box was used to write the .ASF/.AMC (as parsed by Lawrence's MOCAP MATLAB toolbox into the form of Eq. 4.2) data to this format. Before writing to a .SVML file, the matrix **F** can easily be manipulated for more manageable analysis. Taking only a subset of the sequence in order to look at one movement or a shorter series of movements allows for more focused analysis, and downsampling the data, which with 100 frames per second of motion, is already quite detailed, reduces running time significantly while still capturing the description of the movement.

The **learn** command in the GP-LVM C++ code implements the optimization outlined in Sec. 3.2. The options of this command allow the user to specify many parameters for the computation. Namely, a dynamical prior with an RBF covariance function with an inverse width of 0.01 and a noise level fixed by a signal to noise ratio of 20 was specified. The output of the algorithm is saved as a .MODEL file.

In order to display the results, this model file was converted into .DAT files by the **gnuplot** function built into the software. As might be expected, these .DAT files, one for the variance and one for the latent variable data, can be visualized in Gnuplot. Gnuplot supports several plotting options and produces a PostScript (.PS) file that was converted to a .PDF using an online converter or to a .PNG image using GhostScript. The images of the produced models are displayed and discussed next.

Chapter 6

Results

The results of the analysis outlined in Ch. 4 are now presented. The presentation will progress as follows:

1. The linear analysis outlined in Sec. 3.1 is applied to data available online from CMU's database.
2. The nonlinear analysis outlined in Sec. 3.2 is applied to the data available on CMU's database.
3. The nonlinear analysis is applied to the data collected as described in Sec. 5.1[1] .

Conclusions about individual results will also be discussed, but some over-arching conclusions will be framed in the context of the next chapter.

6.1 Linear Dimensionality Reduction

Here, the decompositions of the Walk and Arabesque sequences as found by PCA will be presented. The analysis is twofold. To begin, individual components are viewed; these components are ordered according to their eigenvalues where higher eigenvalues correspond

[1]Unfortunately, unforseen issues with Vicon's .ASF/.AMC export tools prevented this data set to be analyzed with the linear technique.

to components which describe more of the data's the structure, that is, the component with the highest eigenvalue is the first component. Second, the sums of these components are viewed. Remember that each component of this decomposition is an animation[2] that is the same length as the original sequence. It is expected that the first component will give the largest contribution toward the full sequence and that the second, third, and fourth components contain progressively higher order, less important details of the sequence.

A component is considered informative if it represents real human motion. Since this decomposition cannot constrain the results to be human motions, the components it produces may be nonsensical in the context of human motion. These are components whose animations jerk, jump, and pop as if being displayed by a skipping DVD player. These components do little to elucidate behavioral information from the data and are considered here to be nothing more than an exercise in reducing the file size of an animation. These are typically higher order components. Additionally, if a finite, reduced number of components can be used to describe the original motion, an animation which includes all of them will very closely resemble the original.

It is to the disadvantage of the reader that these results are dynamic animations that are impractical, if not impossible, to visualize on paper. Alternatively, the text of the matrices filled with numbers that define these animations could be included, but that would produce pages of numbers for every few seconds of data. This would simply overwhelm and explain little. Instead, these results will be described in words, qualitatively, as seen through the human eye. Before scoffing at this rudimentary analysis, take a moment to consider whether any better tools exist for this task of analyzing whether a motion is authentically "human." In fact, it is one of the motivations for this thesis that there are none. Additionally, humans are excellent at differentiating human movements - we can even identify *specific people*, just from the way they move. So, if the next sections are frustrating, consider it further motivation for why the ability to quantify human motion needs to be gained.

[2]Specifically, they are .AMC files that correspond to the original .ASF skeleton file.

6.1.1 Walking

Like each of the components, the first component of CMU's Subject #2's walking sequence decomposition is a short, 298 frame clip. In it, the figure is clearly portrayed to be lifting their right foot as their leg moves through about 60 degrees of motion. Simultaneously, the are shifting their weight to the left as indicated by a lowering of the entire left side of the body that starts in the left foot and travels upwards to the left shoulder. This occurs three times (corresponding to the three steps in the original sequence). The motion is smooth, and while viewing, it is easy to imagine doing the action yourself[3] .

The motion of the second component is less smooth, though not yet jerky. The figure bends both legs, and then extends the right, only to bend it back to meet the left again. The lower legs move through about 45 degrees of range. This motion can be perceived through frame by frame analysis, but when viewing the footage, only the impression of some swift kicking in the lower limbs is discernable.

It is a stretch to describe the clip of the third component as an animation of human movement. In the clip, the figure pulses back and forth two times, bending backward at the shoulders and hips. Although the motion is smooth, doubt in the movement's authenticity comes from the quickness with which the figure repeats a motion. A similarly founded doubt is escalated in the fourth component's animation. Here, a similar pulse occurs in the lower left foot causing what looks like a shake through the whole body. This pulse jumps and jerks between endpoints in a disconnected way that is uncharacteristic of human movement.

