Proof of Concept for the use of Motion Capture Technology in Athletic Pedagogy

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Derrick T. Danyluk

Permission to Use

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

Visualization has long been an important method for conveying complex information. Where information transfer using written and spoken means might amount to 200-250 words per minute, visual media can often convey information at many times this rate. This makes visualization a potentially important tool for education. Athletic instruction, particularly, can involve communication about complex human movement that is not easily conveyed with written or spoken descriptions. Video based instruction can be problematic since video data can contain too much information, thereby making it more difficult for a student to absorb what is cognitively necessary. The lesson is to present the learner what is needed and not more. We present a novel use of motion capture animation as an educational tool for teaching athletic movements. The advantage of motion capture is its ability to accurately represent real human motion in a minimalist context which removes extraneous information normally found in video. Motion capture animation only displays motion information, not additional information regarding the motion context. Producing an automated coach would be too large and difficult a problem to solve within the scope of a Master's thesis but we can perform initial steps including producing a useful software tool which performs data analysis on two motion datasets. We believe such a tool would be beneficial to a human coach as an analysis tool and the work would provide some useful understanding of next important steps towards perhaps someday producing an automated coach.

Acknowledgements

I would like to acknowledge several people whose support and guidance made this work possible.

First, I would like to thank my supervisor, Dr. Eric Neufeld. This work would never have been completed were it not for his patience, guidance and encouragement. As challenging as this process has been at times, I can honestly claim to be better for it. Dr. Neufeld is a major reason for that simply by refusing to accept anything less than the best work possible from me.

I would also like to thank the members of my graduate committee, Dr. Gord McCalla, Dr. Julita Vasileva and Dr. Joel Lanovaz for their input and comments.

I would like to thank my family for their incredible patience and understanding during all the times I had to be absent while writing, working on software, reading or any of the numerous other things that were necessary for the completion of this work.

To all of you, I offer my most sincere thanks and appreciation. I will never forget what you have given me.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

This work began as an effort to apply computer graphics in the context of athletic instruction. This is a broad topic, and could include anything from simple diagrams to photorealistic rendering of sports events. It could also include some ideas from smart graphics, incorporating expert system type constructs to understand and explain motion and, furthermore, give advice to a novice how to improve the motion. There was also a pragmatic wish to provide a ”proof of concept” that the idea was within the state of the art of current hardware and software - that is, that there was reason to believe that an intelligent analysis of motion could be done in real time, and that the results could be conveyed clearly.

Eventually, we defined the thesis project as follows. A software system would be built that would process motion capture data recorded from a learner by comparing the motion with motion capture information of an expert, and then showing the novice where the novice's motion differed from that of the expert. This idea in itself proved to be considerable. Commercial products like Microsoft's Kinect, Sony's Move and Nintendo's Wii obviously required enormous investments of resources to produce and hit the market while this work was being done. As a result, certain aspects of the problem had to be simplified. As well, the solution drew from several areas of computer science and mathematics, including graphics, visualization, intelligent tutoring and functional analysis[128]. Computer graphics and visualization are related but separate disciplines. Computer graphics encompasses all forms of production of graphical media using a computer. Visualization in the context of computer science is specific to producing graphics that provide additional understanding to the viewer of an underlying set of data. The area of intelligent tutoring studies types of systems that provide direct customized instruction or feedback to students without the intervention of human beings, while performing a task[126]. Functional analysis is a particular analysis approach which involves breaking down a problem into it's functional components. These components can be further broken down until the necessary level of granularity is achieved.

It is difficult to concisely define visualization since almost any visual entity could be considered a visualization of some physical or conceptual object. For example, a map visually represents a subset of the surface of our physical planet. An abstract drawing consisting of particular shapes and colours could be said to visualize emotion. Typography, such as the words on this page, can be said to be a visualization of language. Bath[12] describes debate within the area of typography regarding the use of Roman-style font versus Gothic script as "a movement away from the difficult to read Gothic typeface first employed by Gutenberg and towards clarity and legibility" thus casting typography not just as a matter of aesthetics, but as a matter of perception, by facilitating the process of reading. Amateurs experimenting with visualization have been known to create visualizations of data that are visually interesting, but do not convey meaning very well since visual media can represent almost anything from a physical object to a concept to a process. It is instructive to look at some classic ideas from visualization that have become so commonplace, we have forgotten that they are constructs. The first scientific drawings of Copernicus (figure 1.1) used diagrams to communicate concepts that were difficult to describe outside of formal mathematical specification. Copernicus's drawings attempted to visualize that which could not be seen by humans of that time. Maps now are in everyday use. Some planets can be viewed by a person in their backyard with an inexpensive telescope. Statisticians like William Playfair (figure 1.2) used diagrams as a means of communicating abstract concepts such as statistical summarization of data. Tables of numbers, however organized, can be cumbersome for humans to fully absorb. Playfair's diagrams provided a way of visually summarizing data to, in turn, make it's meaning more readily comprehensible.

One issue with visualizations at this time was the cost of producing them both in terms of basic resources and the time and energy expended by whomever would do so. Modern computer graphic technology has provided the latest evolutionary steps with real-time animation and three dimensional graphics. Additionally, the technology has become ubiquitous such that the cost of producing certain types of visualizations has become negligible apart from labour. These technologies can be especially effective with complex multidimensional data. As data processing has evolved from the manual calculations of Playfair's day to modern data analysis functions built into the common spreadsheet, we

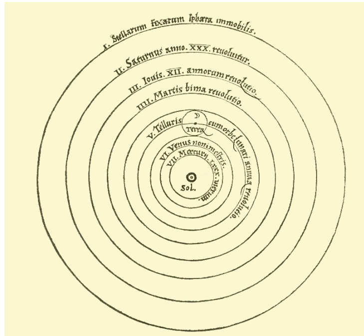

Figure 1.1: Diagram of Copernicus' heliocentric model of the solar system. (Image courtesy of Wikimedia Commons)

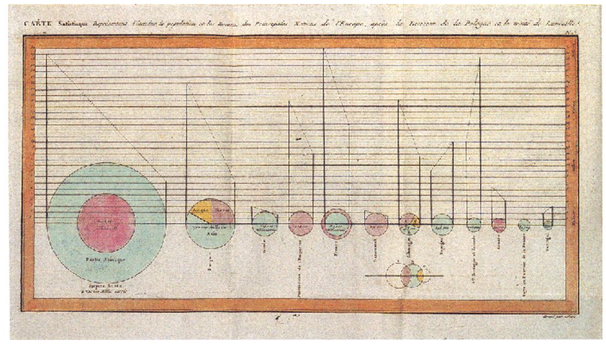

Figure 1.2: Examples of the first pie and bar charts by William Playfair. (Image courtesy of Wikimedia Commons)

now have arguably greater means for producing more complex data. This increase in data complexity in turn drives the need for further advances in visualization techniques.

Modern data visualization takes into account the physiological and neurological bases of how we perceive things. Ware discusses the concepts of Geon theory and diagrams and how we perceive meaning in even very simple object primitives. Ware compared Geon diagrams to traditional Unified Modelling Language(UML) diagrams[159]. UML diagrams consisted of simple primitives like lines and boxes. Geon Diagrams consisted of stylized 3D primitives. A comparison of the two appears later in figure 2.3. A theme in modern visualization is that some people in the field (i.e. Tufte) prefer minimalist visualizations and scorn adornments. Ware's experiments suggested that stylizing a diagram could make it more memorable. Geon theory originates with Hummel and Biederman[73]. The authors describes a hierarchical processing pattern whereby perception of objects begins at a very high level and progresses through increasing levels of detail. According to this theory, for example, a human being would first be perceived by their overall shape followed by vertices and component axes and then three-dimensional primitives such as cylinders and spheres. Details such as color and texture are secondary albeit still important characteristics. Later we will see more examples of this. Bateman et al.[11] discuss whether ornaments assist in understanding or are just *chartjunk* in statistical graphics. More informally, McCloud notes a range of levels of details in comic books, from stick figures to highly detailed ones.

Neufeld et al. [114] explored the use of additional visual attributes with graphical models for the purpose of visualizing multivariate data. That work explored visualizations of directed networks of random variables. Such graphs are usually visualized as simple normal curves. However, colours and images, along with text, can be used to make it easier to distinguish variables from one another. The authors have found that animated drawings, where nodes physically changed in response to interventions on other nodes , to lose context information, and perhaps to be overkill. The authors assert that by careful selection of specific visual attributes to be used with such models, 'there is a great potential for quickly conveying statistical relationships among variables to nonexperts.'

One particular animation technology has gained prominence through its use in the entertainment industry. Motion capture animation involves the recording of real human motion using sensors and animating the results (figure 1.3). This provides a visualization of real human movement which can be used creatively or to facilitate the study of human motion as in Kinesiology. Annotations can be added to these animations to convey more descriptive information to a viewer.

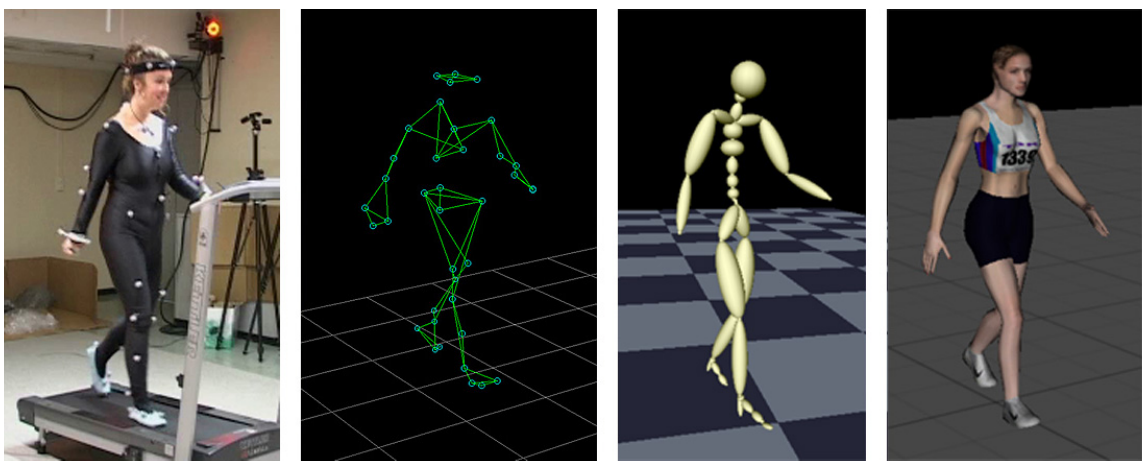

Figure 1.3: An actor wearing a motion capture suit on the left and the corresponding animations in the following three images.(Image used by permission from École Polytechnique Fédérale de Lausanne

One advantage of motion capture animation used for pedagogy is that it provides a way of removing extraneous information. When attempting to study a motion by watching a real life video, it is necessary to ignore the large amount of additional information the video contains such as background environment, specific characteristics of actors, etc. Motion capture animation shows a blank background with a simple visualization representing the human form. This simplified view allows the viewer to focus on the remaining information, the actual motion itself, without distraction. Figure 1.4 shows the difference in information between a motion capture animation and a photograph of the actor the data is derived from.

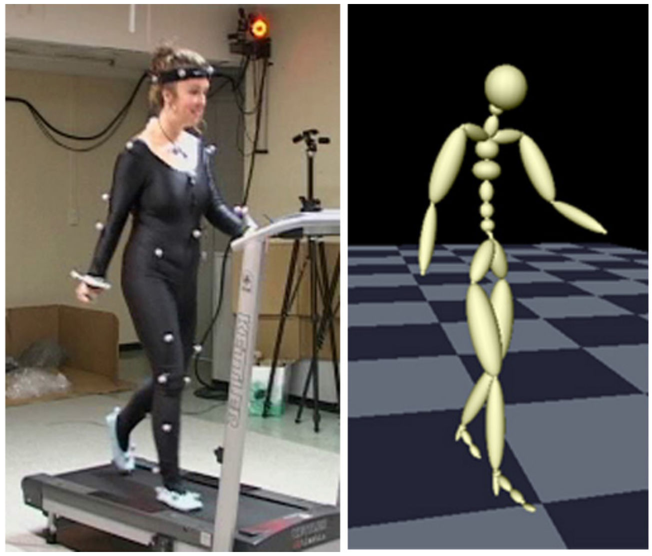

Figure 1.4: An actor wearing a motion capture suit on the left and the corresponding animation on the right.(Image used by permission from École Polytechnique Fédérale de Lausanne)

An initial version of our system used a figure which was essentially a stick figure. In the course of our work, we also observed the tradeoff between minimalism and memorability playing off when it came to designing our avatars. After allowing various individuals informally view the visualization, it became apparent that a stick figure might be too simplistic. In almost every case, viewers required some prompting to fully understand the figure and its motion. By making the figure appear more like a mannequin, we were able to add enough detail to provide greater clarity without adding back extraneous detail that might be distracting. We should note that we did not do a formal user study by rather inferred the additional effectiveness of the new figure anecdotally.

The areas of visualization and motion capture animation were linked together and combined with graphical annotation by Bouvier-Zappa, et al[21]. The goal of their study was to produce an interactive system to synthesize a 2D image of an animated character by generating motion cues derived from three-dimensional skeletal motion capture data (figure 1.5). The authors describe a method for adding visualization techniques to motion capture animations of human figures to better convey the mechanics of the animated motion to the viewer. The most prominent visualization object used was the arrow to denote direction and magnitude of movement. The authors discuss the notion of using their work as a pedagogical tool with some further work to solve remaining problems, notably, the ability to make a comparison of two motions and visualize the difference between them.

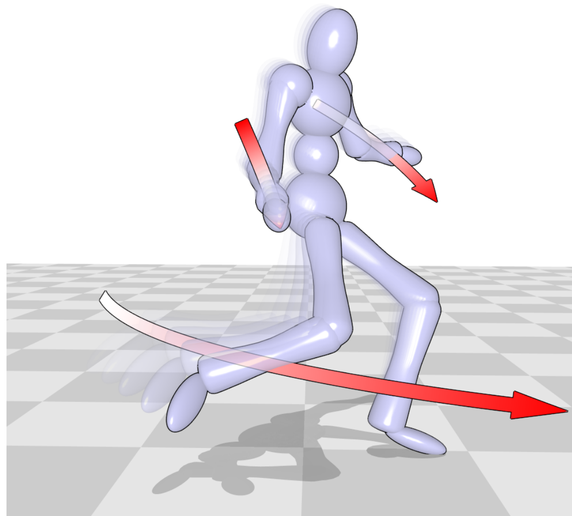

Figure 1.5: A 2D image of an actor executing a soccer kick with arrows added.(Image used by permission from Simon Bouvier-Zappa)

1.1 Overview

Chapter two surveys the current and recent work on motion capture animation and analysis and describes the state of the art for consumer graphics products. We also describe relevant research from the area of cognitive science as it pertains to our pedagogical application. This will establish the context of our contribution to this area of using analysis of motion capture data for educational purposes.

Chapter three specifies details of our stated problem along with a more detailed explanation of some of the work we are using as a basis.

Chapter four discusses our proposed method of representing motion capture data which will facilitate analysis and visualization. This section will include a detailed description of all processing steps necessary. Chapter four will continue with a discussion of our particular method of visualizing this data. The final sections of chapter four will describe our two primary outcomes: motion capture data comparison and result visualization.

Chapter five demonstrates the results of our software system showing examples of our visualizations with accompanying description.

Chapter six concludes the thesis with a final discussion of the results and future work which could build from our basis.

Chapter 2

Survey

2.1 Visualization and Cognition

Written and spoken language convey information at approximately 200-250 words per minute[62]. Tufte[150] notes that visual media can convey much larger volumes of information in as much time. This provides a notable cognitive advantage to visual media when used for pedagogical purposes.

Tufte's first book, *The Visual Display of Quantitative Information*, is, in a certain sense, his most concise. Printed mostly in black and white, it discusses achieving clarity and simplicity in displaying multivariate information, where variables range over space, time, and numerous other quantities. The first part of the book discusses mainly graphs, including time series, and deriving principles of graphical excellence. He speaks against visual deception, but also against *chart junk* (figure 2.1), extraneous embellishments that convey no data, from 3D bar graphs to editorial cartoons. Other authors such as Gutwin, et al.[11] have countered that there is value in some amount of embellishment which may serve to make a graphic more engaging or memorable without directly conveying data (figure 2.2).

Envisioning Information begins by asking how we can escape *flatland*, the two-dimensional world of paper, when depicting the complex and multidimensional world we live in. While staying true to principles of simplicity and integrity, this second book raises subtle issues some might call aesthetics. For instance, the Vietnam Veteran's Memorial has the names of all soldiers killed in that war, and Tufte reports on the decision to list the names of the veterans in chronological order of their deaths. An alphabetical ordering would have made the wall resemble a phone book, with 16 people in a row, named James Jones. Chronological ordering enhanced the spirit of the individual. The idea is uplifting, but it is hard to infer a widely applicable principle from it. Part of the challenge of this work was finding

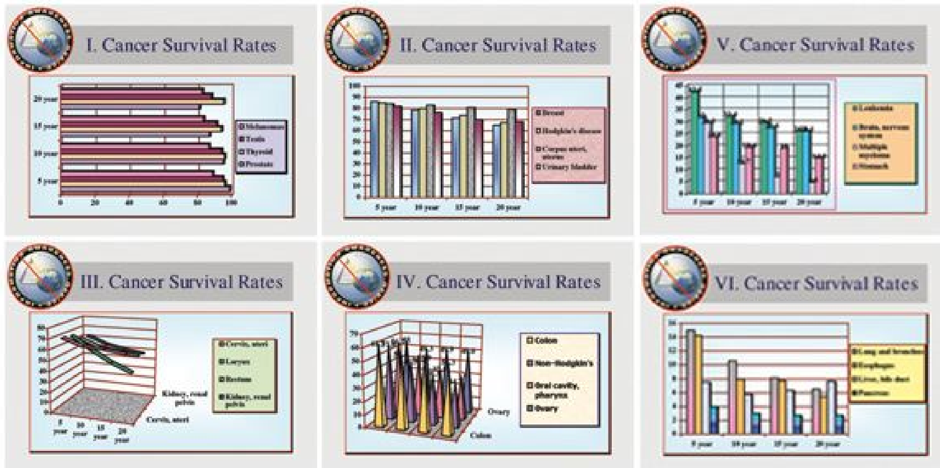

Figure 2.1: An example of statistical graphics employing what Tufte refers to as 'Chart Junk'.(Image used by permission from Graphics Press)

Figure 2.2: An example of a statistical graphic with embellishment.(Image used by permission from Dr. Carl Gutwin)

the right level of visual information.

Irani and Ware[75] asserted that the use of 3D primitives in *geon diagrams* provided a more effective paradigm. The authors compared two-dimensional Unified Modeling Language(UML) diagrams to geon diagrams and conducted experiments to determine the comparative effectiveness of the geon diagrams for user recognition of sub-parts represented in the diagrams. The authors noted recognition times and error rates approaching half of those measured for the UML diagrams. This supported the hypothesis that geon diagrams are easier and faster to interpret than 2D UML diagrams. The Geon-UML comparison shows that there may be additional value in adding a small amount of additional visual information beyond just that which explicitly transmits data. Diagrams such as these may also facilitate transmitting information that is only implied in the standard UML diagram such as the organizational hierachy of a particular model. With a Geon diagram, it is explicitly visible.

