Spring 2023

Entering the Dance Input/Output Machine

Antonio James Martinez
Bard College, tm7603@bard.edu

Entering the Dance Input/Output Machine

A Senior Project submitted to
The Division of Science, Mathematics, and Computing
of
Bard College

by
Antonio Martinez

Annandale-on-Hudson, New York
May, 2023

Abstract

My goal for this project is to provide an exploration into the intersection of Computer Science and Dance. Although there are many avenues of exploration at this intersection, I decide to focus on the topic of movement, specifically the idea of movement data. With this idea, we break it apart into two subtopics: movement input and output. Throughout this project, we mainly focus on the side of input, uncovering the nuance of movement language and the information that comes with it to understand the process of motion capture. We also demonstrate the use of movement input and how others may make use of the recorded information. This is all to then give us more context in understanding movement as output.

Contents

Dedication

To my brothers, Austin and Aaron, and their journey into high school and college.

Acknowledgments

I would first like to thank both Valerie Barr and Dominique Townsend. Both of these women inspire me every day and have helped me believe in my ability to succeed at times when it didn't seem possible. You remind me what it really means to not only be a professor but also a teacher. Valerie has shown me once again that failure isn't something that I should shame myself for and I can still be confident in my ability to move forward. Dominique has shown me how much I could push myself and helped me understand what strength I actually have to persevere. I appreciate these women so much and I hope they continue to impact the lives of more students to come!

Next, I want to thank my family. This is for being my unconditional support system, no matter the situation. My parents specifically have sacrificed so much just so that my brothers and I would be able to have a decent education. I wanna give credit to the financial, physical, and emotional labor they have put in for us so that I would be able to see this moment. Thank you Mami and Papi, te amo mucho.

This next thank you is for my absolute favorite person I have met at Bard College whom I still continue to give my love to the fullest extent. Kyla-Jerusha Bailey, totally random, but I always struggle to put into words how I feel about our amazing relationship because you never fail to put a smile on my face or be the highlight of my day. You genuinely are one of my biggest inspirations and I seriously couldn't have asked for anything more in a friendship. I love you, I put that on everything, and I can't wait to be reunited!

My next thank yous go out to my Computer Science study gang: Emely Galeanop, Alex Nguyen, and Erica Lee! You guys are genuinely amazing people and I'm so glad to have officially met you during this past year. Y'all have been the anchor I needed to keep going and not give up because you completely understand how real the struggle actually is. I've always felt comfortable admitting that I didn't know something to any of you, because you guys have made it a safe space for all of us to learn and power through together. Love and appreciate you guys so much.

Finally, I want to send a huge thank you to the whole reason why I've grown to fully embrace and appreciate Dance so much to the point I wrote my project on it. This is for Afropulse, Chronik Vibez, and KDC — three of the best dance teams on campus that I've had the blessing of being a part of. I love each and every one of ya and I hope that is very clear. These clubs are where I've come back to whenever I'm not having the best day and they never fail to make an impact on me. I, of course, appreciate coming to rehearsals and improving my skills, but even more than that, you guys are my everything and I'm just so happy that we've been able to create a space for this amazingly talented community. You guys are everything I've wanted in a chosen family and I'm only wishing the best for you all!

1
Introduction

Between Computer Science and Dance, we're able to explore a relationship, specifically in the realm of movement, that may not be initially obvious. In this project, we will focus on the general questions concerning movement, recording data from input motion, and also demonstrate the use of this input data. Many understand Dance as something that people do to entertain and have fun through movement, while Computer Science is known to be about sitting down at a computer or laptop in order to "hack into the mainframe". However, while some of that is partially true, both of them individually have incredibly diverse complexities and levels of nuance within them. While both topics are fundamentally different in a multitude of ways, there are subsets of each that allow for an intersection of ideas and even creativity. As a means of understanding this intersection, I wish to provide examples of where these two topics overlap. I also hope to give clarity for how movement data is used through the mechanics of motion capture.