Based on the definition of a meaningful component, outlined above, it is immediately clear that the third and fourth components do not contain much information that pertains to the human behavior of walking. This is confirmed by the combination of the first two

[3]This intuition may also have a neural correlate in what have become known as mirror neurons. These neurons are thought to allow humans to internalize the actions of others on an empathetic level and make learning from imitation possible. [22]

components as the resulting animation shows a person walking. The recreation is not the original: a certain loss of quality can be discerned. This difference is comparable to watching an old, analog TV with a bit of static. Additionally, if the third and fourth components are included, only the most minimal improvements are made. Thus, these and the other 53^4 higher order terms are taken to be uneccesary for the basic movement.

6.1.2 Arabesque

In the first component of this more complex movement of CMU's Subject #5, there is already unnatural jerking in the animation. The decomposed reincarnation seems to reach forward once and then, after drawing back to a more or less upright position, reaches farther forward, again returning to an upright position with the arms held low and extended away from the body. In the lower body, the feet slide forward and backward slightly. There is a slight vibration of the entire skeleton for which an anatomical origin is difficult to pinpoint. The jerky vibration occur at what are approximately the end points of these reaches, with the most pronounced vibration at the end of the clip. In breif, this component is an arm reach, interrupted by inhuman jerking.

Likewise, the second component contains a movement that is interspersed with abrupt shaking. Between the seemingly meaningless static that the sequence begins and ends with, there is a clear extension of the figure's left leg behind and to left of the figure about 45 degrees off the ground. This motion is followed by more jittering, and then, the left leg lifts in front of the body with the knee bent and foot hanging below it, slightly off the floor.

The third component also contains human behavior mixed these jarring jerks. Beginning with the torso twisting toward the right with arms straight out from the shoulder socket, amid the jerks, the figure is depicted as twisting the opposite way, towards the left side of the body, and returning to a neutral position. As if letting the movement carry into the lower limbs, the left foot extends forward as well.

[4]For a total of 57 components since the matrix $\mathbf{F}$ has 57 columns 4.2).

Surprisingly, in a reverse of the previously observed trends, the fourth component shows perhaps the smoothest and most behaviorally consistent movement yet. The motion displayed looks very similar to a weak attempt at a jumping-jack. In this clip the jerking is slow enough that it looks like the figure is retracting their arms in hesitation. The arms of the figure are extended, and after two such hesitant reaches down and up, the arms raise overhead in a smooth trajectory. The knees bend in rhythm with the arms three times.

In order to recreate the motion of the original arabesque sequence, it takes about six components. The animation that includes those six components is even more poor in quality than the original sequence, but each motion is essentially represented. As with the Walking sequence's third and higher order components, which began looking more or less similar to each other, the seventh component of the Arabesque sequence contains only the jerky vibrations while the sixth contains some lower limb movements.

6.2 Nonlinear Dimensionality Reduction

First, the results presented by Wang and his collaborators [32] will be reproduced. The simplicity of these results provides a perfect framework for beginning to understand and interpret the later results. Next, new results from several sequences collected for this thesis are presented. See Appendix B for more results not discussed here.

The results of the nonlinear analysis for the Walking sequence are shown in Fig. 6.1 and 6.2. It is immediately evident that these results show a clear, circular pattern. It is even possible to convince yourself that there are three such circular patterns, corresponding to each repeated step in the sequence.

This plot is a two dimensional projection of the latent variable space found by the GPDM algorithm; it is a tool for our visualization of something that is many dimensional. Remember that the latent space is a mathematical construction found through iterative optimization and used to find a decomposition of the data. GPDM's decomposition is

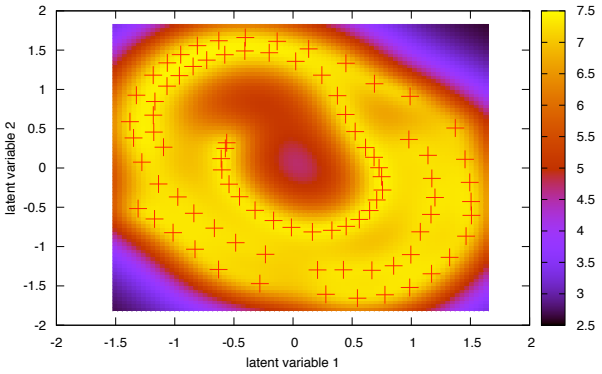


Figure 6.1: **Results of Nonlinear Decomposition: Walking Sequence.** A two dimensional projection of the latent variable space. A 3-D view of this plot is shown in Fig. 6.2. In this sequence the subject takes three steps beginning with their left foot. Motion capture data courtesy of Carnegie Mellon University motion capture lab.

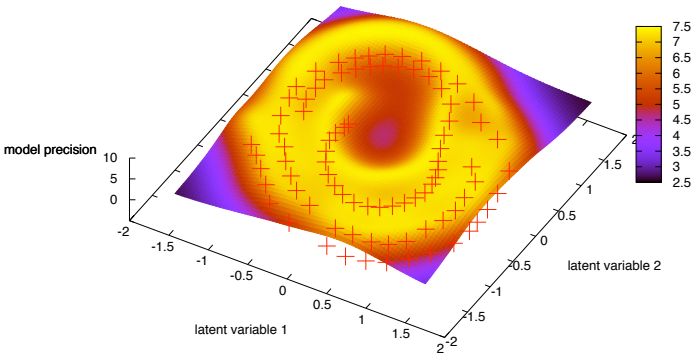


Figure 6.2: **Alternate Results of Nonlinear Decomposition: Walking Sequence.** A three dimensional view of a two dimensional projection of the latent variable space. In this sequence the subject takes three steps beginning with their left foot. Motion capture data courtesy of Carnegie Mellon University motion capture lab.

like the decomposition PCA produced, except that PCA decomposed the data as it was expressed in the three dimensional world in which we live. Instead of performing the decomposition in these terms, GPDM uses a many dimensional space that has a nonlinear map to our world. Units like meters and degrees do not make sense in this space. The units on the axes do not have units in the usual sense. The model is expressed in terms of two latent variables. These variables are nonlinear combinations of the original data's joint angles as projected to the plane of the Fig. 6.1. However, this plane is for our visualization and is not where the model lives.