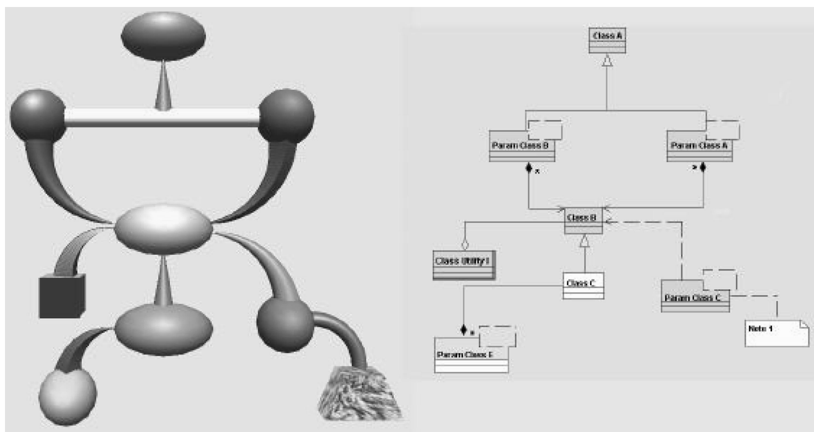

Figure 2.3: An example of a Geon diagram with its corresponding UML diagram.(Image used by permission from Dr. Pourang Irani)

Two of Tufte's later books explore diverse areas of the human enterprise, including art. *Visual Explanations* and *Cognitive Style of Powerpoint* both study the Challenger disaster and the response of the US bureaucracy to it. In *Cognitive Style of Powerpoint*, Tufte notes the importance of an appropriate amount of detail as demanded by context. In the Challenger case, much detail was required but little was distributed as seen by the use of Powerpoint to communicate information of a detailed nature. Tufte's last book, *Beautiful Evidence*, devotes two pages to dance notation, presenting an elaborate set of engravings of a contredanse showing four couples moving in three dimensions showing floor plans and music. This complex diagram, shown in figure 2.4, tells a lot on two paper dimensions.

Figure 2.4: An engraving attempting to combine footstep diagrams with additional visual information.(Image used by permission from Graphics Press)

There are several visualizations for dancing. The most familiar is the footstep diagram (figure 2.5). It provides only information about the movement of the feet. This is an important part of dancing, but the entire rest of the body is left unrepresented. Nonetheless, footstep diagrams are common and popular. The following figures show two types of diagrams used for dance instruction.

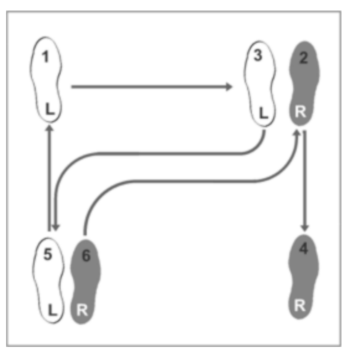

Figure 2.5: An example of a footstep diagram for dance instruction.(Image used by permission from Dancing4Beginners.com)

Figure 2.7 shows Benesh dance notation which appears to be the standard for dance notation. However, the notation poses a considerable barrier for a novice. Figure 2.6 suggests that the relation between Benesh notation and the actual positions is not intuitive. This degree of complexity, if necessary to learn how to dance, would provide a formidable obstacle to an amateur learner.

Figure 2.5 provides only a little information to its user. Figures 2.6 and 2.7 require

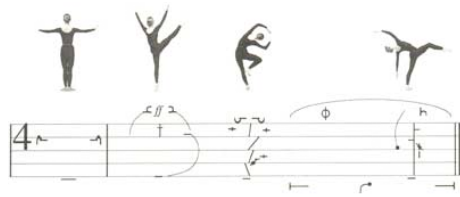

Figure 2.6: An example showing the correspondence of the notation to actual body movement and position.(Image used by permission from Royal Academy of Dance)

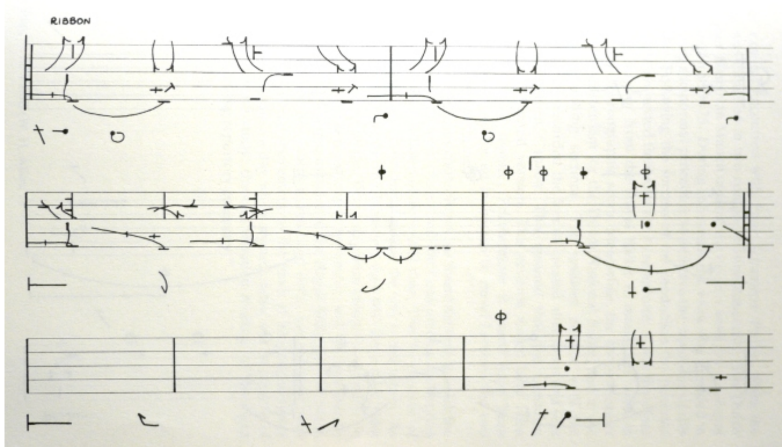

Figure 2.7: An example of Benesh dance notation.(Image used by permission from Oxford University Press)

considerable training to understand. The engraving in Tufte (below) shows a complex set of relationships between music, two views of the dance floor, and several couples. The engraving from Tufte represents another compromise of the kind we have been describing. The human figures assist the student as regards positions of all parts of the body, and the perspectives show the relationships of all the dance partners. The ubiquitous arrow implies the direction of motion within a frame, and the text presumably clarifies other details.

Similarly, many guitar players have difficulty reading formal music notation but, instead, work with tablature notation which provides a simple abstraction of a guitar fretboard, finger placements and even advanced techniques(figure 2.8). Once again, the two kinds of notation provide an interesting tradeoff. A guitar (or any stringed instrument) is different from a piano or horn instrument in that a note of exactly the same pitch may be found on many places on the instrument. For example, the open high 'E' string plays a note with a pitch frequency of 330 Hz. The same note may be found on the 5th fret of the 'B' string, the 9th fret of the 'G' string, and so on.

Figure 2.8: A comparison of formal music notation with tablature notation.

The traditional musical staff does not explicitly tell the guitarist which position to use when playing any note. This is important because chords of the same notes but at different positions have a different intonation. However, it gives the player very precise information about the pitches of notes, their duration, the beat (usually in the time signature 3/4, 4/4, etc). The tablature notation provides a visualization of the six guitar strings at each point in the composition by indicating which fret is to be pressed and which strings are to be sounded. However, the notation does not say much about beat and timing. The combined notation shown below provides both kinds of information to the student. Interestingly, the rise of online sharing of tablature (which is easy to produce using typewriter fonts) among amateur musicians raised a significant legal issue. Tablature became influential as a representation medium compared to formal musical notation because it made it possible for so many untrained musicians to share and discuss variations or differing interpretations or even different understandings of the same piece of music. This resulted in social networks for sharing tablature. Such was tablature's influence that it spawned lawsuits because the sheet music industry felt a threat to royalty revenues.

The engraving shown in figure 2.4, complex in its day, conveys a lot of detail. The animations we produce are comparable in that they concisely capture key aspects of motion, using critical points where sensors are places. They allow the user to view the motion from different angles. Apart from the graphical simplifications, they convey as much detail as possible about the individual motion.

In 2002, Gutwin[64] asserted the importance of *traces* added to pointers used in groupware applications. A trace is some form of added visual cue to show the immediate movement history of the telepointer in the application. Examples are shown in figure 2.9. The author augmented visual pointers with these traces to show the path the pointer had taken as it was used by a member of the group. After allowing participants to work with the groupware application using traces, the author interviewed the participants to obtain feedback on their effectiveness. The author noted that “overall, people felt that traces improved gestural communication”.

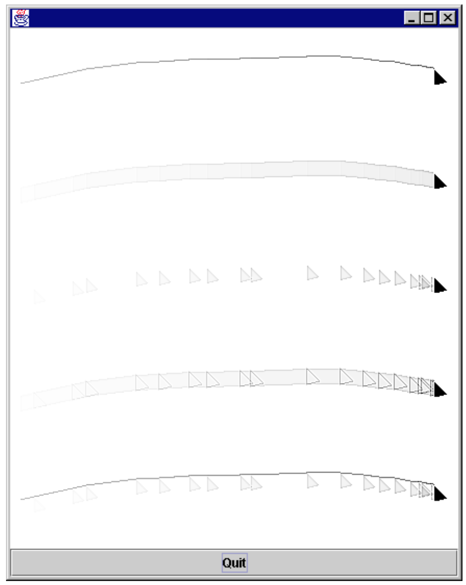

Figure 2.9: Different types of traces added to a pointer within an application.(Image used by permission from Dr. Carl Gutwin)

Another example of visualization used to convey information is in comic books. For example, a sequence of pawprints and overturned flowerpots might imply the path of a rambunctious puppy. Carmine Infantino’s depiction of “The Flash”, a character who can run at the speed of light used *speed lines*, which appear frequently. Scott McCloud discusses the use of visualization in comics as a means of communication. In his book *Understanding Comics*[106], McCloud discusses many examples of common comic iconography which are used to communicate concepts that cannot be easily represented in a two-dimensional medium such as time and motion.

Research in cognition has shown that visualization in the form of diagrams can aid a user in constructing mental models as well as using those models for comprehension. Novick et al.[119] showed that users were far more successful at performing assembly tasks using diagrams. The authors conducted an experiment whereby subjects would construct

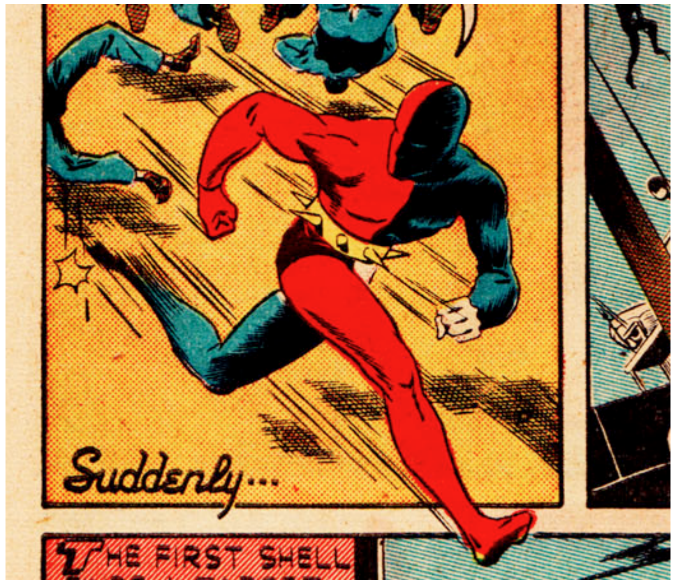

Figure 2.10: A comic character depicted in motion via the use of speedlines.(Image courtesy of digitalcomicmuseum.com)

origami models. The subjects executed this task with instructions in text-only form, text accompanied by a final assembly diagram and text accompanied by both a final assembly diagram as well as step-by-step diagrams. The authors noted that both a final assembly diagram as well as step diagrams were useful but problem complexity directly affected which diagram was more useful. A global view worked well for simple tasks. For tasks with increasing complexity, the ability to see individual steps was shown to be a critical factor in a diagram's success as a facilitating tool.

Cheng[33] describes how diagrams can play one or more of twelve functional roles in problem solving. The author describes these roles as "capacities or features that a diagram may possess which can support particular forms of reasoning or specific problem solving tasks". Three examples are: displaying spatial structure, abstracting complex process flows and encoding temporal sequences. The author notes that different problem contexts will necessitate that associated diagrams employ different combinations of roles. These roles are important from an educational perspective because they also aid in conveying information to a learner.

It is also important to be able to portray actions, especially longer ones, as being made up of components. A learner's initial view of a complex activity may be at a high level but this view is not conducive to learning the activity. As Novick showed that complex tasks can be executed more effectively using individual steps, Zacks et al.[165] discuss how individuals are biased to perceive activities as being made up of inter-related discrete events. A system which provides pedagogical visualization should, then, operate within these inherent bounds. Motion can be segmented accordingly in two distinct ways. A motion can be temporally segmented into distinct gestures or a motion may be segmented anatomically by focussing a learner to a particular anatomical portion of a motion animation. Our system will allow a user to view a motion as a whole or in terms of its discrete component actions segmented anatomically.

Another recent area of cognitive research pertains to the effectiveness of animation as an educational tool. Jones and Scaife[80] assert that the usefulness of animation for educational purposes may be limited. The authors conducted an experiment using animated and static graphics to instruct two groups of learners about a dynamic process. The learners were taught about the function of a heart valve with one group being shown a static graphic depicting the subject matter. The other group were shown an animation of the valve function. Afterwards, the learners were tested on their newly acquired knowledge. The authors found from the test results no particular advantage to using animation. The authors did note, however, that ability of animation to depict more information regarding changes in motion and temporal sequencing. Tversky et. al.[153] conducted a metastudy analyzing results from several experiments to address the question of whether animation can facilitate learning. The authors were selective in their choice of experiments to review because they note 'some of what has been called animation has involved other aspects of communication situations, especially interactivity, which is known to benefit learners on its own'. The authors primarily review the use of animation to teach complex systems. The authors note some limitations to the use of animation but also noted some important exceptions to this view. For instance, they describe the problem of finding static graphics which were equivalent to the animations since many animations contain additional information that a static graphic cannot. One could argue that this observation may support the notion of an inherent advantage of animation over static graphics based on the animation's ability to carry more information. The authors also note that animations lend themselves better to incorporating interactivity which they note as a known cognitive aid. The authors close the paper with a caveat to their work where they note that animation may be beneficial in instances where the purpose is to convey spatiotemporal changes and reorientations.

2.2 Motion Capture

A formal discussion to define motion capture can be difficult. Generally, the idea is to record the motion of a set of points on an object. A common use of this data is to retarget the motion onto animated characters. This eliminates the need for talented animators who can recreate such movements, or, the design of mathematical formalisms that describe the spatiotemporal movements for objects in a complex world. Doing this in real time requires a technological solution. As well, the set of points chosen must be representative of the whole object. There have been various terms used to refer to it such as performance animation, performance capture, virtual theater, digital puppetry[46], real-time animation or even the contracted version of the original 'motion capture' to mocap. These last two seem to have become the accepted nomenclature but while the name may be settled, the technology continues to change and advance.

An optical motion capture environment is a type of lab environment. It requires considerable space as well as a significant financial investment for the system and associated support staff. Figure 2.11 shows an example of how such a setup would typically be implemented. Six to ten motion capture cameras are mounted near the ceiling and angled down towards the location where an actor would be positioned. The actor must be fitted with reflective markers which work in conjunction with the cameras. The cameras are actually a combination transmitter/receiver of infrared flashes. The cameras emit an infrared flash which is reflected off of the markers. This reflection is then recorded by one or more of the cameras in the setup which can then calculate the location of the marker based on the reflection recorded. This process is repeated some number of times per second for each marker and this becomes the framerate for that recording session.

This traditional methodology has seen some competition in recent years with the advent of other types of motion capture, namely, video-based motion capture and inertial motion capture systems. Liu et al.[101] proposed a method of markerless motion capture which they refer to as video-based motion capture involving the use of a human model which is used to track the motion of an actor within a video. The authors leverage their a priori knowledge of this model to calculate body positions and construct a capture. The obvious advantage with this method is the elimination of equipment and space requirements and their associated cost. However, accuracy for video-based motion capture has been a

Figure 2.11: A motion capture lab. Note the track-mounted motion capture cameras in upper part of the image.(Image used by permission from the Human Balance and Ambulation Research Laboratory at the University of Missouri - Kansas City)

significant concern. Gleicher et al.[58] noted in an analysis of video based motion capture that 'incorrect reconstructions are not only possible but inevitable'. Still there may be applications for which this type produces enough accuracy.

A more recent development in motion capture is inertial motion capture. This type makes use of inexpensive accelerometers and gyroscopes to infer the three-dimensional location of the actors body during motion. Zhang et al.[166] describe an inertial system using such devices to track the location of a actor's limb. By recording angular velocities from gyroscopes and acceleration information from the accelerometers, the authors were able to construct location information using quaternion based methods.

Such an inertial system has some distinct advantages. First, it can make a form of direct measurement of motion rather than trying to approximate location information as in the case of video-based or optical methods. Because of this, accuracy is largely dependant on advances of the hardware being used. That is, improvements to the gyroscopes and accelerometers to provide more accurate information. Additionally, systems such as *xSens MVN*© also employ wireless technology for data transmission which gives the system a certain amount of mobility not available in a traditional setup. Finally, this type of equipment boasts a much lower hardware cost than traditional lab-based systems as seen by its inclusion in devices such as the Nintendo Wii system as well as in Apple Computer's

Figure 2.12: A motion capture camera used by an optical motion capture system.

iPod, iPad and iPhone devices. Because of this cost factor, inertial systems could eventually be within reach of the average consumer.

A great deal of work has been done in the area of motion capture regarding segmentation[10] of captured motion and retargetting[25] captured motion to new actors. Motion capture does entail a cost factor. While some new systems do not require the laboratory-like setup that many systems employ, there is still overhead. There is a need to hire an actor and have the desired motion planned and scripted for the actor to perform. As with all acting, this process is subject to human error which requires re-executing the desired motion until acceptable results are produced. Consider also that this process may be necessary to repeat even for motions which have subtle differences from existing motion capture data. There is a clear advantage of being able to synthesize motion from existing data. This is done by segmenting existing motions into small individual actions, synthesizing new motions by transitioning a sequence of these components together and retargetting the result to a new actor. This allows for a potentially large number of new motion sequences to be generated from existing data.

Automated segmentation of motion capture data is an important processing step for data used in games, animations and commercials. Segmenting a motion sequence into small discrete component actions makes the data more easily applied to retargetting. Segmenting longer motion sequences manually is accurate but tedious and labour intensive. Barbic et al. discuss methods of automatically segmenting motion sequences into distinct behaviours[10]. The authors employ three methods for segmentation and compare the results to determine which displays the best performance. They show that using a probabilistic principal component analysis(PCA) method produces resulting accuracy above ninety percent. The authors hypothesize that the probabilistic method's performance is due to its ability to track changes in the distributions that characterize motions.

Motion capture segments are used to synthesize new behaviours. Of primary concern is how to seamlessly join discrete motion segments. Kovar et al.[90] describe a method of creating motion graphs. Motion data is viewed as individual clips of motion that can be joined together. A motion graph is a directed graph where motions are the edges of the graph and nodes are the joins between the motions. The primary issue here is transitions. Any two motions are unlikely to join together seamlessly without an appropriate transition. A transition is another type of motion clip and so therefore is another edge in the graph placed between two motion segments to join them together. By constructing a graph of motion clips, the clips can be "sewn" together to create longer motion sequences.

Retargetting is another important area of research. Bregler et al.[25] describe a technique called cartoon capture. The authors describe a process for extracting motion information from cartoon animation and retargetting the motion to different animated characters. This is accomplished by parameterizing the motion in terms of affine and key-shape deformations. The affine parameters describe the coarse or global motion in terms of translation, rotation, scaling, etc. while the key-shapes capture more local deformations. For example, a frog can move from point to point as a global translation. Part of this movement, however, involves localized deformations like the frogs legs extending and contracting as it jumps. The authors retarget this captured motion to different characters that have also had their own localized key-shapes identified. For example, the frog jumping can be retargetted to a rabbit. The rabbit's key shapes are unique to itself but there must be some retargetting from one set of key shapes to the other.

A particular issue with retargetting motion is differing scale in the target actor. The target may be identical proportionately but a different size or the target may be the same size but proportionately different. This issue is discussed by Gleicher[55] who proposes a method which accounts for this scaling issue by first capturing parameters of the motion, scaling them and identifying a center for scaling. This center is context specific and therefore not necessarily the origin.

The issue of differing scale in actors is also a consideration if one is to compare two examples of the same motion captured from two different actors.

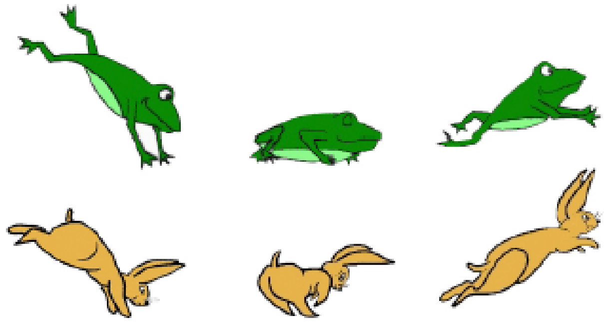

Figure 2.13: An example of a motion retargetted from a frog to a rabbit.(Image used by permission from Dr. Lorie Loeb)

2.3 Comparison and Analysis of Motion Capture Data

Comparison of two examples of the same motion is not a common research area. Maekawa et al.[103] describe a learning system whereby captured motion of a learner can be compared to that of an expert. The authors create a system which records motion data from a learner and the software displays the animation of this data along with the expert data for visual comparison. While this provides a useful educational tool, it is limited. The learner must discern the differences in the motion visually and determine what is necessary to make corrections. We propose a system which will compare the data and display an animated visualization of the differences for the learner.

Another shortcoming of this method is the naive scaling used. The authors calculate a simple ratio based on differences in the actors' height. This ratio is then applied to all body segments equally. The problem with this method is it assumes all actors to have identical body proportions. When comparing datasets from two different actors, we would propose that a method of normalizing all learner data to the expert data segment by segment be utilized. This would require a scaling factor to be used for each body segment but would result in a more accurate comparison since this form of normalizing will account for differences in proportions between expert and learner.

Motion capture data can be analyzed for a particular purpose. Assa et al.[8] describe a method for producing an overview video based on analysis of motion capture data. The author's method involves analyzing motion data to determine salient motion segments and limbs within these segments. This analysis is then used to determine the optimum camera viewpoint and path for the video. The authors note the difficulty in determining saliency since there can be considerable subjectivity in what constitutes a salient motion segment. Two of the authors, Assa and Cohen, have done previous work in pose selection[7] where the authors describe a method of selecting key poses based on analysis of skeletal animation.

Motion capture analysis and processing has also seen elements of signal processing applied. Bruderlin and Williams[28] describe a system for modifying animated motion by treating the motion as a signal. By converting the original information to this format, the authors can then apply various signal-based transformations on the data to produce different effects. This is not a new technique in itself as it has been used in computer vision for some time. It is novel for these techniques to be applied to motion capture data.

2.4 Annotation

Annotation of data is a common operation in various contexts. The ability to attach additional semantic information to data can have considerable benefits for processing. We concern ourselves with annotation pertaining to graphics. Research in this area has revolved around annotating static graphics to provide additional meaning to the viewer. Sonnet et al.[144] describe a method of adding dynamic text annotations to exploded diagrams(figure 2.14).

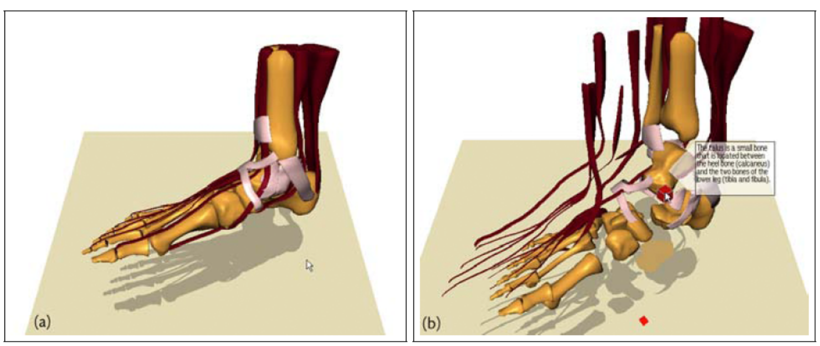

Figure 2.14: An example of an exploded diagram with an annotation added.(Image used by permission from Dr. Thomas Strothotte)

The author's system not only provides the annotation mechanism but also maintains readability in spite of user manipulation of the model components. Preim et al.[124] also describe the use of text-based caption annotations in visual interfaces ranging from anatomical models to cartography.

Annotation does not have to be strictly text-based. Graphics-based cues can be added to graphical models to demonstrate importance of certain components. Isenberg et al.[76] describe a method for adding emphasis to particular parts of line drawings by using different types of line effects. Visual cues can also be used to annotate text as in a paper by Bouvin et al.[22] where the authors describe a method for producing annotations for hypermedia.

Graphical cues have been shown to be beneficial within a strictly graphical context as well. Agrawala et al.[2] describe a system which employs visual primitives such as arrows and dashed lines to imply assembly direction to the viewer of assembly instructions. The system works by first choosing an assembly sequence and then presenting the sequence in a series of static images annotated with the above primitives.

Visual cues and the data to be annotated can both be animated. In the case of motion capture animation, it is possible to provide visual cues derived from the captured motion. Bouvier-Zappa et al.[21][20] developed a system that analyzes motion capture data to produce both static and animated annotations. The static annotations are created as part of static images. The animated annotations are produced in the context of an animation

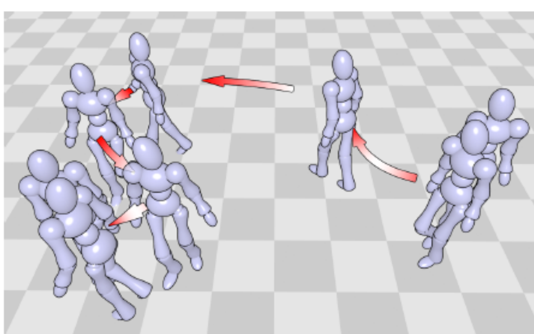

Figure 2.15: An example of a key frame sequence.(Image used by permission from Simon Bouvier-Zappa)

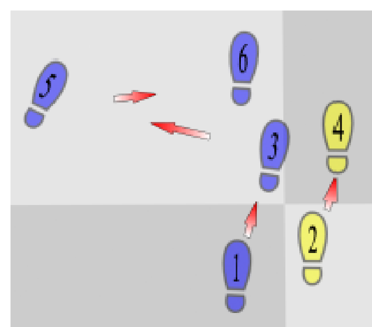

Figure 2.16: An example of a foot sequence diagram.(Image used by permission from Simon Bouvier-Zappa)

of the motion capture data. The authors determined that a reasonably small visualization vocabulary was necessary, producing compelling results using arrows, noise waves and stroboscopic effects. The system is also capable of producing key frame illustrations and foot sequence diagrams(figures 2.15 and 2.16).

The authors note a potential extension to their work would be annotating a comparison of expert and learner motions to provide greater learner insight and therefore improvement. This extension will be the subject of the remainder of this thesis.

2.5 Graphics

It is difficult to believe that as recently as 1980, computer graphics was taught from a purely theoretical point of view at many universities, because hardware was simply too expensive. In fact, relatively ordinary computers by today's standards were expensive, and devices for printing images were beyond the budgets of most universities. The development of computer graphics hardware has been fuelled and subsidized by the huge demand of video gamers for increasingly detailed real time animations. Even with modern graphics cards, there are important tradeoffs, and we review some background.

The earliest concepts of computer graphics were based on ray tracing, that is, tracing the activity of light through a scene, into a viewer's eye. There is far too much light bouncing around the real world to make this problem tractable, so the problem was simplified as follows. First imagine that the retina/back of the viewer's eye is a rectangular grid of pixels (picture elements) each of which can contain exactly one colour. For each pixel on the retina, a ray is cast through the lens of the eye, into the scene, until it intersects with some geometry in the scene. All intersections are recorded, and for the nearest intersection, the colour of the surface at the point of intersection is determined, and the pixel on the retina is *painted* accordingly. The colour of the surface takes into account, to the greatest extent possible, the lighting in the scene. This may get very complicated where mirrors and translucent objects exist, and when light bounces off coloured objects to a surface, but makes possible an arbitrarily complex level of visual detail. Another advantage of this form of rendering is that curved surfaces may be represented using conics and other exact geometry, which allows a high level of surface realism[51].

Rasterization is a very different approach. In this system, surfaces are represented as meshes of planar polygons, typically triangles. Triangles have many surprising properties that make them ideal for work in real time computer graphics. For example, imagine placing values at the vertices of a triangle, and linearly interpreting those values on the interior. For quadrilateral and other shapes, the interpolation is not stable under rotation. When the interpolated value represents a colour, this gives an image stability as the object moves about. For each triangle, the colour of light is computed only at the vertices, and interpolated using simple linear interpolation at interior points. Each triangle is rendered into a frame buffer, which almost exactly corresponds to the buffer of pixels in ray tracing.

The difference here is that every single pixel of every triangle is rendered. Occlusion is handled by rendering the triangles from back to front, or by using a technique called the z-buffer, which provides an assist by only rendering pixels known to be nearer than the previously rendered pixel. In the worst case, where triangles are presented to the renderer back to front, all pixels will still be rendered in any case. Whereas ray-tracing is still typically implemented in software, it is possible to design very fast hardware implementations of rasterization.

Thus, rasterization was the basis of the OpenGL API, which was originally sold by Silicon Graphics Ltd., and eventually released free with many operating systems in the 1990s. What makes rasterization appealing is that it is easy to render animations of simple scenes in real time. As well, it is possible to turn certain features of the OpenGL engine on and off as needed, if hardware cannot keep up to the desired frame rate. For example, at one time, programming manuals recommended that specular illumination (highlights, such as the reflection of a light source on a marble) be turned off for realtime animation, as this component of the lighting equation was the most expensive to compute and the other components permitted a satisfactory image. Hardware advances have made this unnecessary in all but extreme situations[140].

A recent development that has changed the rendering game considerably is the recent advent of GPU(graphics processing unit) cards, where, in essence, an entire (but limited) processor is placed at each pixel, and all processors act in parallel. Given access to geometry for the entire scene, this could make hardware-based ray-tracing feasible. This is not necessarily an issue for the present work, as we felt photorealistic animations were not necessary for the instructional goals pursued here.

Chapter 3

Problem Statement

In this work, we present a proof-of-concept prototype for visualizing datasets representing expert and learner data in a way that would provide learners with a useful training tool. We believe that by providing a learner with this tool, this could help to accelerate or at least facilitate athletic learning. This work is exploratory research intended to demonstrate the potential of using motion capture data in a pedagogical tool that helps student athletes to compare their performance to that of virtualized world experts. Additionally, the scope of the work involved to design and produce the prototype was sufficiently extensive that it was not possible to both build the prototype, and perform classical user studies.

An ideal product would observe the student learner and, if the product believed the student was exhibiting buggy behaviour, it could so advise the student. If the behaviour was correct, but containing errors of degree, it could also advise the student. The purpose of this work was to take some initial steps in this direction, by way of determining feasibility of building a real time system of this nature. To do this, we had to make many simplifications.

First we decided to proceed with a prototype which compared two full datasets and visualized the results rather than attempt comparisons to a stored 'bug library'. This was necessary since either of these two types of operations could constitute a full scale project and therefore would be too much to fit into the scope of this work.

Secondly, it was necessary to put our recorded data into a strict form for input into the system. Our data is truncated at the beginning and end to ensure temporal alignment of motion sequences. This was necessary to avoid problems with trying to detect the precise start and end of a motion in two separate datasets as this also would prove to be a significant problem in itself.

This work is an extension of work done by Bouvier-Zappa et. al [20][21]. The authors produce a system to process motion capture data in order to produce visualizations of the data. The visualizations are static images or animations created from a combination of the input motion capture data and particular visual cues to provide the user with additional insight into the viewed motion. Examples are shown in figures 3.1 and 3.2.

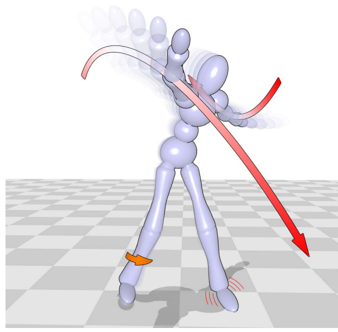

Figure 3.1: A motion illustrated using motion arrows and stroboscopic motion.(Image used by permission from Simon Bouvier-Zappa)

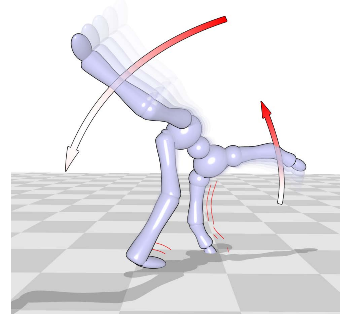

Figure 3.2: A motion illustrated using motion arrows and noise waves.(Image used by permission from Simon Bouvier-Zappa)

The authors base their system on a hierarchical view of the motion data. By subdividing the human body into a tree structure, the authors can produce animations at varying levels of their hierarchy. Higher levels represent more general groupings such as upper and lower body. Lower levels represent more detailed grouping such as right or left hand. This allows for producing animations at varying levels of detail depending on what the user requires.

Another important component of Bouvier-Zappa's work is the notion of *motion cues*. The authors define a small set of significant visual cues for conveying information about properties of a particular motion. The cues used are motion arrows, noise waves, stroboscopic motion, pose illustration and foot step illustration.

These cues provide varying degrees of information about a motion. Motion arrows are simply arrows added to an animation or static image to show linear, curvilinear or rotational motion. Arrows are noted as a very powerful indicator of motion and are the primary indicator used in this system. Noise waves are a type of action line drawn as a repetition of the actor's silhouette outline. The purpose of a noise wave is to express motion that is too subtle to be expressed with an arrow.

Other motion cues provide different ways of viewing a motion. Stroboscopic motion is a repetition of several poses of an animation taken at time intervals to display a progression of motion within a single static image. Key pose illustration itself juxtaposes several poses of a motion within an image. The important difference is key pose illustration requires that the system *chooses* what the key poses are. Foot sequence illustration provides a means for illustrating foot sequence patterns that may be difficult to capture using arrows or other cues.

To indicate a motion using a particular cue requires that the motion be analyzed and a particular cue fitted. All motions are represented with motion curves and quaternion data. A motion curve represents a motion as a progression of locations for a particular control point over time. A control point might be a location at a joint or a location at the midpoint between two consecutive joints (see figure 3.4). While this representation works well for translational movement it cannot capture rotational movement. Quaternion data is used in certain cases where rotational movement is to be described. A quaternion is a three dimensional vector coupled with a rotation angle. The angle is the amount of rotation occurring around the vector.

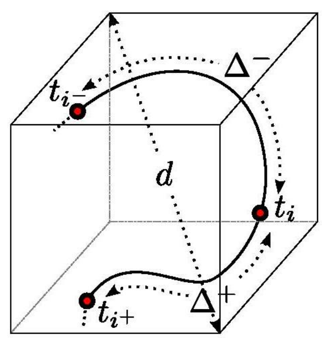

Figure 3.3: The amplitude of a motion is represented as d.(Image used by permission from Simon Bouvier-Zappa)

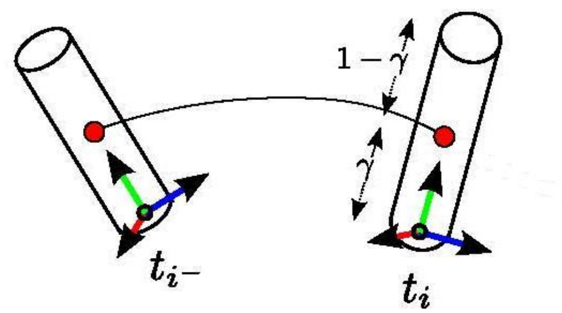

Figure 3.4: A motion curve comprised of locations of the midpoint between two joints.(Image used by permission from Simon Bouvier-Zappa)

Motion arrows are used to represent most obvious motions as translations and rotations. A particular motion curve from a movement is analyzed according to the length, velocity and acceleration of the curve to find if the motion is translation or rotation. The authors' system attempts to fit an arrow to the motion curve based on this analysis. Longer and faster curves are favored to be viewed as translations. In cases where the motion curve does not have an appropriate fit to denote the motion as a translation, the system can then attempt to calculate a roll angle from the quaternion data at a skeletal node. Either type of arrow is illustrated such that it is always visible and therefore never occluded by the animation actor.

To produce a noise wave, the system calculates an amplitude for a particular motion (see figure 3.3). Noise wave motions will have an amplitude significantly less than that of a motion described by motion arrows but greater than that of simply noisy data. The amplitude is calculated as the length of the diagonal of the three-dimensional bounding box of the motion curve. If the amplitude of the motion curve is in an appropriate range noise waves will be added.

Stroboscopic motion is determined by simply drawing successive poses of an animation and illustrating prior poses with increasing transparency. Key pose illustration involves a combination of illustrating particular poses deemed as key poses and motion arrows to display the progression of the poses. Foot sequence illustration makes use of research into foot plant retrieval and the foot skating artifact in motion capture animation [54] as well as research in motion graphs [90]. Footplant locations are determined on a 2D ground and again further illustrated using motion arrows.

Annotations, however, do not directly instruct the learner. The annotations provide cues to *inform* the user of the mechanics of what the user is viewing. To instruct the learner it is necessary to provide additional information to direct the learner towards a particular desired outcome. Traditionally this would be executed by the expert verbalizing or demonstrating an important difference between the learner's action and their own. Doing so allows the expert to provide feedback which is directive or corrective to the learner to guide the learner toward the intended outcome. Cho et. al[37] define directive feedback as "explicit suggestions of specific changes"' and note it as being particularly beneficial for learners albeit in another area of pedagogy. We propose building upon the work of Bouvier-Zappa by producing a system that is able to process two datasets comparatively. The output of this system is a third dataset representing the salient differences between the original datasets as well as additions to the original motion capture animation to visualize these differences for the user.

Chapter 4

Data Collection and Visualization Procedure

4.1 Outline

The first step for this experiment was data gathering. This was done using motion capture equipment made available by the College of Kinesiology. Actor volunteers performed four motions. Those motions were:

1) weightlifting movement: arm curl

2) martial arts front kick

3) martial arts cross punch

4) a golf swing

The actors executed these motions two times. The first was the correct motion and the second was a specific incorrect motion. We determined, in advance, what the intentional mistakes were to be. Our actors performed the following incorrect motions:

1) weightlifting movement: arm curl - lean back instead of remaining straight

2) martial arts front kick - kicking by only hinging the knee joint as opposed to moving the entire leg appropriately

3) martial arts punch - leaning forward to reach towards a target

4) a golf swing - this was executed once without a corresponding error and was eventually used for temporal difference detection

This phase was completed at the PAC facility at the University of Saskatchewan campus.

Optical motion capture systems capture human motion using reflective *markers*. Several infrared cameras transmit infrared light which reflects off of the markers and is registered by other cameras to triangulate the markers location in 3D space. An actor will have several markers placed on him/her to capture a desired motion as shown in figure 4.1. The particular marker setup and marker locations is dependent on what sort of output is desired. For our work, we wanted joint center data in an *(x, y, z)* format. By joint center, we mean the point that would exist at the physical center of an actor's joint such as a knee or elbow. This meant that actors would have markers placed on them such that joint center locations could be inferred from the marker locations. As an example, we would place markers on either side of the wrist and the joint center inferred from this would be the point halfway between them which should approximate the center of the wrist joint. The locations of these markers are recorded over time at a predetermined frame rate. For our data collection, we used 100 frames per second. Post-recording processing results in a set of data which represents specific joint centers. The set of locations over time for a particular joint center forms a curve in 3D space. We will refer to these as motion curves in our discussion.

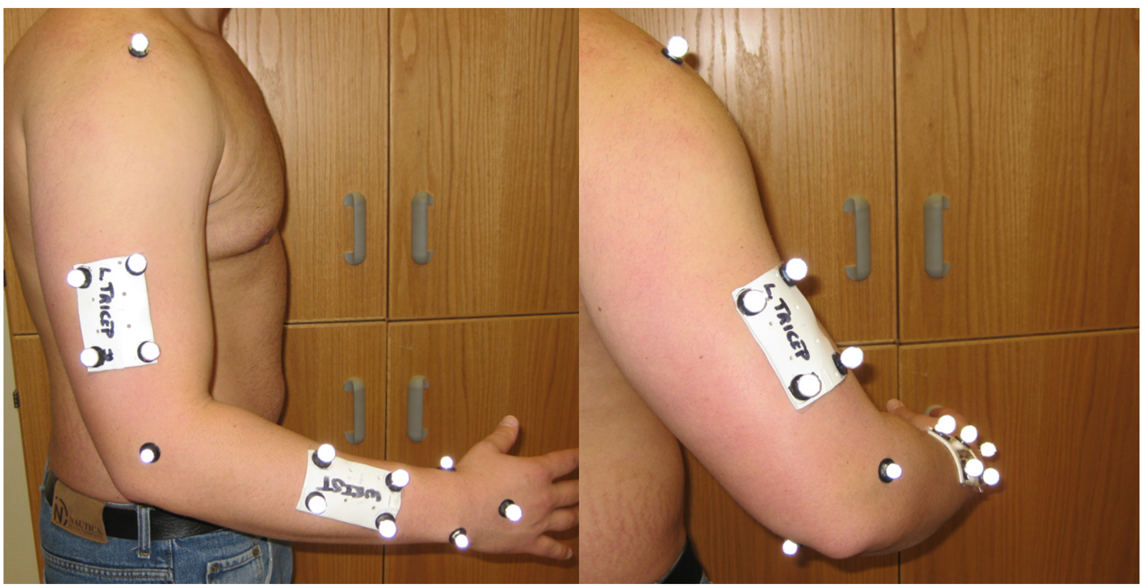

Figure 4.1: An example of a marker setup which would allow for approximation of elbow and wrist joint centers.