The red "+" signs are the data points of the original sequence as expressed in this projected latent space. The gradient plotted behind the "+" signs is the model's precision, and the scale given on the color map is inversely related to variance. Thus, the yellow regions are those postures which have a high likelihood of being a part of the modeled sequence. Think of these like error bars.

In the spirit of the successful pattern production by the GPDM of the simple Walking sequence, two more cases of simple, repeated movements will be presented. These movements are the basic demi plié. One of the most simple and fundamental movements in classical ballet, this motion is simply the bending of the knees - but only as far as the Achilles tendon at the back of the heel will allow while maintaining full contact with the ground during the motion. The repeatability between this movement when it is performed in three different orientation of the legs is tested in Figs. 6.3, 6.4, and B.3. The striking similarity between the meandering double-"S" structure of the graphs is evident and displays the GPDM's ability to discern movement from posture. For completeness the corresponding three and two dimensional views are included in Appendix B. Additionally, the topography shown by these models indicates that the precision of this model is better than for the Walking GPDM due to the narrow regions of yellow. These regions are shown as sharp peaks in the three dimensional views.

Finally, comparison between a classical ballet and modern dance sequence will be made.

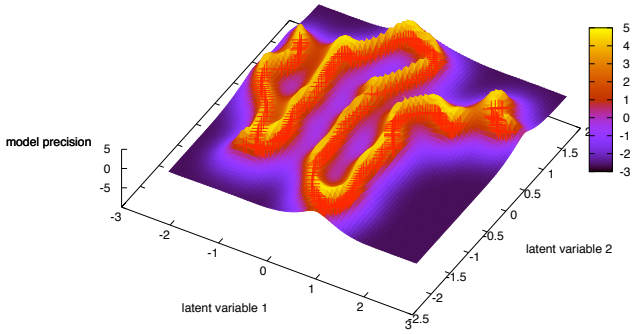


Figure 6.3: **Results of Nonlinear Decomposition: Five Pliés in First Position with Port de Bras.** A 3-D view of a two dimensional projection of the latent variable space. The 2-D view is shown in Fig. B.1. In first position, the dancer stands with the backs of their heels pressed together while the toes point outward away from the body[6] .

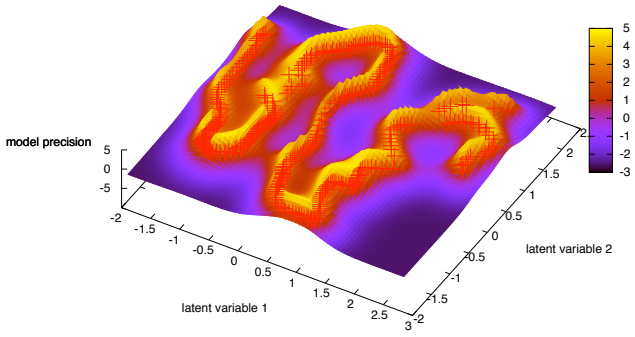


Figure 6.4: **Results of Nonlinear Decomposition: Five Pliés in Fifth Position with Port de Bras.** A 3-D view of a two dimensional projection of the latent variable space. The 2-D view is shown in Fig. B.2. In fifth position, the dancer stands with the backs of their heels pressed against the metatarsal of the opposite foot, making a zig-zag between the two feet for dancers without perfect turnout.

Figures 6.5 and 6.6 represent the GPDMs generated from the Ballet 2 and Choreographed sequences. In order to compare the sequences, which were of different length, each sequence has been downsampled by the same amount (every 10th frame is deleted). Alternatively, a shorter segment of the Choreographed sequence can be compared with the full Ballet 2 phrase in Figs. 6.7 and 6.8.

Similarities exist between each case of comparisons between the ballet and modern dance sequences, but first, notice that the complexity of both latent space trajectories increases from Fig. 6.5 to Fig. 6.7 and Fig. 6.6 to Fig. 6.8, respectively. Additionally, the precision of each model also increases. The raises the question of how much data is appropriate for the GPDM algorithm. Is the full representation of the data necessary to capture a meaningful structure? It is possible that the downsampled data is an oversimplification. Or, did the downsampled versions of the clips prevent the algorithm from over-fitting? In other words, perhaps the downsampled data enforces enough uncertainty in the model (which is only a model after all).

The second observation of note is that in both cases, the structure of the ballet sequence is more simple, and even, in the downsampled case, linear along sections of the trajectory. On the other hand, the model of the modern choreography weaves in a tighter, more curvilinear fashion that creates rippling ridges along its trajectory. Discussion of these results is saved until the next chapter.

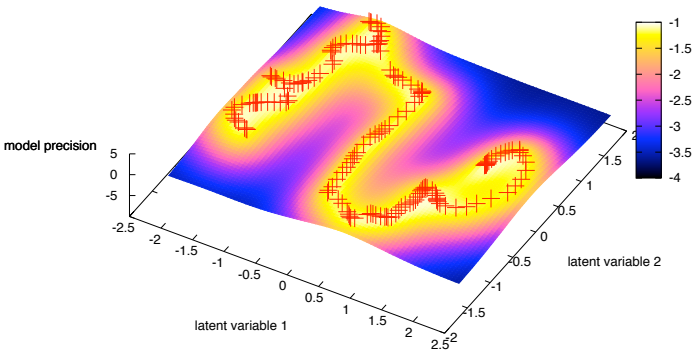


Figure 6.5: **Results of Nonlinear Decomposition: Classical Ballet Sequence (downsampled) - Grand Allegro.** The 2-D view is shown in Fig. B.5. A grand allegro is a type of ballet sequence characterized by big, presentational movements and often features soaring jumps. This sequence has been downsampled by removing every tenth frame.

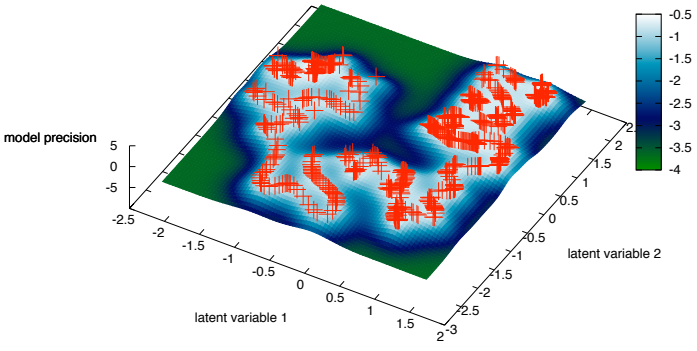


Figure 6.6: **Results of Nonlinear Decomposition: Modern Dance Choreography (downsampled).** The 2-D view is shown in Fig. B.6. The choreographed sequence was derived from the improvisation also collected in the data set.

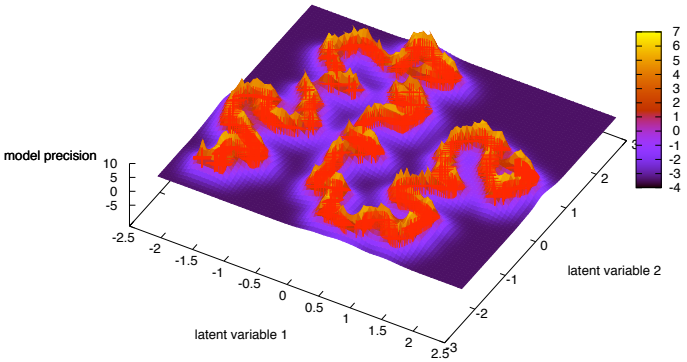


Figure 6.7: **Results of Nonlinear Decomposition: Classical Ballet Sequence - Grand Allegro.** The 2-D view is shown in Fig. B.7.

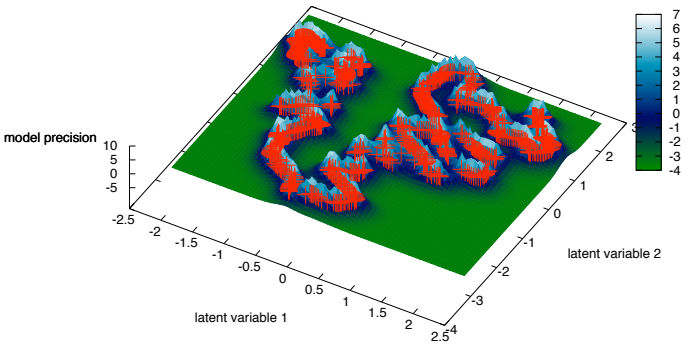


Figure 6.8: **Results of Nonlinear Decomposition: Modern Dance Choreography (1st section).** The 2-D view is shown in Fig. B.6. The first part of the choreographed sequence parsed from the full clip in order to make it shorter for comparison to the grand allegro.

Chapter 7

Conclusions and Future Work

The results of the linear analysis indicate (1) that walking is a fairly simple movement that can be reasonably decomposed with linear methods (2) and that this method quickly looses its ability to provide meaningful results with increasingly complex results. Conclusion (1) is supported by the straightforward analysis of the Walking sequence and was expected based on Olson's previous, unpublished work. Conclusion (2) is supported in that each component found here has a tenuous connection to natural movement, especially for the only slightly more complex case of the Arabesque sequence. Although testing this method on a longer data set would be interesting, these results indicate that PCA has limited use in trying to find a simpler representation, much less a basis of postures (like in [26]), for complex human motion with a result that can be interpreted by our bodies.