Once the data was gathered, it was processed by our software tool. In our case, each actor has a correct and incorrect dataset. We found, upon comparing different actors, that there was additional noise that was otherwise unnoticeable but manifested itself when we attempted to normalize different actors. As a result, we use correct and incorrect datasets for the same actor. Normally, the learner and expert would be different actors and this would necessitate a normalization step. We would not be able to assume that physical proportions of various actors will match so we would need to normalize the actor datasets to the expert dataset. The normalized datasets could then processed using spline methods to convert them to a canonical form appropriate for direct comparison. The two canonical datasets are compared for spatial and/or temporal data differences in their individual data points. Any differences are stored as a third dataset. Temporal differences in particular are difficult to process and quantify. Some assumptions are necessary here to facilitate being able to produce an initial result. For instance, it must be assumed that we know what the start and end points of the motions are.

This difference data will be visualized by the software tool as a pausable animation with visual cues added to represent the differences stored in the third dataset.

4.2 Normalization

As noted above, we are not comparing different actors so a normalization is not necessary. However, since it would be necessary in practice, we have included a short description of it here. Normalization of datasets is an essential process step. The data we are comparing represents different actors characterized by different body shapes, scales and proportions. In a purely raw form, the data could not be directly compared. We use a normalization step to normalize learner datasets to the expert dataset as a first processing step.

We start by ensuring correspondence in the actor's core location. The core location can be defined as one central point that can be said to define the actor's location as a whole. We will use the center of the chest as the actor's core location. So we ensure that the learner's core is centered to be the same as the expert's. Normalization will then work analogously to a tree traversal where the core location is equivalent to the root of the tree. We start by adjusting the distance between the core location and the hip center. we can then progress outwards moving to hips/shoulders followed by elbows/knees, and so forth. Once this step is complete, we can then parameterize the datasets for comparison.

4.3 Parameterization

There is an important problem in comparing two datasets consisting of samples that are independent of each other. We cannot assume that sample points will line up precisely. Such an assumption could lead to an erroneous comparison as shown in figure 4.2. To solve this problem, we will parameterize the data. The process we use is fairly straightforward and fast. First, we assume that each set of points is generated by a continuous function. Many functions could generate such data, so we approximate each function as a piecewise polynomial. We then use the two functions to generate new datasets of points that are temporally aligned. Each marker such that a motion curve comprised of a set of (x,y,z) coordinate points will be decomposed into three subcomponents: the sets of x, y and z values.

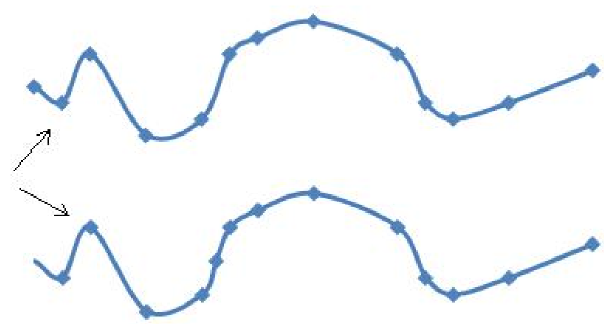

Figure 4.2: The *second* sample point in different locations of curves which are the same.

The conversion of the datasets to a canonical form involves using spline methods to treat the motions as signals. The first step in this process is to create a list of 3D points corresponding to temporally ordered data from each marker. We then subdivide this list into three constituent listings for each dimension, that is, a list of x values, a list of y values and a list of z values in the same temporal order. We treat these subsets as samples of one dimensional signals. Each sample is processed using spline interpolation to produce a canonical format which can be used for later comparison.

The particular type of spline used here is a uniform non-rational b-spline. These splines consist of curve segments whose polynomial coefficients are dependant on a small number of control points. In our case, the coefficients for each curve segment are dependant on four control points. This method has the advantage of lower computation time for the coefficients compared to natural cubic splines. This is of particular importance for this work as our software tool is intended to work in realtime. B-splines have the same continuity as natural splines but do not interpolate their control points.

B-splines approximate a series of $m+1$ control points $P_0, P_1, ...P_m, m \geq 3$, with a curve that consists of $m-2$ cubic polynomial curve segments. Where $m > 3$, and thus more than a single curve segment, there is a joint point or *knot* between each curve segment. Further, the initial and final points of a segment are also considered to be knots. There are $m-1$ knots. This arrangement is shown in figures 4.3 and 4.4. These examples might make our approximation method appear crude but, in fact, it is highly accurate. Since we are recording one hundred frames per second, one could imagine that these diagrams, by analogy, represent an extreme close up of those points. Figure 4.5 and its accompanying information regarding our error demonstrate this.

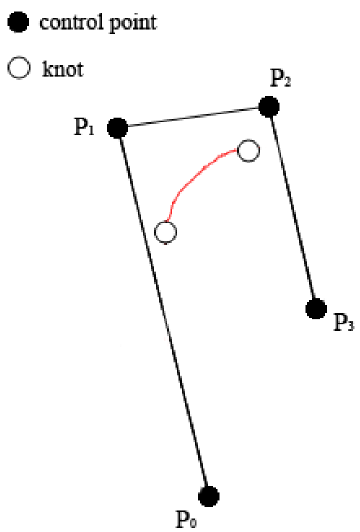

Figure 4.3: A simple spline consisting of one curve segment defined by four control points.

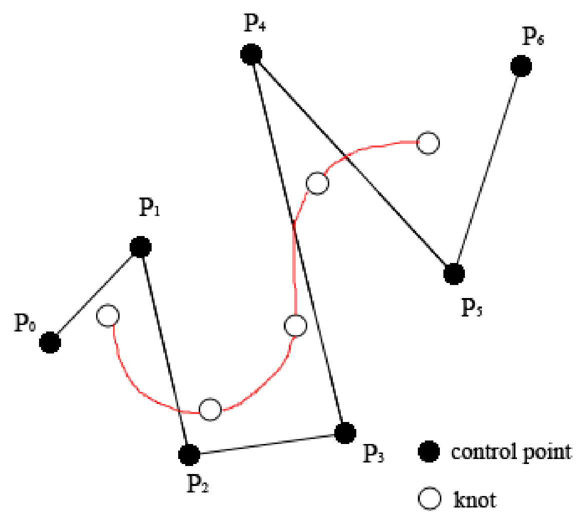

Figure 4.4: A longer spline consisting of several control points and segments.

The name of this type of spline warrants some concise explanation. The term uniform refers to the fact that knots are spaced at equal intervals of a parameter t. This parameter can be thought of as a temporal parameter, thus indicating the points are a fixed time apart. The term non-rational is meant to differentiate these splines from rational cubic polynomial curves where $x(t)$, $y(t)$ and $z(t)$ are ratios of two cubic polynomials. The "B" stands for *basis*. This refers to the fact the splines can be represented as a weighted sum of polynomial basis functions.

The mathematical descriptions of these curves is straightforward. The following closely follows the explanation provided by Foley et al.[51]. The equations express the combination of the control points, basis functions and values of the parameter t. The first curve segment,

C_3 is defined by control points P_0, P_1, P_2 and P_3 over a range of parameter t of t_3 to t_4. The next curve segment, C_4, is defined by control points P_1, P_2, P_3 and P_4 over a range of parameter t of t_4 to t_5. The final curve segment, C_m, will be defined by control points $P_{m-3}, P_{m-2}, P_{m-1}$ and P_m over a range of parameter t of $t_m = m - 3$ to $t_{m+1} = m - 2$. We can note that, in general, a given curve segment C_i will begin somewhere near P_{i-2} and end somewhere near P_{i-1}.

This relationship is expressed by the equation

$$C_i(t) = T_i \cdot M_{B_s} \cdot G_{B_{S_i}}, \quad t_i \leq t < t_{i+1} \tag{4.1}$$

We define T_i as the row vector

$$T_i = \begin{bmatrix} (t - t_i)^3 & (t - t_i)^2 & (t - t_i) & 1 \end{bmatrix} \tag{4.2}$$

which expresses the individual increments of t to be used between $t_m = m - 3$ and $t_{m+1} = m - 2$. M_{B_s} is the *b-spline basis matrix* as

$$M_{B_s} = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \tag{4.3}$$

Finally, $G_{B_{S_i}}$ represents the *b-spline geometry vector* for a curve segment C_i. The geometry vector is a column vector as

$$G_{B_{S_i}} = \begin{bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{bmatrix}, \quad 3 \leq i \leq m \tag{4.4}$$

This vector is populated from the inputted sample points of each 1D signal to be processed. The method of population is a *sliding window*. First, process points 0 to 3, then 1 to 4, then 2 to 5 and so forth. This is important because it provides for maintaining continuity across knot points along the curve as any two consecutive curve segments $C_i(t)$ will share three control points.

The algorithm to perform this function is described by the psuedocode in algorithm 1. This required input for this process is one 1D signal from any motion curve in the form

Algorithm 1 Spline Interpolation for Parameterization

$T_i[\,], M_{B_s}[\,][\,], G_{B_{S_i}}[\,] \leftarrow 0$

$Signal[\,] \leftarrow CurrentSignalValues$

$G_{B_{S_i}}[1] \leftarrow Signal[0]$

$G_{B_{S_i}}[2] \leftarrow Signal[1]$

$G_{B_{S_i}}[3] \leftarrow Signal[2]$

for $j = 0$ to $Signal[].length - 1$ **do**

  $G_{B_{S_i}}[0] \leftarrow G_{B_{S_i}}[1]$

  $G_{B_{S_i}}[1] \leftarrow G_{B_{S_i}}[2]$

  $G_{B_{S_i}}[2] \leftarrow G_{B_{S_i}}[3]$

  $G_{B_{S_i}}[3] \leftarrow Signal[j]$

  for $k = 0$ to 1 step by a **do**

    $Ti[0] \leftarrow k^3$

    $Ti[1] \leftarrow k^2$

    $Ti[2] \leftarrow k$

    $Ti[3] \leftarrow 1$

    $Result \leftarrow Ti[\,] \cdot M_{B_s}[\,][\,] \cdot G_{B_{S_i}}[\,]$

  end for

end for

of a one dimensional array. This process will then execute three times for each motion curve in the current dataset. The output of this process (listed as *Result*) is another array consisting of a list of calculated points which represent the original signal in canonical form.

There is one characteristic of this type of parameterization which warrants discussion. This type of spline does not directly interpolate its control points. This means that a certain quantity of error is present in the result. To provide some quantification of this error, we analyzed one sample 1D signal to calculate the average error imparted. The difference between the control points and spline points was compared in two different ways: comparing differences at the control points and comparing differences at the midpoint between control points. The raw differences were calculated for a dataset consisting of five thousand control points and fifty thousand generated points. The raw difference was then divided by the original control point value to express the error as a percentage.

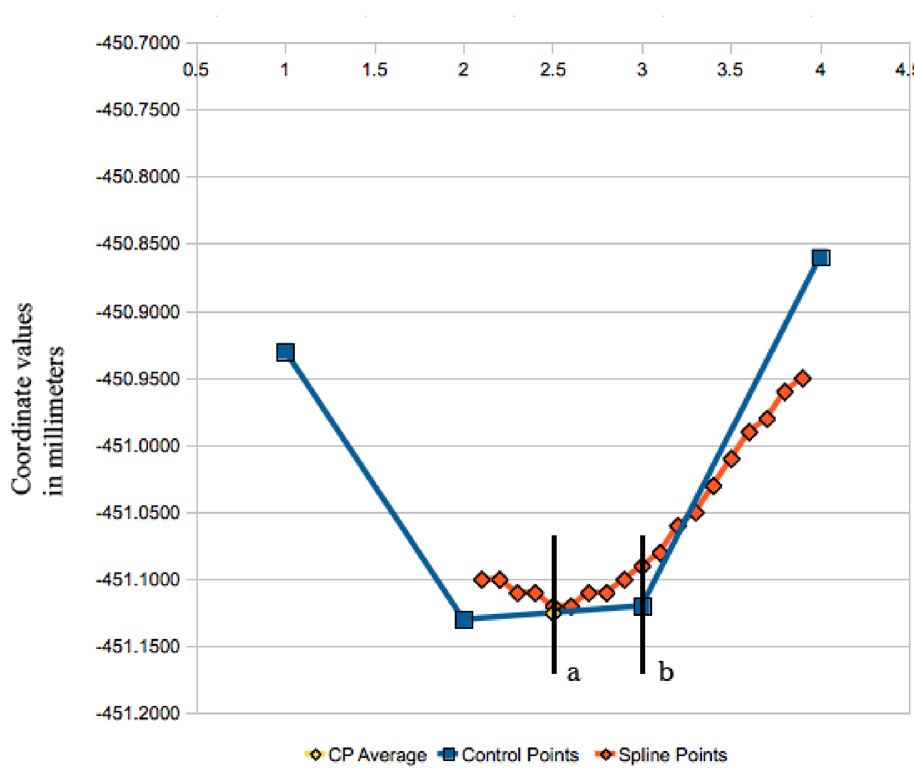

Figure 4.5: Comparison of control points and calculated spline points. a) and b) represent the location of error calculation. a) mean of interior control points. b) exact control point value.

Comparing at the control points, we obtain an average difference in value of 0.08mm.

This value represents an average percentage error of 0.02%. For midpoint comparison, we calculate an average value of the two middle control points for each group of four. This value is then compared with the median calculated value of the corresponding curve segment C_i. The average difference in value in this case was 0.05 which represents a percentage error of 0.01% (see figure 4.5). These values are well below practical margins of error.

4.4 Difference Calculations

There are two types of differences that will be analyzed by our software system: spatial differences and temporal differences. Spatial differences can be calculated in a number of ways. They can be calculated using the midpoint of a body segment (ie. midpoint of femur). Rotational motion can also be tracked and compared spatially. We have chosen to track and analyze the movement of joint centers.

4.4.1 Spatial Differences

The calculation of spatial differences between actor motions is reasonably straightforward once the datasets have been normalized and parameterized. To perform a spatial comparison of two datasets, we calculate a direct difference between corresponding markers in all three dimensions for all markers for all frames. This results in a list of vectors for each marker representing the spatial differences between learner and expert over time.

We start with an expert data set A_e and a learner data set A_l. Each data set comprises the set of joint centers $JC_\alpha[\ ]$ for that actor. Thus we have

$$\{A[JC_\alpha] \mid \alpha = \{ch, rs, re, rw, \ldots, \alpha\}\} \tag{4.5}$$

The following list gives the joint center locations and their corresponding abbreviation:

1) centre of head (ch)
2) right shoulder (rs)
3) right elbow (re)
4) right wrist pair (rw)
5) right hand (rh)
6) left shoulder (ls)
7) left elbow (le)
8) left wrist pair (lw)
9) left hand (lh)
10) right hip (rh)
11) right knee (rk)
12) right ankle (ra)
13) right heel (rh)
14) right toe (rt)
15) left hip (lh)
16) left knee (lk)
17) left ankle (la)
18) left heel (lh)
19) left toe (lt)

Each joint center location is itself an ordered list of (x, y, z) coordinates corresponding to the locations of a particular joint center over time. So any given JC_n can be described as

$$\{JC_\alpha[\zeta] \mid \zeta = \{(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), \ldots, (x_n, y_n, z_n)\}\} \tag{4.6}$$

To obtain a spatial difference between data sets is to find the difference between each coordinate in each list of joint center data. This is done as

$$A_r[\,] = A_e[\,] - A_l[\,] \tag{4.7}$$

Which can be further expressed as

$$A_e[JC_{ch}[], JC_{rs}[], JC_{re}[], \ldots, JC_\alpha[]\,] - A_l[JC_{ch}[], JC_{rs}[], JC_{re}[], \ldots, JC_\alpha[]\,] \tag{4.8}$$

The specific operation at the coordinate level can then be described as

$$\begin{aligned}
A_e[JC_{ch}[(x_1, y_1, z_1)]] - A_l[JC_{ch}[(x_1, y_1, z_1)]],\\
A_e[JC_{ch}[(x_2, y_2, z_2)]] - A_l[JC_{ch}[(x_2, y_2, z_2)]],\\
A_e[JC_{ch}[(x_3, y_3, z_3)]] - A_l[JC_{ch}[(x_3, y_3, z_3)]],\\
\ldots,\\
A_e[JC_{ch}[(x_n, y_n, z_n)]] - A_l[JC_{ch}[(x_n, y_n, z_n)]]
\end{aligned} \tag{4.9}$$

The result set A_r is identical to both A_e and A_l in structure and data appearance. There is one difference with regards to the list of coordinates. In the expert and learner datasets, the list of triplets represent coordinates. in the case of the result set, these triplets are actually vectors which represent the difference between the learner and expert motions. Additionally, the vectors represent the direction in which the learner should adjust their own motion to match the expert.

4.4.2 Temporal Difference

Temporal comparisons of motion data are more difficult. Two examples of the same motion could be roughly compared and one shown to take longer but this alone is not particularly informative. Localized analysis might show the learner's motion is too fast in one segment but too slow in another with a net result of the entire motion taking longer. Providing directive feedback in this case requires that the learner be shown the localized differences.

The calculation of temporal differences between motion data sets will relate to our view of the data as *motion curves*. Looking at a visual example of two one-dimensional signals, we can see what a temporal difference looks like(figure 4.6). The spatial extents of the curve are the same but the curve on the left is more compact while the right-most curve appear *stretched out* by comparison. It may seem incorrect to state that the spatial extents are the same but, in terms of human motion, they are precisely the same since they would move through the same physical locations. What differs is the time of transition from specific location to specific location and therefore a longer timeframe for the same motion.

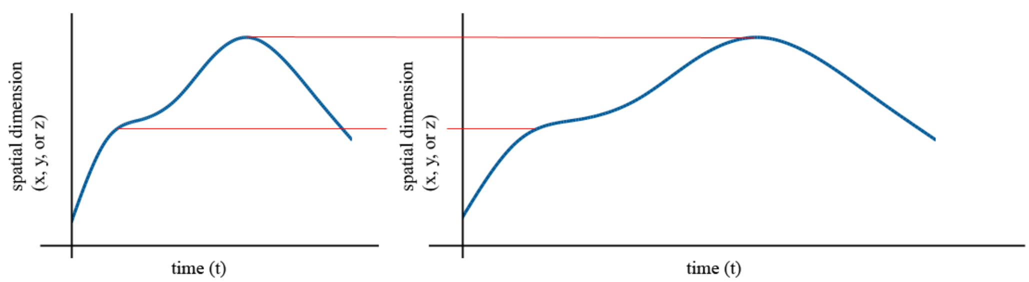

Figure 4.6: Two curves which are the same spatially but differ temporally

To detect temporal differences will first require some means of determining what the localized motion time periods are. We will divide each motion curve into segments that can then be individually compared for temporal differences. To determine the subdivision points, we propose using local extrema. This would allow for temporal segmentation of the curve at spatially equivalent points. Figure 4.7 shows possible extrema in a motion curve. It is important to note that there can be other points in the curve that could be viewed as extrema as well. In fact, in a continuous function there can be infinite extrema. We have used a threshold value to specify, for the purpose of example, a set of extrema in this case and this method is used in the software system as well.