The results of the nonlinear analysis for the cases of Figs. 6.5 to 6.8 raise an intersting discussion. Comparisons like this are at the heart of what this thesis aims to accomplish, and it is one of the most interesting applications presented because it moves beyond simple movements whose mechanics can be examined through existing tools. Here, the structure of choreographed movement is finally explored. Remember that there are known fundamental differences between classical ballet and modern dance. As discussed in Sec. 5.2,

Experimental Design, the guiding principles of the two choreographic styles have different origins. It is often instead said that the aesthetics of the two styles of dance are different, and indeed, a result of these different choreographic and ideologic starting points, the two styles *look* very different.

This difference was noted in one of the papers that helped shape the very beginnings of this thesis. Elizabeth Bradley and Josh Stuart's paper *Using Chaos to Generate Variations on Movement Sequences* involves a scheme for interpolating between human poses in a chaotically generated reordering of the poses. The success of their interpolation schemes[1] was highly dependent on the style of the sequence, and they make the following point in a footnote:

> In ballet, body parts tend to describe piecewise-linear paths through space, emphasizing the positions at the junctions of those linear segments; in modern dance, on the other hand, arcing motions are much more common. [2]

While there are many aspects about the GPDM algorithm that are yet to be understood in this context, particularly the significant changes that occur when the data is downsampled, it is proposed that the truth of this quote, long observed qualitatively by dance scholars and critics, may be evident in the preliminary results of this thesis.

From its outset this thesis has stated two main goals which were developed independently throughout the year: (1) understand the methods for dimensionality reduction employed in two dimensions for C. Elegans at Princeton in order to extend these tools to ones appropriate for human movement (2) and establish and employ a sufficient method for representing and capturing the more complicated 3-D postures that make up human movement and the subtleties of posture changes that dance so often employs. Each goal has been met to the extent that the two have been successfully united in this thesis.

However, each component draws from a realm of research that is rich in complexity and

[1] And, furthermore, the idea that if a complete basis or description of movement could be achieved, the success of this scheme would be even greater, was a, if not the, motivating factor for the final proposal of this thesis.

deserves its own in depth study, leaving some obvious shortcomings and wide margins for improvement. The preliminary results presented will require an iterative process in order to fully understand. That is, these findings should mold the next round of analysis from this data set and adjust the assumptions applied in the dimensionality reduction techniques, and furthermore, inform choices about new sequences to study and new algorithms to try. This process could provide more appropriate analysis that is specific for the underlying structure of this particular data set and, eventually, the larger breadth of human movement.

Beyond this project, the scope of future work is rich and diverse. Understanding quantitatively how to recognize human movement, opens many doors in the field of robotics and human computer interaction. It is only with great understanding of how we work that our technology can best serve us. The ever-popular iPhone employs the smallest action of human motion: the flick of our little finger; yet, with it, engineers at Apple, Inc. have created one of the most intuitive and efficient user interfaces available in today's technology. What then could a cell phone or computer that recognized the motion of an entire arm, face, or body do for improving this interaction? In addition to teaching computers how to understand us, learning about the natural motions of our own bodies would provide medical practitioners with invaluable data and breakthrough methods that allow them to better asses and evaluate patients' progress and diagnosis.

The results of Raquel Urtasun and her collaborators move forward in both veins of future work discussed above and provide additional demonstration of the same GPDM algorithm used in this thesis. In their paper *3D People Tracking with Gaussian Process Dynamical Models* [29], Urtasun and her collaborators demonstrate the predicitve nature of these models and provide analysis that lends itself directly to movement recognition and biomechanical analysis. The significant stylistic variation their results hold for is another step in the direction of having an interactive computer interface based on the computer's recognition of the user (and, indeed, regonition of several users) is achieved. Likewise, taking the results presented in Figs. 6.2 and 6.1 a few steps farther, these researchers

identify abnormal walking patterns using a GPDM, thus demonstrating the use of these models as a diagnostic tool in the field of biomechanics.

Finally, the clarity with which a more explicit knowledge base of how a dancer's body moves and how a choreographer's creativity is implemented would motivate this performing art to a higher level. It would be handing these artists the paintbrush of mathematics - the same tools employed here in this thesis by engineers and scientists like Neil Lawrence. And that's not just a pretty metaphor. It represents a real, implementable system that needs to be engineered, needs to be designed, in order to fit the constraints and answer the questions posed here. These design solutions should not solve a simple data processing task, but need to solve a problem with facets in a physical world and in a behavioral context and with an artistic application - of humans dancing.

Appendix A

MATLAB Code

This appendix presents minor MATLAB scripts generated by the student for various reasons.

```
{
%%%%%%%%%%%%%%%%%%%%% gaussaian.m %%%%%%%%%%%%%%%%%%%%%
%%% generates a Gaussian distribution given mu and sigma^2
%%% Amy LaViers %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%choose mean and variance
sig=.3; %variance=std^2 (sigma^2)
mu=0; %mean

%establish constants for gaussian function
a=1/sig/sqrt(2*pi);
b=mu;
c=sqrt(sig);

%define function
x=[-5:.1:5];
f=a*exp(-((x-b).^2)/(2*c*c));

%plot results
plot(x,f)
title('Plot of Various Gaussian Functions')
hold on
```

```
%% define and plot another

%choose mean and variance
sig=3; %variance=std^2 (sigma^2)
mu=1.5; %mean

%establish constants for gaussian function
a=1/sig/sqrt(2*pi);
b=mu;
c=sqrt(sig);

%define function
x=[-5:.1:5];
f=a*exp(-((x-b).^2)/(2*c*c));

%plot results
plot(x,f,'r')

%% define and plot another

%choose mean and variance
sig=1; %variance=std^2 (sigma^2)
mu=-2; %mean

%establish constants for gaussian function
a=1/sig/sqrt(2*pi);
b=mu;
c=sqrt(sig);

%define function
x=[-5:.1:5];
f=a*exp(-((x-b).^2)/(2*c*c));

%plot results
plot(x,f,'k')

%% define and plot another

%choose mean and variance
sig=.5; %variance=std^2 (sigma^2)
mu=0; %mean
```

```
%establish constants for gaussian function
a=1/sig/sqrt(2*pi);
b=mu;
c=sqrt(sig);

%define function
x=[-5:.1:5];
f=a*exp(-((x-b).^2)/(2*c*c));

%plot results
plot(x,f,'g')
legend('\sigma^2=0.3, \mu=0.0','\sigma^2=3.0, \mu=1.5','\sigma^2=1.0, \mu=-2.0','\sigma^
}



{
%%%%%%%%%%%%%%%% pca.m %%%%%%%%%%%%%%%%
%%% performs principle component analysis
%%% for demonstration of concept %%%%%%%%
%%% Amy LaViers %%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all

%%initialize some data
    % %random
    % dataX_orig=rand(10,1);
    % dataY_orig=rand(10,1);
    % %correlated
    % dataX_orig=[2.8 7 1 4 8 6 9 3 4.4 3.6]; %hours studied for exam
    % dataY_orig=[52 88 50 71 80 68 90 61 73 57]; %grade made on exam
    %     %plot(dataX_orig,dataY_orig,'o'); hold on

%smith
dataX_orig=[2.5 0.5 2.2 1.9 3.1 2.3 2 1 1.5 1.1];
dataY_orig=[2.4 0.7 2.9 2.2 3.0 2.7 1.6 1.1 1.6 0.9];

%%subtract mean (centers data around zero)
dataX=dataX_orig-mean(dataX_orig);
dataY=dataY_orig-mean(dataY_orig);
data=[dataX;dataY];
```

```
data_mean=[mean(dataX_orig);mean(dataY_orig)];
%view
figure(1)
%subplot(2,2,1)
plot(dataX,dataY,'+')
%title('Mean-Subtracted Data')
%xlabel('(a)')
V=axis;

%% compute covariance  Thanks Matt Tilghman
C=cov(dataX,dataY);
[eVectors, eValues]=eig(C);
e2=eVectors(:,1)';
e1=eVectors(:,2)';%higher eigenvalue

scale = 2*max(max(dataX),max(dataY));

figure(2)
%subplot(2,2,2)
plot(dataX,dataY,'+')
%title('Mean-Subtracted Data with Component Eigenvectors')
%xlabel('(b)')
hold on

x1=scale*e1(1)*[-1 1];
y1=scale*e1(2)*[-1 1];
plot(x1,y1,'r') %higher eigenvalue

x2=scale*e2(1)*[-1 1];
y2=scale*e2(2)*[-1 1];
plot(x2,y2,'k')

axis(V)

%% transform the data (ie. express it in terms of new coordinates)
dataTrans=eVectors'*data;

figure(3)
%subplot(2,2,3)
plot(dataTrans(1,:),dataTrans(2,:),'+')
%title('Transformed Data')
%xlabel('(c)')
```

```
axis(V)

%% reduce data with 'principle component' or higher eigenvector
dataRed=eVectors(:,2)'*dataTrans;

%% replot simplified data
dataNew=eVectors(:,2)*dataRed;

figure(4)
%subplot(2,2,4)
plot(dataNew(1,:),dataNew(2,:),'+')
%title('Reduced Data')
%xlabel('(d)')

axis(V)
}
```

Appendix B

Additional Results

The figures in this appendix are meant to supplement and futher inspire the reader. Related figures are grouped on the same page. Enjoy!

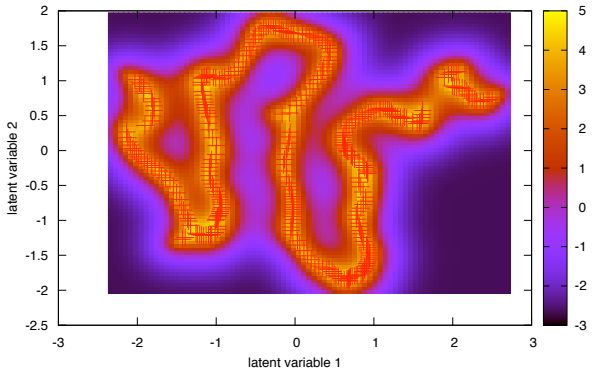


Figure B.1: **Alternate Results of Nonlinear Decomposition: Five Demi Pliés in First Position.** A two dimensional projection of the latent variable space. In first position, the dancer stands with the backs of their heels pressed together while the toes point outward away from the body.