To detect these critical points, we propose a method of scanning the list of points to find particular parts of the list where changes in trend occur. By this method, a trend of decreasing values followed by a trend of increasing values would be characteristic of a local minima. There would be a question as to how many consecutive points constitute a trend. It is a simple matter to choose a value (i.e. five consecutive points) to be considered necessary to constitute a trend. This value can be parameterized within the software

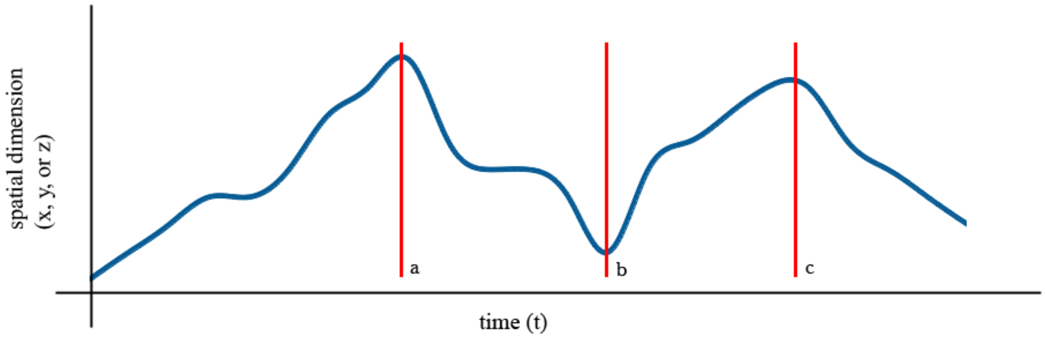

Figure 4.7: Three critical points labeled at locations *a, b, c.*

application and therefore controllable by the user. This would be advantageous as the user could then control how fine-grained the temporal analysis is.

Temporal processing will follow a similar process as spatial processing. Each joint center must be analyzed in terms of its three dimensions. For each joint center dimension, we begin by parsing through the signal values tracking increasing and decreasing trends to find critical points. This is described by algorithm 2.

The output of this algorithm will be a list of indices which correspond to the locations of critical points for the processed curve. In addition to this information, the algorithm also notes the type of the index. That is, does the index represent a peak(P), a valley(V) or a start(S) or end(E) point? This information is important as it will be used in the next step of processing.

Another challenge in temporal processing is similar to the sample point alignment problem. We assume the critical points will not line up and therefore require some matching. This is due to differences between the rate of execution for the learner and expert. This problem is similar to a problem in speech recognition. Sakoe and Chiba[137] noted the requirement of time-normalization of speech signals from different speakers due to non-linear temporal differences in human speech patterns. The authors describe a dynamic programming method which normalizes the time component of the speech sequence so that a comparison can be made for recognition purposes.

To perform dynamic matching, we must first arrange the two sequences to be compared on the axes of a grid as follows:

We use the type information(V, P, S or E) from the previous algorithm as the sequence

Algorithm 2 Tracking of Motion Curve Critical Points

```
x ← firstSignalValue
POScounter, NEGcounter ← 0
for i = 1 to Signal.length do
  currSignalValue ← Signal[i]
  if currSignalValue > x then
    POScounter++
    if POScounter ≥ Threshold then
      if NEGcounter > 0 then
        RecordIndexValue
        RecordIndexType
      end if
      NEGcounter = 0
    end if
  else
    NEGcounter++
    if NEGcounter ≥ Threshold then
      if POScounter > 0 then
        RecordIndexValue
        RecordIndexType
      end if
      POScounter = 0
    end if
  end if
  x ← Signal[i]
end for
```

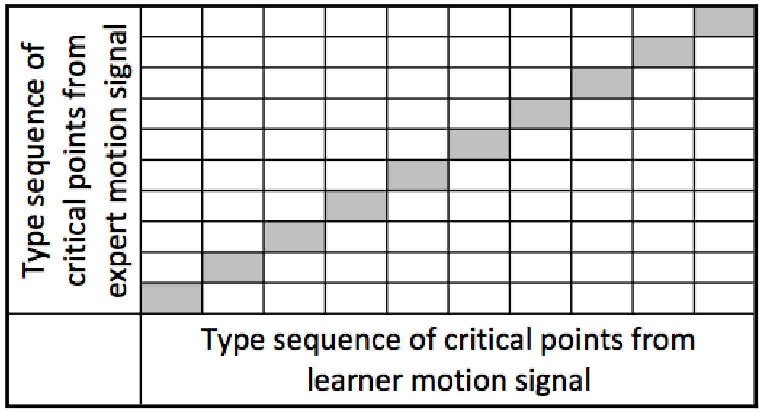

Figure 4.8: Arrangement of signals for dynamic matching.

	V	P	S	E
V	1	5	1	1
P	5	1	1	1
S	1	1	0	8
E	1	1	8	0

Table 4.1: Weights for cost calculation.

information that will be processed in this step. The dynamic algorithm will calculate a cost for every potential match between the learner and expert type sequences. This gives us a matrix DTW of the resulting costs analogous to figure 4.8. We then calculate the shortest cost path from the lower left of this grid to the upper right. Whichever path is the lowest cost will be considered to represent the closest match. A perfect path is shown in the figure as the greyed area of the grid. The distance calculation is performed by algorithm 3 and is sometimes referred to as the *dynamic time warp distance*.

Table 4.1 shows the matrix of weights used for the cost calculation. These weights can be adjusted and are not fixed for all uses. Essentially, we are calculating a lowest cost path so we assign low values to the matches i.e. V to V and higher values to obvious mismatches i.e. V to P. We also use an additional weight in the cost calculation. This weight provides for a higher cost if two tokens are further away from each other in time and space.

Once the indices are located and matched, we can compare segments in one curve with the equivalent segment in another curve to see if the segments are temporally longer or

Algorithm 3 Calculation of Dynamic Time Warp Distance

$cost \leftarrow 0$

for $i = 0$ to $Signal1.length$ **do**

  $DTW[i, 0] \leftarrow \infty$

  $DTW[i, Signal1.length] \leftarrow \infty$

end for

for $i = 0$ to $Signal2.length$ **do**

  $DTW[0, i] \leftarrow \infty$

  $DTW[Signal2.length, i] \leftarrow \infty$

end for

$DTW[0, 0] \leftarrow$

for $i = 0$ to $Signal1.length$ **do**

  for $j = 0$ to $Signal2.length$ **do**

    $cost \leftarrow (Signal1[i].Value - Signal2[j].Value)^2$

    $cost \leftarrow \sqrt{cost + (Signal1[i].Time - Signal2[j].Time)^2}$

    $cost \leftarrow cost * Weight[Signal1[i], Signal2[j]]$

    $DTW[i, j] \leftarrow cost + Minimum(DTW[i-1, j], DTW[i-1, j-1], DTW[i, j-1])$

  end for

end for

shorter. We assume that since we are finding critical points in curves which are assumed to be similar, we can compare the curves segment by segment without concern for wholesale mismatches.

4.5 Visualization

There are multiple forms of feedback provided in our software system. We provide visualizations of both spatial and temporal differences between learner and expert. We allow spatial differences to be aggregated so that we can view the differences in differing levels of detail.

Spatial differences are visualized using arrows. Arrows were chosen in this case as they are an intuitive way to denote direction[154].

Interestingly, humans seem hard-wired to perceive certain kinds of objects pre-attentively (i.e., without semantical processing)[154]. Dreyfuss[48] catalogued international graphic symbols and noted only 14 such symbols, notably the ellipse, the square, blob, line array and cross. Tversky et al.[154] found it interesting that there were so few such symbols, and that all were so simple. Such symbols seem to nonetheless provide a rich, expressive language for diagrams.

Ware[115] observes that humans naturally perceive a closed curve as representative of an object, and, similarly, that humans naturally interpret a line between two such objects as signifying a relationship. Thus circles and lines become the basis for organizational charts.

The origin of arrows in diagrams is not known, though Tversky et al.[154] cite Gombrich[61], who found them used to indicate direction in the 1700s, which is how they are used in the present work. Bertin, in his classic Semiology of Graphics[17], states that the arrow is the "most efficient and often the only formula for representing the complex movement of a point, and, by analogy, that of a line or an area". Tversky et al.[154] notes that Horn[71] found 250 meanings for arrows, including pointing, time and increases and decreases. Given this analysis, our use of arrows to indicate movement, and differences in movement seems semantically well-founded.

With spatial differences, we want to visually direct the learner that to correct their motion requires that they execute the erroneous portion of their motion more so in a particular direction i.e. “move your hand more to the right” or “lift your foot higher”. We do not provide this explicitly but, instead, imply this type of direction visually(figure 4.9).

Our system also allows a user to aggregate the direction arrows. This allows a user to summarize a group of arrows in one area of the body and represent the motion of that area with a single aggregated arrow. For example, an actor’s elbow and wrist could be summarized with one arrow representing the arm as a whole. Figure 4.10 shows an example of this.

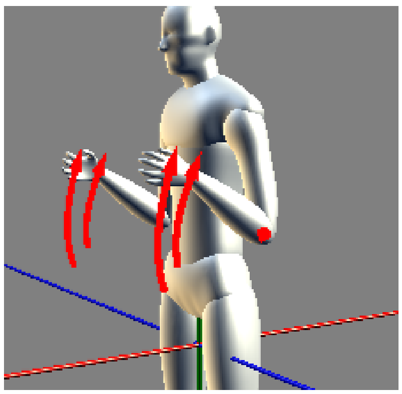

Figure 4.9: An animation with direction arrows implying necessary corrections to a user.

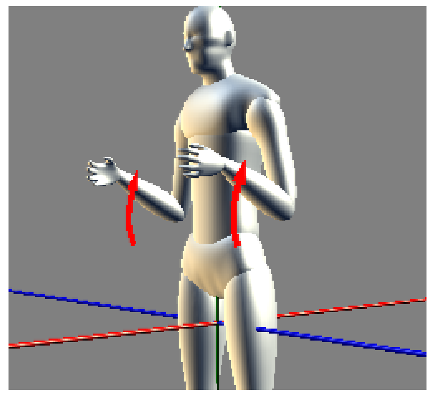

Figure 4.10: An aggregate arrow showing a summarization of the arrows in the figure on the left.

An additional feature is the ability to toggle arrows on and off if a user wishes to focus on a particular part of a motion. A user can also adjust a sensitivity control which sets a threshold for what degree of difference will be visualized. By increasing the control, the user can visualize only the largest spatial differences.

In the case of temporal differences, a different method of visualization was chosen. It is necessary to direct the user differently for temporal differences to avoid confusion between the two. Also, while arrows work well for communicating spatial direction, they are not necessarily effective for communicating movement in time or temporal differences. Instead, we chose to alter the color of the actor’s body parts to denote when a particular body part was moving too quickly or too slowly. This method allows for clearly delineating between the two forms of visualization and could allow for both to coexist with minimal interference between the two. If a marker is moving too quickly, the corresponding body part will be colored red. If a marker is moving too slowly, the corresponding body part will be colored blue.

Chapter 5

Results

The results of our system are encouraging. We have been able to extend the functionality achieved by Bouvier-Zappa to include direct comparison of two sets of motion capture data. The first successful comparison type is strictly a spatial comparison. Following that is spatio-temporal comparison. The figures below show screen captures of the resulting visualizations. In the following screenshots, the lower half of the window is the learner data while the upper half is the expert data. The learner data in figures 5.1 and 5.2 contain specific errors as noted at the beginning of chapter 4.

5.1 Spatial Comparisons

The user needs to receive feedback regarding errors and how to correct them. This requires first that the software is able to identify those errors, and second, that the software can communicate errors in an effective way. The previous chapter showed that we can find critical points in a motion path quickly enough to give back the information in real time. The question arises - how do we display this to the user? We have discussed the use of the arrow as a primary visualization object. The red arrows in the expert animation visualize the movements of the expert's joint centers. The green arrows in the learner animation visualize the differences between the expert and learner at particular joint centers.

This type of visualization provides a considerable amount of information for a user so the interface allows for toggling some of the difference arrows on or off. If a user wanted to focus on just the learner's knee, for example, the could turn off all other arrows by simply clicking on each arrow's associated joint center leaving only the desired arrow visible. Figure 5.3 shows how this feature allows a user to focus on a particular portion of the visualization without being overwhelmed.

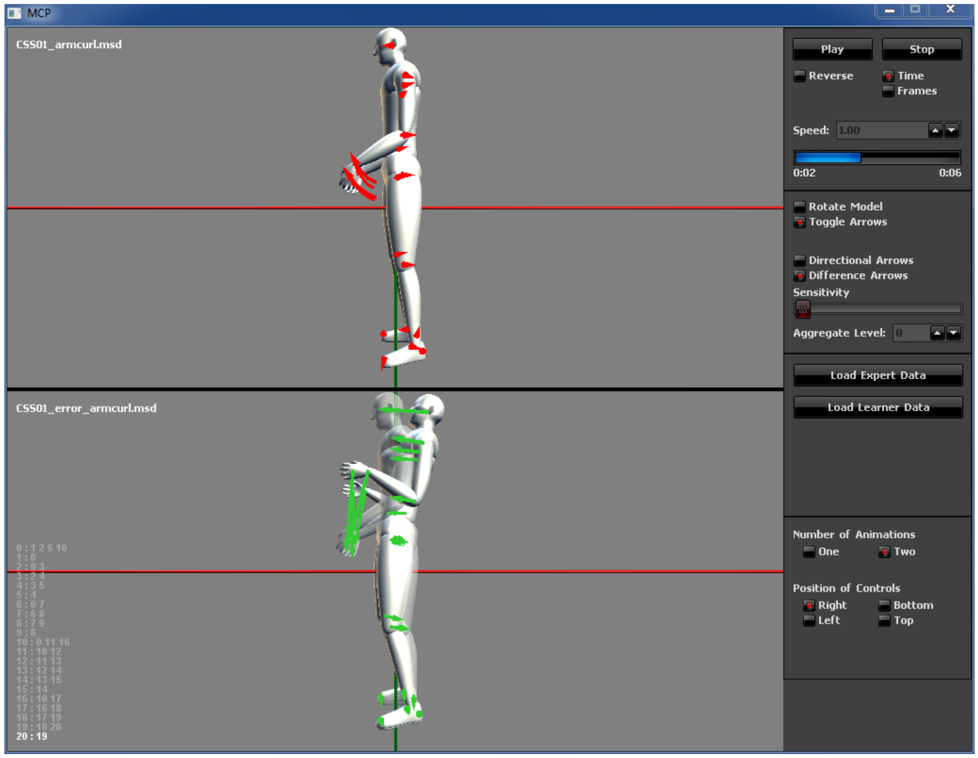

Figure 5.1: A motion capture comparison visualizing spatial differences only.

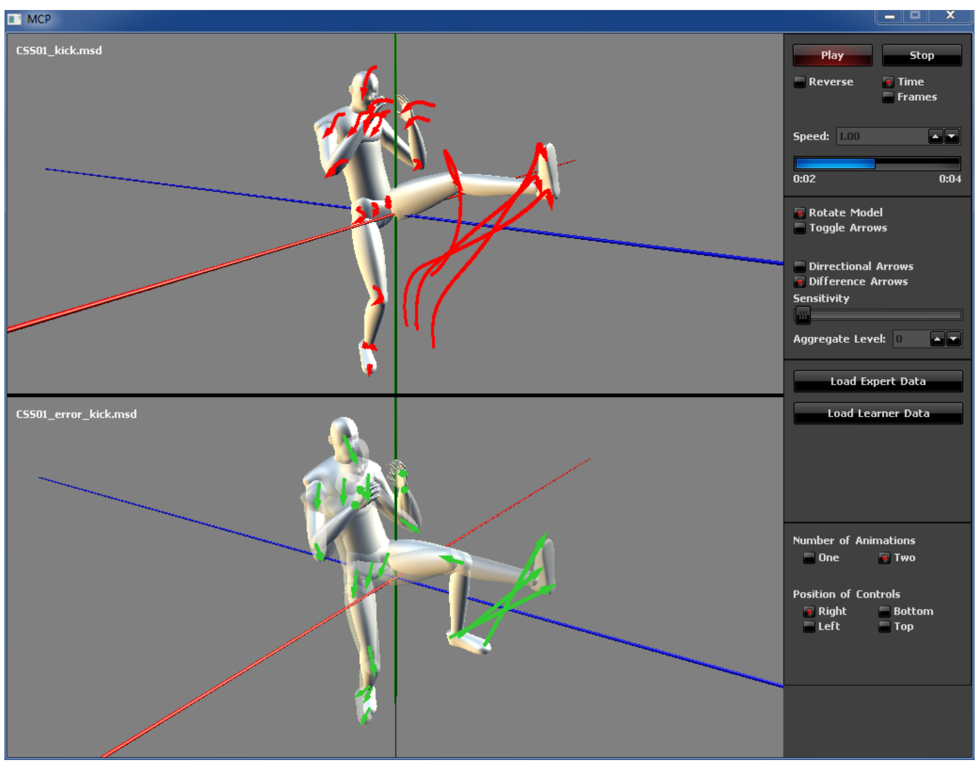

Figure 5.2: A motion capture comparison visualizing spatial differences only.

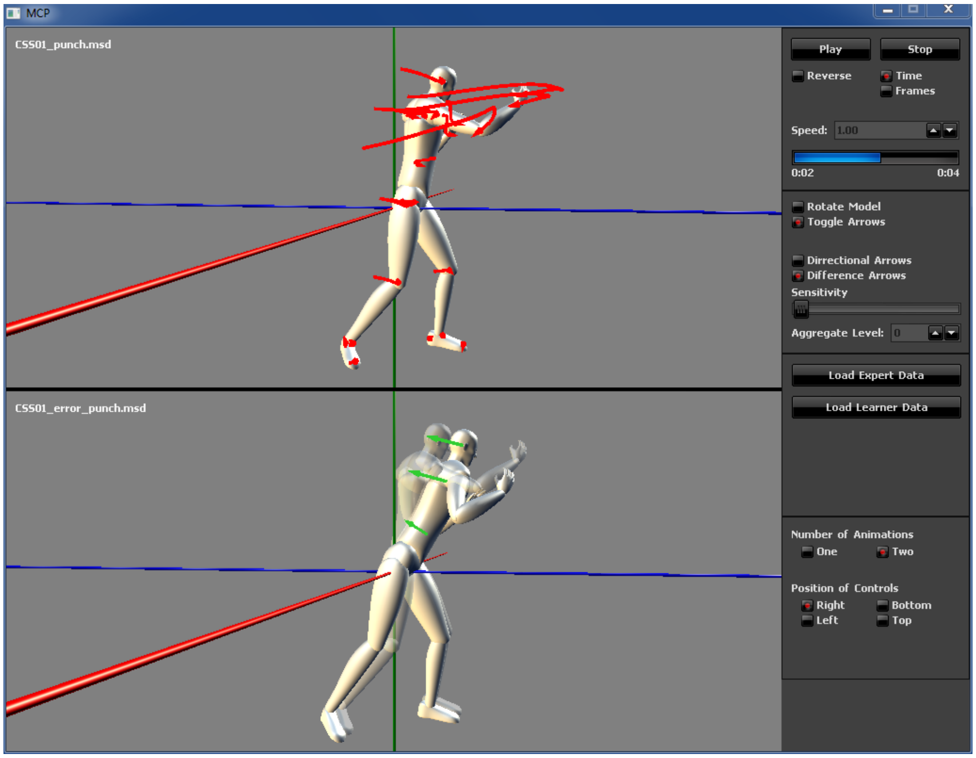

Figure 5.3: A motion capture comparison visualizing spatial differences with some arrows toggled off.

Additionally, we incorporated three particular techniques: ghosting, aggregation, and sensitivity adjustment. We explain each in turn.

5.1.1 Ghosting

Initially, we visualized spatial differences using only difference arrows. This proved to be somewhat less than intuitive as the arrows would extend from a point associated with the learner avatar out into space. While we did not formally test learners' understanding of the arrows, we assume a learner would have to have explained to them what the arrow's meaning was. As we have attempted to devise our system to communicate as intuitively as possible in as many cases as possible, this seemed less than satisfactory. We then decided to include in the learner animation a ghosted rendering of the expert since this is what the spatial comparison is done with. By adding this feature, we are able to provide the viewer with an additional reference point so that the meaning of the difference arrows is more immediately intuitive.