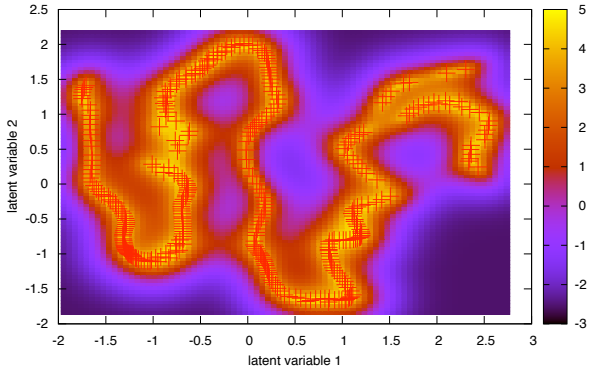


Figure B.2: **Alternate Results of Nonlinear Decomposition: Five Demi Pliés in Fifth Position.** A two dimensional projection of the latent variable space. In fifth position, the dancer stands with the backs of their heels pressed against the metatarsal of the opposite foot, making a zig-zag between the two feet for dancers without perfect turnout.

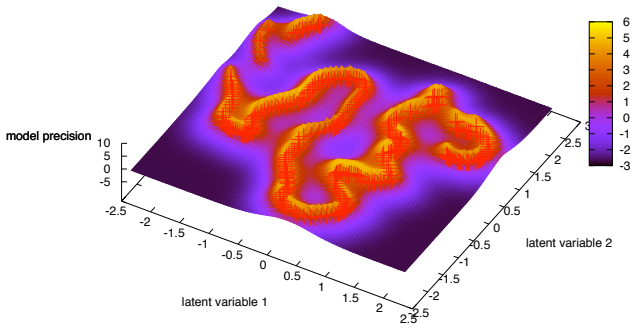


Figure B.3: **Additional Results of Nonlinear Decomposition: Five Demi Pliés in Fourth Position.** A 3-D view of a two dimensional projection of the latent variable space. Fourth position is similar to fifth position, except the dancer's front foot is offset about a foot away from the back foot.

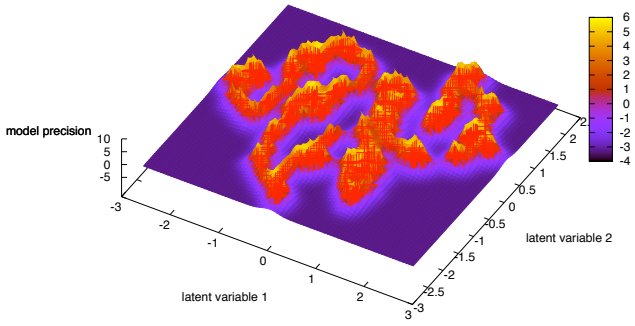


Figure B.4: **Results of Nonlinear Decomposition: Six Pirouettes.** A three dimensional view of a two dimensional projection of the latent variable space. The sequence referred to as Ballet 4 in Sec. 5.1, was parsed to contain the series of six double pirouettes. Each pirouette is preceded by a preparation from fourth position and followed by a landing. The sequence was downsampled by removing every tenth frame.

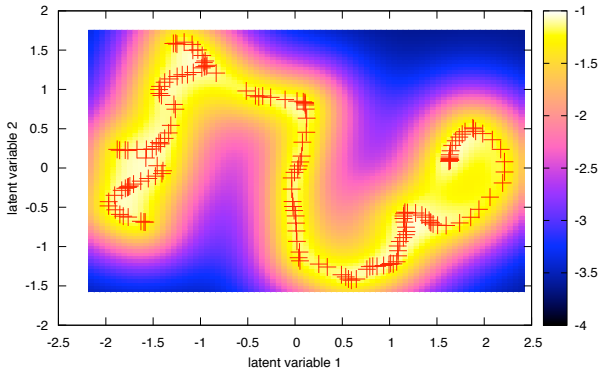


Figure B.5: **Alternate Results of Nonlinear Decomposition: Classical Ballet Sequence (downsampled) - Grand Allegro.** A two dimensional projection of the latent variable space. A grand allegro is a type of ballet sequence characterized by big, presentational movements and often features soaring jumps. This sequence has been downsampled by removing every tenth frame.

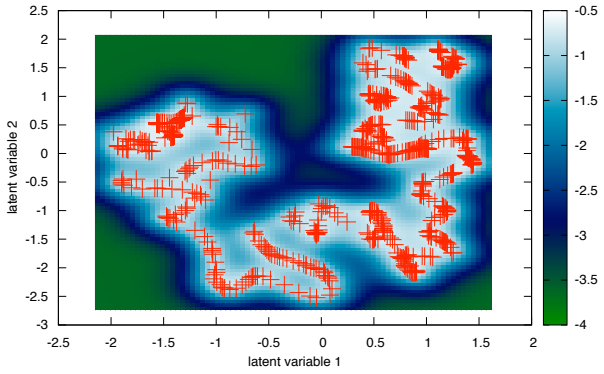


Figure B.6: **Alternate Results of Nonlinear Decomposition: Modern Dance Choreography (downsampled).** The choreographed sequence was derived from the improvisation also collected in the data set.