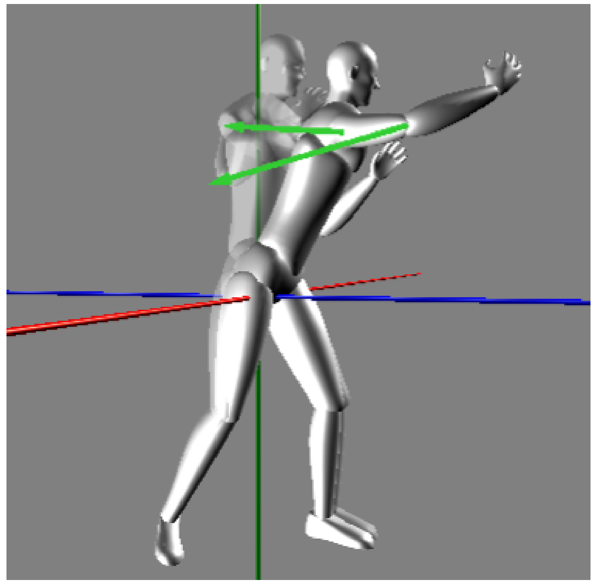

Figure 5.4: A close-up of the learner portion of the window showing the ghosting effect.

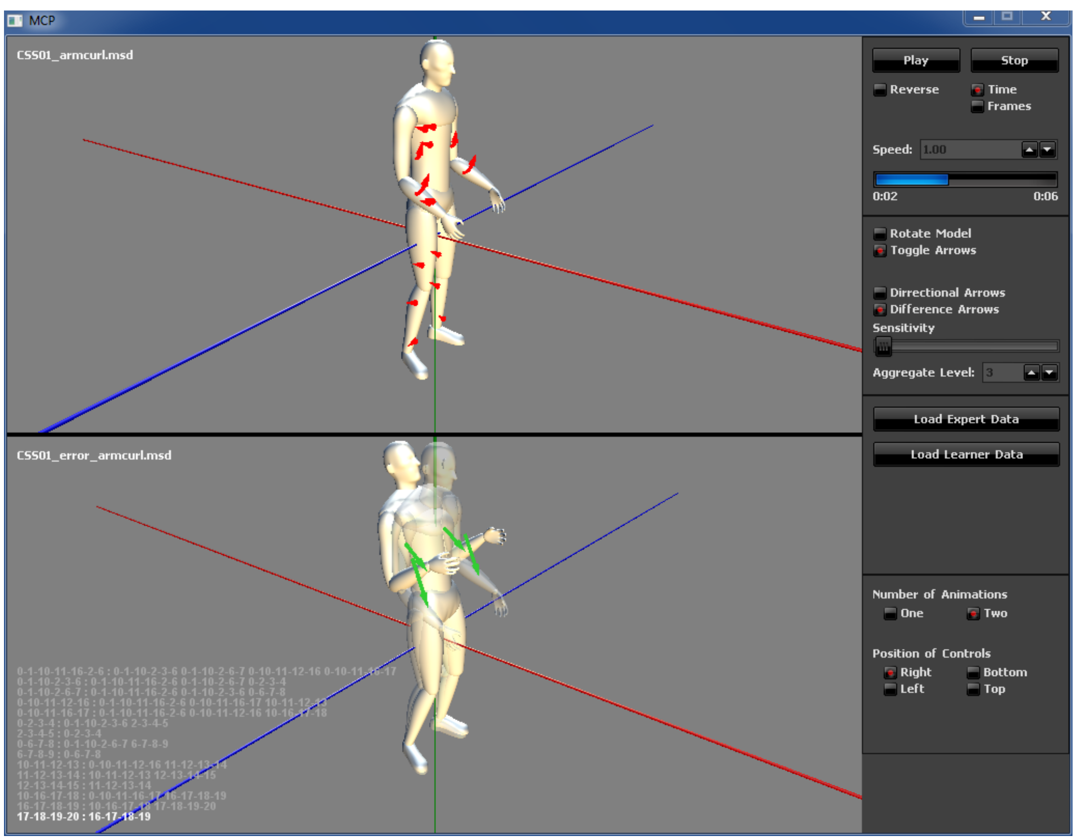

Figure 5.5: A motion capture comparison visualizing spatial differences using the aggregation feature.

5.1.2 Aggregated Movements

Another feature that can be applied to the arrows in the visualization is aggregation. The default animation might provide too much information for the viewer. For example, a learner may not want to see the motion path of each joint from their shoulder to their hand but perhaps would like to see the general path their arm is following. This is what the aggregation feature does. By performing weighted averages of joint center locations, the system approximates paths followed by limbs instead of joint centers. One level of aggregation of, say, the elbow and wrist gives the approximate path of the forearm. An additional level will average the three main joint centers of the arm to provide an approximate path of the arm as a whole. By aggregating the trajectories of several markers the system gives the user a more general sense of the problem. Figure 5.5 shows an example of this type of visualization.

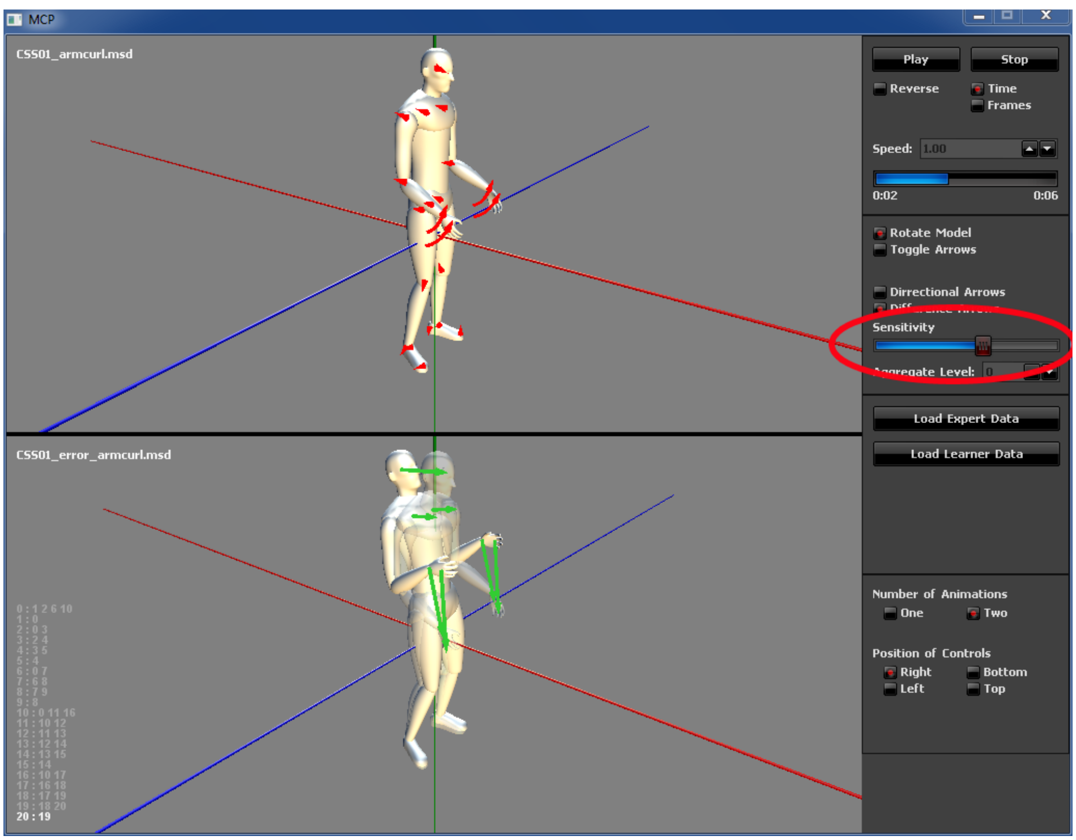

Figure 5.6: A motion capture comparison visualizing spatial differences and using the sensitivity threshold.

5.1.3 Sensitivity

Slight but consistent errors can generate frequent flashes of arrow movement that makes it difficult for the user to see the real problem - similar to the situation in computer programming when a single missing semi-colon generates hundreds of error messages. By adjusting the sensitivity, only differences of a certain magnitude are displayed. As the user becomes more skilled, the user can focus on smaller details. By moving the slider to the right, the learner can exclude smaller differences by increasing the threshold. Figure 5.6 shows the resulting visualization after this threshold has been increased.

5.2 Spatio-Temporal Comparisons

The following figures show visualizations of spatio-temporal comparisons. One of the difficulties in visualizing something of a temporal nature was deciding on an appropriate visual cue that would be separate from the arrows used for spatial differences but immediately intuitive to the user as a representation of temporal differences. The figures show how coloring particular body parts was used to convey the messages *too slow* or *too fast* since these are the principle messages necessary. As the animation proceeds, different body parts are colored either red or blue to note that they are moving too slowly (blue) or too quickly (red) at that juncture of the motion. Figure 5.10 shows a learner whom is moving too slowly in certain body parts. The learner in figure 5.8, however, is uniformly moving too quickly.

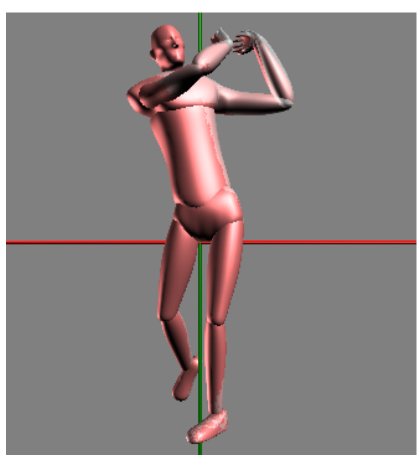

Figure 5.7: An actor shaded in varying levels of red to note different temporal differences.

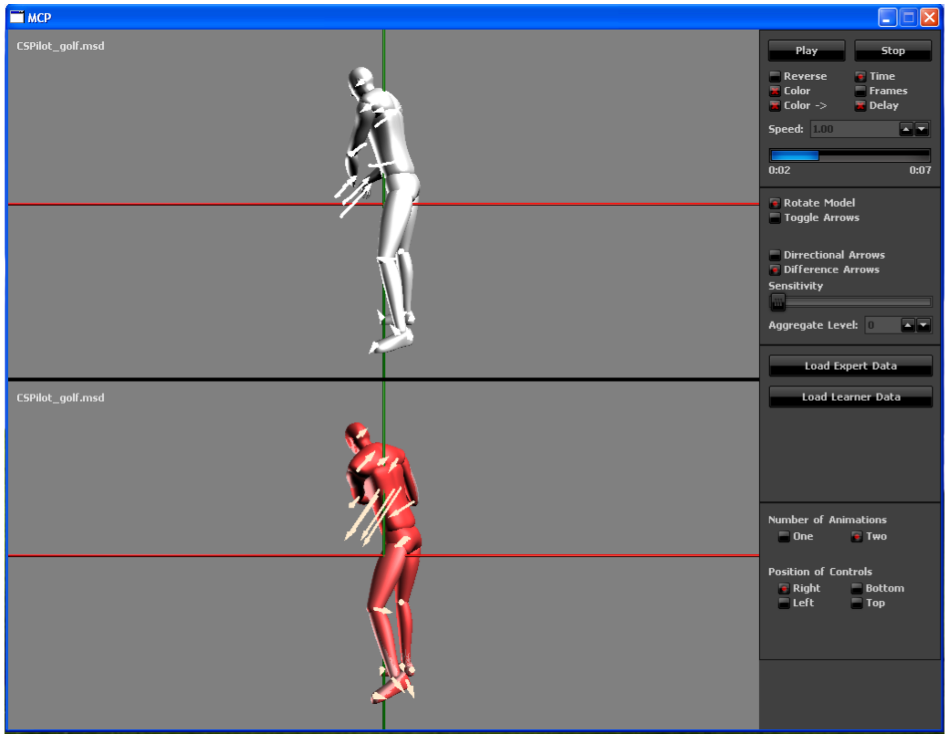

Figure 5.8: A spatio-temporal comparison where the learner is moving too quickly.

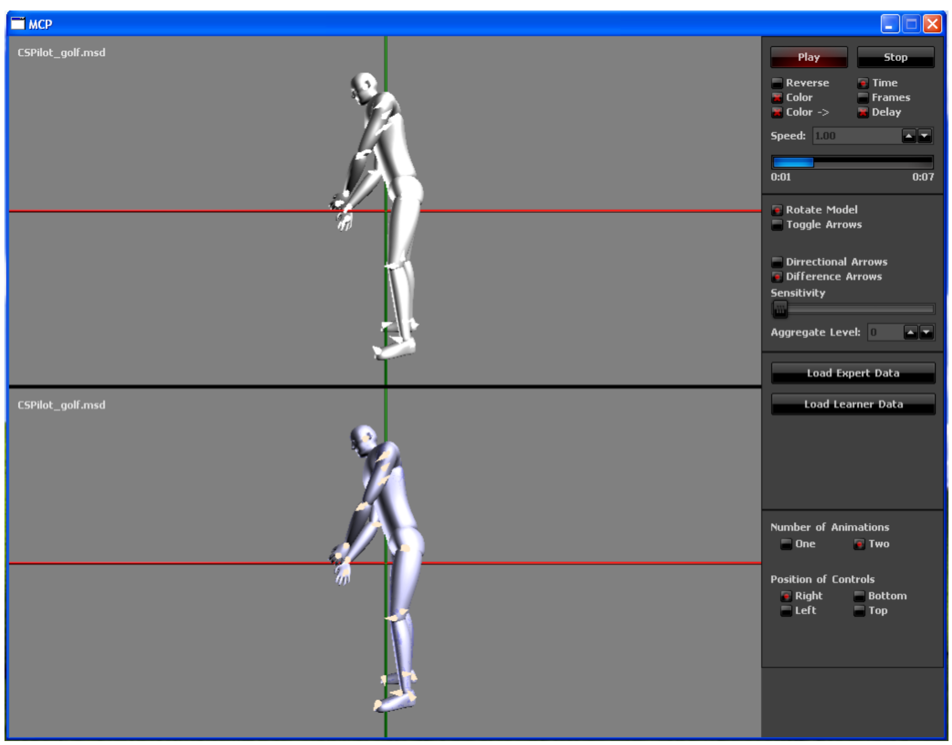

Figure 5.9: A spatio-temporal comparison where the learner is moving slightly slowly.

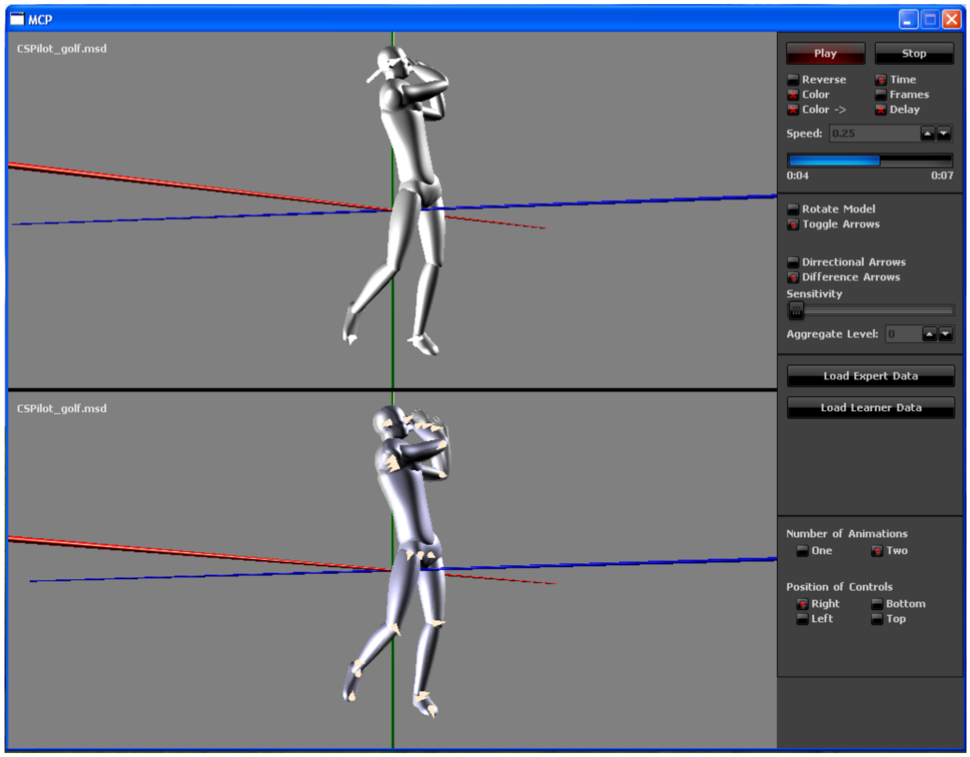

Figure 5.10: A spatio-temporal comparison where the learner is moving too slowly in certain parts, particularly torso and upper legs.

5.3 Discussion

In this work we set out to show proof of concept for the use of motion capture technology in coaching. We did so with certain constraints to ensure the problem scope would remain manageable. We now make direct comparisons between our results and two previous works in this area, namely Bouvier-Zappa[20][21] and Maekawa[103].

As noted previously, the work of Bouvier-Zappa, while important, was lacking an important feature for pedagogy: feedback for the learner. The figures below show the difference between our system and the directive visual feedback it produces compared to the work of Bouvier-Zappa where visual cues are used to communicate what is happening in an individual motion.

We can also contrast our results with those of Maekawa. Here we see an attempt at comparison but there is no processing of the motion capture data to perform the comparison. The comparison is left as a visual exercise for the user and an expert. The authors note regarding the balloons and points they correspond to that “These points are based

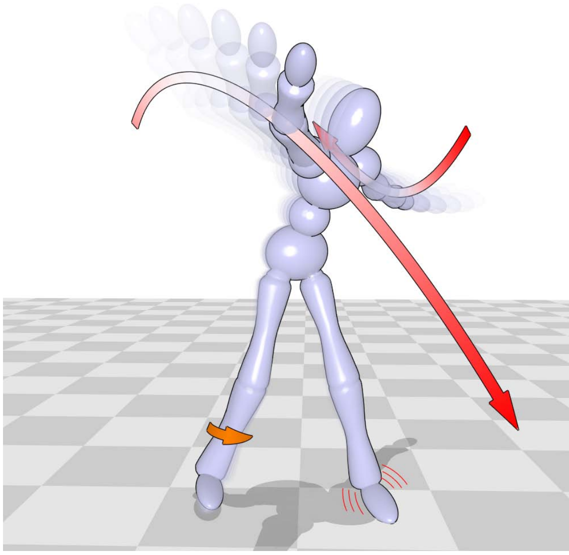

Figure 5.11: Results of Bouvier-Zappa.

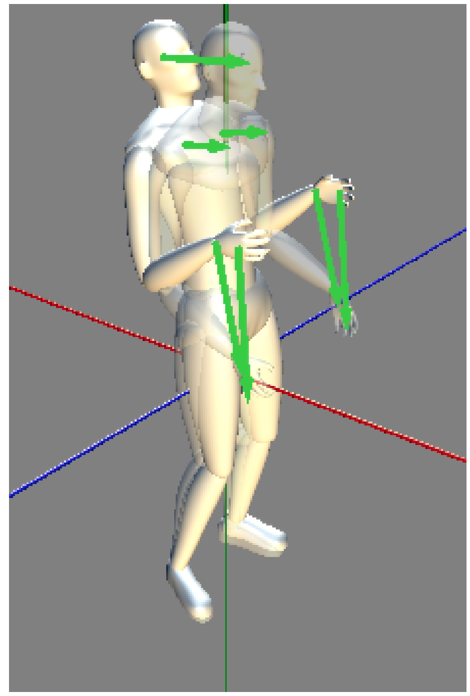

Figure 5.12: Results from our system showing directive feedback.

on the experts opinion." Our system, by contrast, automatically finds differences both spatially and temporally by performing comparisons on the underlying data and visualizes these differences to provide directive feedback to the learner.

As stated above, we imposed certain constraints to maintain manageability of the scope of the work. We now discuss each of those constraints in turn.

The first constraint we imposed was to assume a known start and end for each motion curve. This eliminated the need for the system to perform additional processing to find these points as this is a very difficult problem unto itself.

The next constraints we imposed were with regards to the actors and what motions we would compare. For each comparison, we always compared an actor to themselves. This was done for a few reasons. By doing the comparison this way, we eliminate anomalies that might arise from comparing normalized data from individuals with very different physiques but we also can determine that the differences we have detected are the actual spatial differences we want to detect as opposed to possibly detecting noise. This was done in the case of the first three motions for each actor by having the actor repeat each motion with a specific predetermined error. In the case of our temporal processing, we used

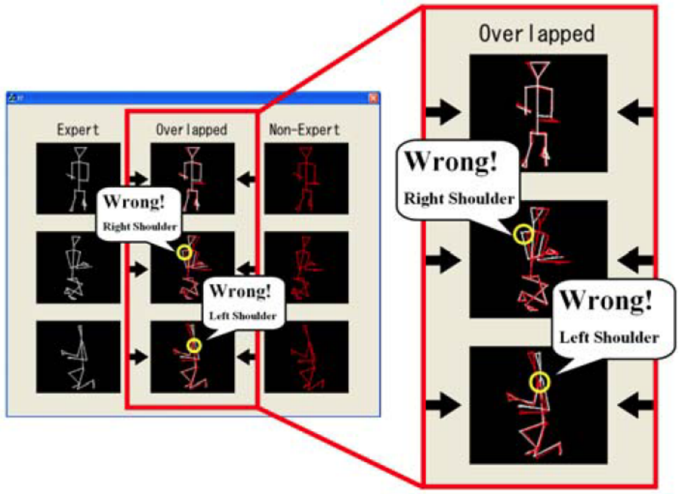

Figure 5.13: Results of Maekawa.

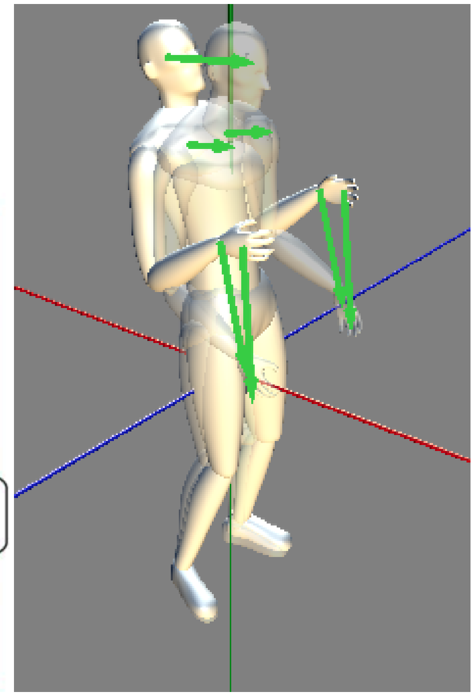

Figure 5.14: Results from our system showing directive feedback.

the golf swing and developed our software to allow us to specify a parameter file which would include parameters for how we might want to slow down or speed up portions of a learner motion. Again, while we are eliminating noise, we also can determine that what is detected is temporal difference and we can see that the system is not only detecting it but also visualizing it appropriately.

Another constraint was the use of *(x,y,z)* format data rather than velocity based data. The use of this format and the process of splitting it into individual signals provided the inspiration to use the segmentation and matching methods we used for our temporal comparison and made spatial comparison and visualization very straightforward. This is not to say that velocity based data is without its benefits. For example, all joints are effectively afforded a full six degrees of freedom using the (x,y,z) format but some joint movement, like at the elbow, can be expressed with as few as one or two.

Another consideration was to avoid engaging in an evaluation process of the pedagogical quality of our particular form of feedback. There is importance in asking the question “Does this type of feedback contribute positively to a learner’s improvement?” Answering this question can involve a great deal of work unto itself. We would likely do so by performing a study of various learners by measuring their motion before and after using our system and other methods to try to improve. In this manner, we could show empirically that our system helps to produce a certain improvement compared to other methods. We leave this type of analysis to future work.

Chapter 6

Conclusions and Future Work

The purpose of this research was to show *proof of concept* for a system that could analyze two motion capture datasets and visualize the differences between them. We believe that such a system constitutes a useful training tool to a learner but would also be an important component in a larger coaching system.

We have demonstrated the feasibility of producing such a system. Our system successfully analyzes two datasets and visualizes the differences in real time. We are able to perform strictly spatial comparisons as well as temporal comparisons. The result is a simple system which produces useful visualizations that may aid a learner.

Our research also presents numerous opportunities for future work. While we have shown feasibility, we did not purport to produce the best result possible nor did we assume that our solution would be the final word on this type of system. There is room for improvement and, therefore, further research in most areas of this system.

The potential for future work can be grouped into two categories: internal function and pedagogical performance. Some aspects of the system can be further analyzed and possibly improved to produce better technical or performance results while others could be analyzed in terms of how effective they are for the learner or coach using the system.

6.1 Internal and Functional Improvements

An initial area of further work is the parameterization method used. Splines are a very well known method in computer science but that is not to say they are necessarily the only or best solution. There are other methods that can be used which might be more optimal in terms of processing cost, especially if the size of the dataset and the number of markers were to increase substantially. Our models use relatively small amounts of data. As motion capture systems advance, the possibility of systems to be able to capture very large numbers of markers will become more likely. It is difficult to predict how our method would scale if we were to increase our number of markers or frames by, say, an order of magnitude. It would be interesting to formally test different methods using varying dataset sizes and marker counts with specific performance metrics to determine optimal performance while scaling to potentially very large data sizes. Hamilton et al.[66], for example, discuss a method of detection for QRS patterns(named for the Q, R and S waves in an ECG) in encephalocardiograms (ECGs). QRS patterns are a series of deflections found in a typical ECG. They begin by applying one optimized detection rule and follow this with additional successive rules. The authors note that their algorithm was efficient enough to be executed in real time.

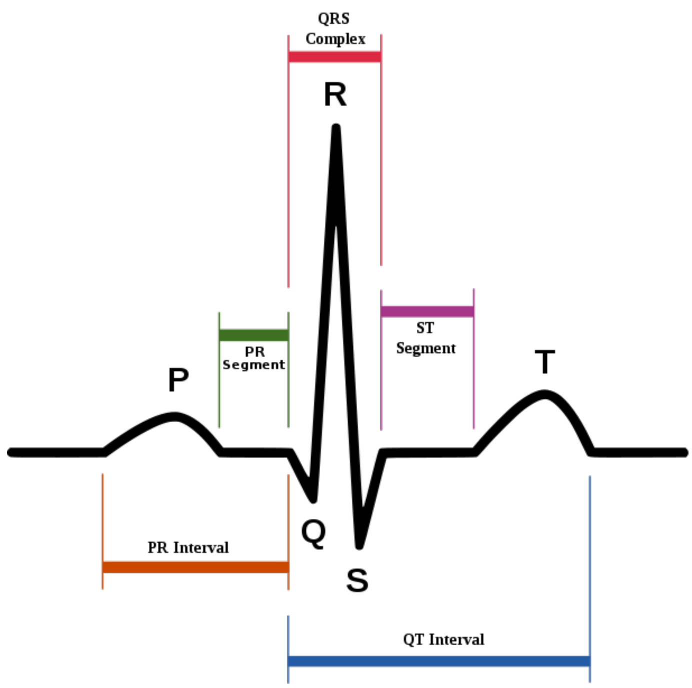

Figure 6.1: An encephalocardiogram with QRS complex labeled.

Ramsay and Silverman[128] discuss different methods of function representation using basis functions such as splines, Fourier analysis and wavelets. The authors also describe different approaches to curve registration, that is, aligning two curves that are to be compared. They describe using least squares, feature registration and warping functions for this purpose.

Signal processing is another potential area that could be applied. Interestingly, there are signal processing methods such as autocorrelation which allow for the detection of a small signal pattern within a larger one. This could have some potential application since our processing method effectively reduces motion data into a collection of 1D signals and there is some body of work already involving the use signal processing techniques with motion[28].

It would also be interesting to test different motion capture system types, especially inertial types, to see if a comparable enough level of detail can be provided. We know that large scale lab-based setups produce highly accurate data but there may be some question as to whether this level of accuracy is necessary. While it might be difficult to gain access to both types of systems for the same study, if it can be shown that inertial systems produce data which is also of high accuracy, this could show that this type of system is potentially very accessible since this type of hardware is becoming more affordable and available to average consumers.

A second area of future work that could be pursued is the addition of intelligence to the system. Our system is able to visualize differences but it doesn't perform real intelligent decision making. The system could be amended to include intelligent comparisons of stored "common mistakes" to learner motion to determine whether a learner is making a common error. With most forms of athletic endeavour, there are a set of mistakes that are common to beginners and novices. We believe that the motion curves of such mistakes would have unique "signatures" in the form of a sequence of inflection points. These sequences can be encoded as strings that one can compare using techniques similar to those already described within here, and further there are well-known algorithms for sorting out DNA sequences that could be easily used to this effect. Being able to catalogue these signatures would be useful in that it would eliminate the requirement for an expert and so make the system more flexible for the user. This would involve the use of a "bug library" which would need to be compiled which would require considerable effort by way of getting experts to produce the bug library, and by testing these. Such work would require considerable empirical investment. Niu[117] and Rivera[134] both point to the work of Burton[29] in proposing a BUGGY model similar to that described above.

Another potential area of investigation regards the way in which we executed our temporal segmentation. Our method proved to be effective within this context but it lacked any semantics in the divisions. Our method picked critical points in motion curves for segmentation but this is somewhat mechanical. In a longer motion sequence like, for example, an interpretive dance that might last several minutes would contain a larger number of particular gestures that would not be detected using our method. A method of segmentation similar to [10] that uses principle component analysis(PCA) to segment motion into *behaviours* could be used instead. This would be applicable for any type of motion which consists of a longer sequence rather than a short discrete movement. There might also be some argument for a combination of these two types of methods. One method which is semantically based for macro-segmentation coupled with another method similar to ours which is perhaps more suitable for a more mathematical micro-segmentation.

An additional issue when working with motion capture data is the time necessary to process the data after it has been recorded. Ultimately, we would want a system which would work completely in real time which would necessitate data from a motion capture system going directly to a system like ours. This could require substantial work to produce a digital interface which could allow the software to work with any motion capture system directly.

Another constraint in allowing the data to proceed directly from the collection system to our software system is the necessity to manually truncate the data at the beginning and end to ensure that the start and end line up. If work could be done to provide a solution for this problem, it would be an important step towards being able to do everything in real time. As it is now, data is recorded but then requires pre-processing to clean and adjust it because of some of the issues above. Ideally, we would want to be able to have a user put on a motion capture suit or some such portable system, turn on our software and immediately begin getting feedback.

6.2 User and Pedagogical Improvements

The first area of possible work is the graphical user interface of the system. This interface was developed based on our own perceptions of what constituted an intuitive interface. A researcher in the area of human computer interaction would be better placed to evaluate our interface or comment on alternatives for evaluation. Additionally, the use of direct feedback from potential users would be invaluable to develop the visual interface in such a way as to maximize its effectiveness and intuitiveness for learners. This is an important factor for a system whose primary purpose is projecting pedagogical information to a learner.

A second area of further work relates to the type of directive feedback we chose to employ in our system. We chose arrows primarily for their history of use and demonstrated effectiveness in other areas. However, it would be interesting to implement the same system with other types of cues for the same visualizations and run a test survey with a group of learners to explicitly test if these methods produce the best end result pedagogically. As was noted by Jones and Scaife[80], a learner may enjoy working with a particular system but that doesn't automatically imply better results. It would also be useful to obtain some form of empirical validation of such a comparison. This might involve enhancing the system to measure the magnitude of differences it detects. Then one could, iteratively, measure a user's performance and note any improvements and the magnitude of improvements over time as means of comparing effectiveness between different forms of visualization.

A third potential idea for future research relates to the type of directive feedback currently provided and whether it would provide a benefit to be more explicit in the content of the feedback. That is, instead of showing differences or detecting the presence of errors, to actually inform the learner explicitly of what they *should* do rather than simply displaying what they are doing incorrectly. This might involve verbal directives from the system or might be accomplished by some other visual means or combination of the two.

References

[1] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. *SIGGRAPH '04: SIGGRAPH 2004 Papers*, Aug 2004.

[2] Maneesh Agrawala, Doantam Phan, Julie Heiser, John Haymaker, Jeff Klingner, Pat Hanrahan, and Barbara Tversky. Designing effective sstep-by-step assembly instructions. *SIGGRAPH '03: SIGGRAPH 2003 Papers*, Jul 2003.

[3] Willi Apel. *The Notation Of Polyphonic Music 900-1600.* The Mediaeval Academy Of America, 1949.

[4] Daniel Archambault, Tamara Munzner, and David Auber. Topolayout: Multilevel graph layout by topological features. *IEEE Transactions on Visualization and Computer Graphics*, pages 305–317, 2007.

[5] Okan Arikan and David Forsyth. Interactive motion generation from examples. *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, Jul 2002.

[6] Okan Arikan, David Forsyth, and James O'Brien. Motion synthesis from annotations. *SIGGRAPH '03: SIGGRAPH 2003 Papers*, Jul 2003.

[7] Jackie Assa, Yaron Caspi, and Daniel Cohen-Or. Action synopsis: Pose selection and illustration. *SIGGRAPH '05: SIGGRAPH 2005 Papers*, Jul 2005.

[8] Jackie Assa, Daniel Cohen-Or, I-Cheng Yeh, and Tong-Yee Lee. Motion overview of human actions. *SIGGRAPH Asia '08: SIGGRAPH Asia 2008 papers*, Dec 2008.

[9] Grégoire Aujay, Franck Hétroy, Francis Lazarus, and Christine Depraz. Harmonic skeleton for realistic character animation. *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Aug 2007.

[10] Jernej Barbič, Alla Safonova, Jia-Yu Pan, Christos Faloutsos, Jessica K. Hodgins, and Nancy S. Pollard. Segmenting motion capture data into distinct behaviors. *Proceedings of Graphics Interface 2004*, pages 185–194, 2004.

[11] Scott Bateman, Regan L. Mandryk, Carl Gutwin, Aaron Genest, David McDine, and Christopher Brooks. Useful junk?: The effects of visual embellishment on comprehension and memorability of charts. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems*, CHI '10, pages 2573–2582, New York, NY, USA, 2010. ACM.

[12] Jon Bath. *Blowing the Crystal Goblet: Transparent Book Design 1350-1950.* PhD thesis, University of Saskatchewan, 2009.

[13] Philippe Beaudoin, Stelian Coros, Michiel van de Panne, and Pierre Poulin. Motion-motif graphs. *SCA '08: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Jul 2008.

[14] Philippe Beaudoin, Michiael van de Panne, and Pierre Poulin. Technical report 1296 automatic construction of compact motion graphs.

[15] Philippe Beaudoin, Michiael van de Panne, and Pierre Poulin. Adapting wavelet compression to human motion capture clips. *Proceedings of Graphics Interface 2007*, pages 313–318, 2007.

[16] Philippe Beaudoin, Michiael van de Panne, and Pierre Poulin. Automatic construction of compact motion graphs. Technical report, UBC, 2007.

[17] Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps.* ESRI Press, 2010.

[18] Thomas Beth, Ingo Boesnach, Martin Haimerl, Jörg Moldenhauer, Klaus Bös, and Veit Wank. Characteristics in human motion–from acquisition to analysis. *International Conference on Humanoid Robots (Humanoids), Karlsruhe/München, Germany*, 2003.

[19] Volker Blanz, Michael J Tarr, and Heinrich H Bülthoff. What object attributes determine canonical views? *Perception*, 28(5):575–99, Jan 1999.

[20] Simon Bouvier-Zappa. Illustration non-photorealiste de sequences de captures de mouvements. Master's thesis, Nov 2008.

[21] Simon Bouvier-Zappa, Victor Ostromoukhov, and Pierre Poulin. Motion cues for illustration of skeletal motion capture data. *NPAR '07: Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, Aug 2007.

[22] Niels Olof Bouvin, Polle T. Zellweger, Kaj Gronbaek, and Jock D. Mackinlay. Fluid annotations through open hypermedia: Using and extending emerging web standards. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 160–171, New York, NY, USA, 2002. ACM.

[23] Matthew Brand. Pattern discovery via entropy minimization. *Artificial Intelligence and Statistics*, 7, 1998.

[24] Matthew Brand and Aaron Hertzmann. Style machines. *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, Jul 2000.

[25] Christoph Bregler, Lorie Loeb, Erika Chuang, and Hrishi Deshpande. Turning to the masters: Motion capturing cartoons. *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, Jul 2002.

[26] Christoph Bregler and Jitendra Malik. Video motion capture. *Proceeding of SIGGRAPH*, 98, 1998.

[27] Gabriel Brostow and Irfan Essa. Image-based motion blur for stop motion animation. *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, Aug 2001.

[28] Armin Bruderlin and Lance Williams. Motion signal processing. *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, Sep 1995.

[29] Richard R. Burton. *Diagnosing Bugs in a Simple Procedural Skill.* New York: Academic Press, 1982.

[30] IW Charlton, P Tate, P Smyth, and L Roren. Repeatability of an optimised lower body model. *Gait & posture*, 20(2):213–221, 2004.

[31] Siddhartha Chattopadhyay, Suchendra M Bhandarkar, and Kang Li. Human motion capture data compression by model-based indexing: A power aware approach. *Visualization and Computer Graphics, IEEE Transactions on DOI - 10.1109/TVCG.2007.13*, 13(1):5–14, 2007.

[32] Peter C-H. Cheng. Functional roles for the cognitive analysis of diagrams in problem solving. *Proceedings of the eighteenth annual conference of the Cognitive Science Society: July 12-15, 1996, University of California, San Diego*, page 207, 1996.

[33] Peter C-H. Cheng, Rick Lowe, and Mike Scaife. Cognitive science approaches to understanding diagrammatic representations. *Artif. Intell. Rev.*, 15(1-2):79–94, 2001.

[34] Stephen Chenney, Mark Pingel, Rob Iverson, and Marcin Szymanski. Simulating cartoon style animation. *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, Jun 2002.

[35] Wallace Chigona and Thomas Strothotte. Contextualized text explanations for visualizations. *SMARTGRAPH '02: Proceedings of the 2nd international symposium on Smart graphics*, Jun 2002.

[36] Paul Chippendale. Towards automatic body language annotation. *FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, Apr 2006.

[37] Kwangsu Cho, Christian D. Schunn, and Davida Charney. Commenting on writing: Typology and perceived helpfulness of comments from novice peer reviewers and subject matter experts. *Written Communication*, 23(3):260–294, 2006.

[38] Min Gyu Choi, Jehee Lee, and Sung Yong Shin. Planning biped locomotion using motion capture data and probabilistic roadmaps. *ACM Trans. Graph.*, 22(2):182–203, 2003.

[39] William S. Cleveland. *Visualizing Data.* Hobart Press, 1993.

[40] John Collomosse and Peter Hall. Motion analysis in video: Dolls, dynamic cues and modern art. *Proc. International Conference on Vision, Video and Graphics*, pages 109–116, 2005.

[41] John Collomosse and Peter Hall. Video motion analysis for the synthesis of dynamic cues and futurist art. *Graphical Models*, 68(5), Sep 2006.

[42] John Collomosse, Dave Rowntree, and Peter Hall. Rendering cartoon-style motion cues in post-production video. *Graphical Models*, 67(6):549–564, 2005.

[43] Seth Cooper, Aaron Hertzmann, and Zoran Popović. Active learning for real-time motion controllers. *SIGGRAPH '07: SIGGRAPH 2007 papers*, Aug 2007.

[44] Stelian Coros, Philippe Beaudoin, Kang Yin, and Michiel Pann. Synthesis of constrained walking skills. *SIGGRAPH Asia '08: SIGGRAPH Asia 2008 papers*, Dec 2008.

[45] James E Cutting. Representing motion in a static image: Constraints and parallels in art, science, and popular culture. *Perception*, 31(10):1165–93, Jan 2002.

[46] Brad deGraf and Emeer Yilmaz. Puppetology: Science or cult? Animation World Magazine, February 1999.

[47] Robert Dony, John Mateer, and John Robinson. Techniques for automated reverse storyboarding. *Vision, Image and Signal Processing, IEEE Proceedings*, 152(4):425–436, 2005.

[48] Henry Dreyfuss. *Symbol Sourcebook: An Authoritative Guide to International Graphic Symbols*. John Wiley and Sons, 1984.