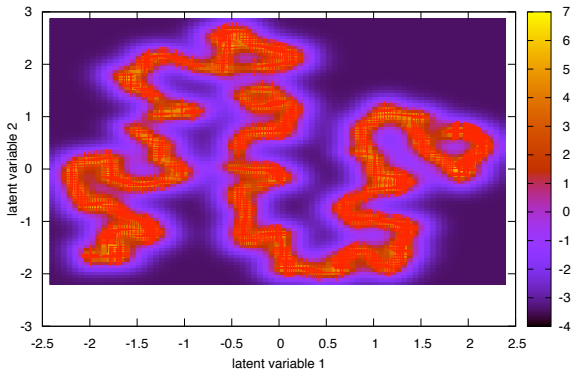


Figure B.7: **Alternate Results of Nonlinear Decomposition: Classical Ballet Sequence - Grand Allegro.** A two dimensional projection of the latent variable space. A grand allegro is a type of ballet sequence characterized by big, presentational movements and often features soaring jumps.

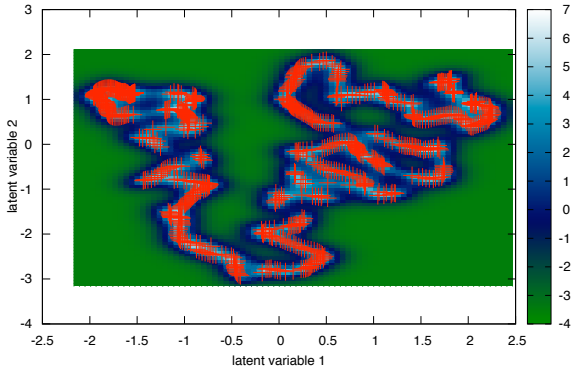


Figure B.8: **Alternate Results of Nonlinear Decomposition: Modern Dance Choreography (1st section).** The first part of the choreographed sequence parsed from the full clip in order to make it shorter for comparison to the grand allegro.

References

[1] I. Bartenieff and D. Lewis. *Body movement: Coping with the environment*. Gordon & Breach Science Pub, —1980—.

[2] E. Bradley and J. Stuart. Using chaos to generate variations on movement sequences. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 8:800, —1998—.

[3] M. Brand. Nonlinear dimensionality reduction by charting. *Mitsubishi Electric Research Laboratories*, —2004—.

[4] C. Bregler. Learning and recognizing human dynamics in video sequences. pages 568–574, —1997—.

[5] D. Chigirev and W. Bialek. Optimal manifold representation of data: An information theoretic approach. In *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*. Bradford Book, —2004—.

[6] R. Copeland. *Merce Cunningham: the modernizing of modern dance*. Routledge, —2004—.

[7] D. Del Vecchio, R. M. Murray, and P. Perona. Decomposition of human motion into dynamics-based primitives with application to drawing tasks. *Automatica*, 39(12):2085–2098, —2003—.

[8] W. Forsythe. *Improvisational Technologies: A Tool for the Analytical Dance Eye*. ZKM/Zentrum fur Kunst und Medientechnologie Karlsruhe, Karlsruhe, Germany, 2nd edition, —1999—.

[9] P. M. Forsythe W. and Z. S. N. Synchronous objects. *Ohio State University*, —2009—.

[10] M. S. A. Graziano and T. N. Aflalo. Mapping behavioral repertoire onto the cortex. *Neuron*, 56(2):239–251, —2007—.

[11] A. Hutchinson and D. Anderson. *Labanotation*. New Directions, New York, —1954—.

[12] Inition. Vicon peak - vicon mx. *www.inition.co.uk*, —2008—.

[13] J. Lander. Working with motion capture file formats. *Game Developer*, —1998—.

[14] N. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*, page 329. Bradford Book, —2004—.

[15] N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816, —2005—.

[16] N. Lawrence and J. Quinonero-Candela. Local distance preservation in the gp-lvm through back constraints. pages 513–520. ACM New York, NY, USA, —2006—.

[17] N. D. Lawrence. Learning for larger datasets with the gaussian process latent variable model. volume 11, —2007—.

[18] D. J. C. MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, —2003—.

[19] V. Maletic. *Body, space, expression*. Walter de Gruyter & Co., Berlin, —1987—.

[20] F. Perez-Cruz. Personal communication. *Princeton University*, —2009—.

[21] S. Rajko and G. Qian. Real-time automatic kinematic model building for optical motion capture using a markov random field. *LECTURE NOTES IN COMPUTER SCIENCE*, 4796:69, —2007—.

[22] G. Rizzolatti and L. Craighero. The mirror-neuron system. *Annual Review of Neuroscience*, —2004—.

[23] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, —2000—.

[24] D. Smit. Pittsburg innovates. *Pop City Media, Model D LLC*, —2005—.

[25] L. I. Smith. A tutorial on principal components analysis. *Cornell University, USA*, 51:52, —2002—.

[26] B. W. R. W. Stephens G., Johnson-Kerner B. Dimensionality and dynamics in the behavior of c. elegans. *PLoS Comp Bio*, —2007—.

[27] J. B. Tenenbaum, V. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, —2000—.

[28] M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, pages 611–622, —1999—.

[29] R. Urtasun, D. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. page 238245, —2006—.

[30] J. Wang. Personal communication. *University of Toronto*, —2009—.

[31] J. Wang, D. Fleet, and A. Hertzmann. Gaussian process dynamical models. *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, 18:1441, —2006—.

[32] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, —2008—.