[49] Rainald M Ehrig, William R Taylor, Georg N Duda, and Markus O Heller. A survey of formal methods for determining functional joint axes. *J Biomech*, 40(10):2150–7, Jan 2007.

[50] Diarmaid Fitzgerald, John Foody, Dan Kelly, Tomas Ward, Charles Markham, John McDonald, and Brian Caulfield. Development of a wearable motion capture suit and virtual reality biofeedback system for the instruction and analysis of sports rehabilitation exercises. *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 4870–4874, 2007.

[51] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley Systems Programming Series. Addison-Wesley, 2nd edition, 1997.

[52] David A. Forsyth, Okan Arikan, and James F. O'Brien. Motion synthesis from annotations. *ACM transactions on graphics*, 22(3):402, 2003.

[53] Chi-Wing Fu and Andrew J. Hanson. A transparently scalable visualization architecture for exploring the universe. *Visualization and Computer Graphics, IEEE Transactions on DOI - 10.1109/TVCG.2007.2*, 13(1):108–121, 2007.

[54] Pascal Glardon, Ronan Boulic, and Daniel Thalmann. Robust on-line adaptive footplant detection and enforcement for locomotion. *Visual Computer*, 22(3):194–209, 2006.

[55] Michael Gleicher. Retargetting motion to new characters. *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, Jul 1998.

[56] Michael Gleicher. Motion path editing. In *I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 195–202, 2001.

[57] Michael Gleicher. Graph-based motion synthesis: An annotated bibliography. *ACM SIGGRAPH 2008 classes*, Jan 2008.

[58] Michael Gleicher and Nicola Ferrier. Evaluating video-based motion capture. *CA '02: Proceedings of the Computer Animation*, Jun 2002.

[59] Frank Godenschweger and Thomas Strothotte. Modeling and generating sign language as animated line drawings. *Assets '98: Proceedings of the third international ACM conference on Assistive technologies*, Jan 1998.

[60] Dan Goldman, Brian Curless, David Salesin, and Steven Seitz. Schematic storyboarding for video visualization and editing. *SIGGRAPH '06: SIGGRAPH 2006 Papers*, Jul 2006.

[61] Ernst Gombrich. *Images and Understanding*, pages 26–45. Cambridge University Press, 1990.

[62] John Gould, Lizette Alfaro, Rich Finn, Brian Haupt, and Angela Minuto. Why reading was slower from crt displays than from paper. *CHI '87: Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface*, Apr 1987.

[63] Henri Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, 20(6), Jun 1971.

[64] Carl Gutwin. Traces: Visualizing the immediate past to support group interaction. In *Proc. Graphics Interface*, pages 43–50, May 2002.

[65] Michael Haller, Christian Hanl, and Jeremiah Diephuis. Non-photorealistic rendering techniques for motion in computer games. *Computers in Entertainment*, 2(4), Oct 2004.

[66] Patrick S. Hamilton and Willis J. Tompkins. Quantitative investigation of qrs detection rules using the mit/bih arrhythmia database. *Biomedical Engineering, IEEE Transactions on*, BME-33(12):1157 –1165, dec. 1986.

[67] Rachel Heck and Michael Gleicher. Parametric motion graphs. *I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games*, Apr 2007.

[68] Chris Hecker, Bernd Raabe, Ryan Enslow, John DeWeese, Jordan Maynard, and Kees Prooijen. Real-time motion retargeting to highly varied user-created morphologies. *SIGGRAPH '08: SIGGRAPH 2008 papers*, Aug 2008.

[69] Julie Heiser, Doantam Phan, Maneesh Agrawala, Barbara Tversky, and Pat Hanrahan. Identification and validation of cognitive design principles for automated generation of assembly instructions. *Proceedings of the working conference on Advanced Visual Interfaces*, pages 311–319, 2004.

[70] Julie Heiser and Barbara Tversky. Arrows in comprehending and producing mechanical diagrams. *Cognitive Science*, 30(3):581–592, 2006.

[71] Robert E. Horn. *Visual Language*. MacroVu, Inc., 1998.

[72] Siu Hsu and Irene Lee. Drawing and animation using skeletal strokes. *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, Jul 1994.

[73] John E. Hummel and Irving Biederman. Dynamic binding in a neural network for shape recognition. *Psychological Review*, 99(3):480–517, 1992.

[74] Antoni Jaume i Capó, Javier Varona, Manuel González-Hidalgo, Ramon Mas, and Francisco J. Perales. Automatic human body modeling for vision-based motion capture. *In Short Communication Proceedings of the 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'06), Plzen, Czech Republic*, 2006.

[75] Pourang Irani and Colin Ware. Diagrams based on structural object perception. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '00, pages 61–67, New York, NY, USA, 2000. ACM.

[76] Tobias Isenberg, Maic Masuch, and Thomas Strothotte. 3d illustrative effects for animating line drawings. *Information Visualization, 2000. Proceedings. IEEE International Conference on*, pages 413–418, 2002.

[77] Doug L. James, Christopher D. Twigg, Andrew Cove, and Robert Y. Wang. Mesh ensemble motion graphs: Data-driven mesh animation with constraints. *ACM Transactions on Graphics (TOG)*, 26(4):17, 2007.

[78] Alexander R. Jensenius, Tellef Kvifte, and Rolf Inge Godøy. Towards a gesture description interchange format. *Proceedings of the 2006 conference on New interfaces for musical expression*, pages 176–179, 2006.

[79] Chao Jin, Thomas Fevens, Shuo Li, and Sudhir Mudur. Motion learning-based framework for unarticulated shape animation. *The Visual Computer: International Journal of Computer Graphics*, 23(9), Aug 2007.

[80] Sara Jones and Mike Scaife. Animated diagrams: An investigation into the cognitive effects of using animation to illustrate dynamic processes. *Theory and Application of Diagrams*, pages 295–307, 2000.

[81] Alark Joshi and Penny Rheingans. Illustration-inspired techniques for visualizing time-varying data. *Visualization, 2005. VIS 05. IEEE*, pages 679–686, 2005.

[82] Kanav Kahol, Priyamvada Tripathi, and Sethuraman Panchanathan. Documenting motion sequences with a personalized annotation system. *Multimedia, IEEE*, 13(1):37 – 45, 2006.

[83] Kanav Kahol, Priyamvada Tripathi, and Sethuraman Panchanathan. Documenting motion sequences with a personalized annotation system. *Multimedia, IEEE*, 13(1):37–45, 2006.

[84] Minoru Kanehisa. Sequence comparison to graph comparison - a new generation of algorithms for network analysis of interacting molecules. *RECOMB '00: Proceedings of the fourth annual international conference on Computational molecular biology*, Apr 2000.

[85] Nikhil S. Ketkar, Lawrence B. Holder, and Diane J. Cook. Comparison of graph-based and logic-based multi-relational data mining. *ACM SIGKDD Explorations Newsletter*, 7(2):64–71, 2005.

[86] M Kipp, M Neff, and I Albrecht. An annotation scheme for conversational gestures: How to economically capture timing and form. *Language Resources and Evaluation*, 41(3):325–339, 2007.

[87] Scott Kircher and Michael Garland. Free-form motion processing. *ACM Transactions on Graphics*, 27(2), 2008.

[88] Reinhard Klette and Garry Tee. Understanding human motion: A historic review. 2007.

[89] Lucas Kovar and Michael Gleicher. Flexible automatic motion blending with registration curves. *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Jul 2003.

[90] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, Jul 2002.

[91] Taesoo Kwon, Kang Lee, Jehee Lee, and Shigeo Takahashi. Group motion editing. *SIGGRAPH '08: SIGGRAPH 2008 papers*, Aug 2008.

[92] Taesoo Kwon and Sung Shin. Motion modeling for on-line locomotion synthesis. *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Jul 2005.

[93] Jehee Lee, Jinxiang Chai, Paul Reitsma, Jessica Hodgins, and Nancy Pollard. Interactive control of avatars animated with human motion data. *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, Jul 2002.

[94] Paul Lee, Alexander Klippel, and Heike Tappe. The effect of motion in graphical user interfaces. *SG'03: Proceedings of the 3rd international conference on Smart graphics*, Jul 2003.

[95] Qi Li, Weidong Geng, Tao Yu, Xiaojie Shen, Newman Lau, and Gino Yu. Motionmaster: Authoring and choreographing kung-fu motions by sketch drawings. *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Sep 2006.

[96] Yan Li, Tianshu Wang, and Heung-Yeung Shum. Motion texture: A two-level statistical model for character motion synthesis. *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, Jul 2002.

[97] Yin Li, Michael Gleicher, Ying Qing Xu, and Heung-Yeung Shum. Stylizing motion with drawings. *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 309–319, 2003.

[98] Yi Lin. Efficient human motion retrieval in large databases. *GRAPHITE '06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, Nov 2006.

[99] Ce Liu, Antonio Torralba, William Freeman, Frédo Durand, and Edward Adelson. Motion magnification. *SIGGRAPH '05: SIGGRAPH 2005 Papers*, Jul 2005.

[100] Guodong Liu and Leonard McMillan. Segment-based human motion compression. *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Sep 2006.

[101] Xiaoming Liu, Yueting Zhuang, Yi Wu, and Yunhe Pan. Video based human motion capture. In *Multimedia Signal Processing, 1999 IEEE 3rd Workshop on*, pages 483 –488, 1999.

[102] Lian Loke, Astrid T Larssen, and Toni Robertson. Labanotation for design of movement-based interaction. *Proceedings of the Second Australasian Conference on Interactive Entertainment*, pages 113–120, 2005.

[103] Yoshinori Maekawa, Takuya Oda, Taku Komura, and Yoshihisa Shinagawa. Learning system for human motion characters of traditional arts. *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 169–172, 2006.

[104] Anna Majkowska and Petros Faloutsos. Flipping with physics: Motion editing for acrobatics. *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Aug 2007.

[105] Mikiko Matsunaga and Victor B. Zordan. A dynamics-based comparison metric for motion graphs. *Computer Graphics International (CGI)*, 2007, 2007.

[106] Scott McCloud. *Understanding Comics: The Invisible Art.* Harper Paperbacks, 1993.

[107] Rachel McDonnell, Fiona Newell, and Carol O'Sullivan. Smooth movers: Perceptually guided human motion simulation. *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Aug 2007.

[108] Tom Molet, Zhiyong Huang, Ronan Boulic, and Daniel Thalmann. An animation interface designed for motion capture. *Computer Animation '97*, pages 77–85, 2002.

[109] Kooksang Moon and Vladimir Paviović. Graphical models for human motion modeling. In *Human Motion Understanding, Modelling, Capture, and Animation*, page 7, 2008.

[110] Julie B. Morrison and Barbara Tversky. The (in)effectiveness of animation in instruction. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pages 377–378, New York, NY, USA, 2001. ACM.

[111] Meinard Müller and Tido Röder. Motion templates for automatic classification and retrieval of motion capture data. *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Sep 2006.

[112] Kumiyo Nakakoji, Akio Takashima, and Yasuhiro Yamamoto. Cognitive effects of animated visualization in exploratory visual data analysis. *Information Visualisation, 2001. Proceedings. Fifth International Conference on*, pages 77–84, 2002.

[113] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443 – 453, 1970.

[114] Eric Neufeld, Mikelis Bickis, and Kevin Grant. Extending the dimensionality of flatland with attribute view probabilistic models. 6809(1):680905, 2008.

[115] Eric M. Neufeld, Sonje K. Kristtorn, Qingjuan Guan, Manon Sanscartier, and Colin Ware. Exploring causal influences. 5669(1):52–62, 2005.

[116] Marc Nienhaus and Jürgen Döllner. Dynamic glyphs - depicting dynamics in images of 3d scenes. *SG'03: Proceedings of the 3rd international conference on Smart graphics*, Jul 2003.

[117] Xiaolin Niu. Purpose based learner modelling. In *Proceedings of the Grad Symposium*. Department of Computer Science, University of Saskatchewan.

[118] Paul Noble and Wen Tang. Automatic expressive deformations for stylizing motion. *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 57–63, 2006.

[119] L.aura R. Novick and Douglas L. Morse. Folding a fish, making a mushroom - the role of diagrams in executing assembly procesdures. *Memory & Cognition*, 28(7):1242–1256, 2000.

[120] James F. O'Brien, Robert E. Bodenheimer Jr., Gabriel J. Brostow, and Jessica K. Hodgins. Automatic joint parameter estimation from magnetic motion capture data. *To appear in the proceedings of Graphics Interface*, page 1, 2000.

[121] W. Bradford Paley. Illustrative interfaces: Building special-purpose interfaces with art techniques and brain science findings. *SG'03: Proceedings of the 3rd international conference on Smart graphics*, Jul 2003.

[122] Min Park, Min Choi, Yoshihisa Shinagawa, and Sung Shin. Video-guided motion synthesis using example motions. *Transactions on Graphics*, 25(4), Oct 2006.

[123] John G. Pedley. *Greek Art and Archaeology*. Prentice Hall, 2nd edition, 1998.

[124] Bernhard Preim, Rainer Michel, Knut Hartmann, and Thomas Strothotte. Figure captions in visual interfaces. *AVI '98: Proceedings of the working conference on Advanced visual interfaces*, May 1998.

[125] Sara Price. Processing animation: Integrating information from animated diagrams. *Diagrammatic Representation and Inference*, pages 127–138, 2004.

[126] Joseph Psotka, L Daniel Massey, and Sharon A Mutter. *Intelligent Tutoring Systems : Lessons Learned*. L. Erlbaum Associates, 1988.

[127] Deva Ramanan and David A. Forsyth. Automatic annotation of everyday movements. *Computer*, 2003.

[128] James O. Ramsay and Bernard W. Silverman. *Functional Data Analysis*. Springer-Verlag New York, 2005.

[129] Kent C. Redmond and Thomas M. Smith. *Project Whirlwind: The History of a Pioneer Computer*. Digital Press, 1980.

[130] Charlotte M. Reed and Nathaniel I. Durlach. Note on information transfer rates in human communication. *Presence*, 7(5):509–518, 1998.

[131] Paul S. A. Reitsma and Nancy S. Pollard. Evaluating motion graphs for character animation. *ACM Transactions on Graphics (TOG)*, 26(4):18, 2007.

[132] Liu Ren, Alton Patrick, Alexei Efros, Jessica Hodgins, and James Rehg. A data-driven approach to quantifying natural human motion. *SIGGRAPH '05: SIGGRAPH 2005 Papers*, Jul 2005.

[133] Liu Ren, Gregory Shakhnarovich, Jessica Hodgins, Hanspeter Pfister, and Paul Viola. Learning silhouette features for control of human motion. *Transactions on Graphics (TOG*, 24(4), Oct 2005.

[134] Juan Diego Zapata Rivera. *Learning Environments Based on Inspectable Student Models*. PhD thesis, University of Saskatchewan, 2003.

[135] George Robertson, Jock Mackinlay, and Stuart Card. Cone trees: Animated 3d visualizations of hierarchical information. *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, Mar 1991.

[136] Alla Safonova and Jessica Hodgins. Analyzing the physical correctness of interpolated human motion. *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Jul 2005.

[137] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 1978.

[138] Michael Schmidt and Hod Lipson. Comparison of tree and graph encodings as function of problem complexity. *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1674–1679, 2007.

[139] Falk Schreiber. Comparison of metabolic pathways using constraint graph drawing. *APBC '03: Proceedings of the First Asia-Pacific bioinformatics conference on Bioinformatics 2003*, 19, Jan 2003.

[140] Dave Shreiner, Mason Woo, Jackie Neider, and Tom Davis. *OpenGL Programming Guide*. Addison Wesley, 2004.

[141] Marco Silva, Yeuhi Abe, and Jovan Popović. Interactive simulation of stylized human locomotion. *SIGGRAPH '08: SIGGRAPH 2008 papers*, Aug 2008.

[142] Baldev Singh, John Beatty, and Rhonda Ryman. A graphics editor for benesh movement notation. *SIGGRAPH '83: Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, Jul 1983.

[143] Clifford K. F. So and George Baciu. Entropy-based motion extraction for motion capture animation. *Computer Animation and Virtual Worlds*, 16(3-4):225–235, 2005.

[144] Henry Sonnet, Sheelagh Carpendale, and Thomas Strothotte. Integrating expanding annotations with a 3d explosion probe. *Proceedings of the working conference on Advanced visual interfaces*, pages 63–70, 2004.

[145] Robert Spence. *Information Visualization.* Addison-Wesley, 2001.

[146] Ivan Sutherland. Sketchpad: A man-machine graphical communication system. *AFIPS '63 (Spring): Proceedings of the May 21-23, 1963, spring joint computer conference*, May 1963.

[147] Christian Theobalt, Christian Rössl, Edilson Aguiar, and Hans-Peter Seidel. Animation collage. *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Aug 2007.

[148] Matthew Thorne, David Burke, and Michiel van de Panne. Motion doodles: An interface for sketching character motion. *ACM Transactions on Graphics*, 23(3):424–431, 2004.

[149] Adrien Treuille, Yongjoon Lee, and Zoran Popović. Near-optimal character animation with continuous control. *SIGGRAPH '07: SIGGRAPH 2007 papers*, Aug 2007.

[150] Edward R. Tufte. The cognitive style of powerpoint.

[151] Edward R. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative.* Graphics Press, 1997.

[152] Edward R. Tufte. *Visual Display of Quantitative Information.* Graphics Press, 2001.

[153] Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. Animation: Can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247 – 262, 2002.

[154] Barbara Tversky, Jeff Zacks, Paul Lee, and Julie Heiser. Lines, blobs, crosses and arrows: Diagrammatic communication with schematic figures. In Michael Anderson, Peter Cheng, and Volker Haarslev, editors, *Theory and Application of Diagrams*, volume 1889 of *Lecture Notes in Computer Science*, pages 389–408. Springer Berlin-Heidelberg, 2000.

[155] Munetoshi Unuma, Ken Anjyo, and Ryozo Takeuchi. Fourier principles for emotion-based human figure animation. *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, Sep 1995.

[156] Daniel Vlasic, Rolf Adelsberger, Giovanni Vannucci, John Barnwell, Markus Gross, Wojciech Matusik, and Jovan Popović. Practical motion capture in everyday surroundings. *SIGGRAPH '07: SIGGRAPH 2007 papers*, Aug 2007.

[157] Jue Wang, Steven M. Drucker, Maneesh Agrawala, and Michael F. Cohen. The cartoon animation filter. *ACM Transactions on Graphics*, 25(3):1169–1173, 2006.

[158] Ruixuan Wang, Wee Kheng Leow, and Hon Wai Leong. 3d-2d spatiotemporal registration for sports motion analysis. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[159] Colin Ware. *Information Visualization : Perception For Design*. Morgan Kaufman, first edition, 2000.

[160] Andrew Witkin and Zoran Popovic. Motion warping. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 105–108, 1995.

[161] Hai-Yin Xu, Dan Li, and Jian Wang. Implicit curve oriented inbetweening for motion animation. *GRAPHITE '06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, Nov 2006.

[162] Weiwei Xu, Yinpeng Chen, Hari Sundaram, and Thanassis Rikakis. Multimodal archiving, real-time annotation and information visualization in a biofeedback system for stroke patient rehabilitation. *CARPE '06: Proceedings of the 3rd ACM workshop on Continuous archival and retrival of personal experences*, Oct 2006.

[163] KangKang Yin, Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. Continuation methods for adapting simulated skills. *SIGGRAPH '08: SIGGRAPH 2008 papers*, Aug 2008.

[164] Jeff M. Zacks and Pascale Michelon. Transformations of visuospatial images. *Behavioral and Cognitive Neuroscience Reviews*, 4(2):96, 2005.

[165] Jeff M. Zacks, Barbara Tversky, and Gowri Iyer. Perceiving, remembering, and communicating structure in events. *J Exp Psychol Gen*, 130(1):29–58, Mar 2001.

[166] Zhiqiang Zhang, Zheng Wu, Jiang Chen, and Jian-Kang Wu. Ubiquitous human body motion capture using micro-sensors. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1 –5, march 2009.

[167] Victor Zordan and Jessica Hodgins. Motion capture-driven simulations that hit and react. *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Jul 2002.