My interest in movement data arose from the question "Would it be possible for a computer to generate its own choreography?" As I explored this question, I was able to break it down into understanding dance as input and output, and therefore explore the use of movement data. What this means is that just as computers are able to take advantage of numbers, words, and files, this would also include the use of information that a computer is able to process that is generated from movement or computational movement. From this dynamic between Dance and Computer Science, a lot of other questions arise such as "how is movement processed" and "what information or features are most important when considering movement for computers?" Through the discussion and focus on movement, I seek to explore the use of computational movement input and apply that to my understanding of generating movement output.

2
Context

2.1 Concerns About Movement Language

What does it mean to "code movement"? This is another question I investigate in this project, and to effectively explain the concept of movement in Computer Science, this project will follow a specific procedure. Before understanding movement as input and translating dance into code, there needs to be a clear insight on how one determines the parts of movement that are necessary to observe. Part of this breakdown involves understanding what language to use when it comes to motion.

Although there is Physics terminology relating to motion such as position, velocity, and acceleration, this project provides insight on the terminology and semantics associated with Dance and the human body to attach vocabulary to aspects of movement that are relevant in motion capture. When thinking about the process of movement manipulation, with the associated language of the moving body, we can then record or produce data related to that language.

2.2 Concerns About Movement as Input

When considering movement as input, the first concern is what is considered important enough to be valuable to a computer. The quantification of movement helps make clear what could potentially be observed, but the next step would be determining what would be the most useful to record. The majority of this project will involve the process of motion capture, but before making use of it, it's helpful to know what this process actually entails. Understanding the mechanics of motion capture is the crucial part of this project that will allow further exploration of movement input once it's understood. Questions such as what motion capture is and how it works will be explored during this project and hopefully give insight into what possibilities are available to those who use it.

After taking advantage of motion capture systems, the associated software would be the next topic of concern. The programs that are able to interact with the information provided by the capture systems are what make movement study possible and will help us understand the information retrieved when movement data is collected. In this project, we will also determine the functions and benefits of software systems for motion capture technology. This will allow us to directly interact with the movement data and gain a better understanding of what movement input looks like. In my own experience interacting with the data, we will see the possibilities of manipulating this data and finally connecting movement to computational information.

3 Intersections Between Computer Science and Dance

3.1 General Overlap in Research

It's important to recognize that, when discussing the combination of Dance and Computer Science, not all experiments involve heavy use of computing and manipulating data. One example of a project with this characteristic is found in the Civic Media project involving dancers who were given data and were told to provide some type of interpretation of said data. The inherent purpose of the experiment was to observe how one may understand what it means to collect and present data in order to use it as a means to provide evidence. The Civic Media's questions involved the idea of how people can get encouraged to understand and tell stories using data by actively engaging with it. One of those approaches involved the literal dance interpretation of presented data to understand immediate associations with the given information.[1]

In comparison to these more Dance related experiments and projects, other projects focus on data directly associated with the Dance community. An example of this is the Dance Data Project (DDP), involving the collection and analysis of data information regarding gender equity in dance and dance companies. The mission of the Dance Data Project is "To promote gender equity in the dance industry, including but not limited to ballet companies, by providing a metrics based analysis" which allows others to draw computational information and hopefully create a better environment within the industry.[2] Information like this does the opposite of the Civic Media project and translates what is going on in the Dance world and provides statistical information that can be computed and analyzed. Although these projects consist of heavily important work, we can now discuss other projects in Computer Science that focus on the movement aspect of Dance.

3.2 Overlap in Movement-Related Research

For this research project, I focus on the topic of movement in Dance and the place movement data has in the Computer Science world. Dance as an art form involves a collection of things such as brain activity, audience interaction, company management, improvement perception, and relationship with the body, and many projects have discussed these aspects, but with movement, there seems to be a disconnect in understanding how movement could be translated into code. Marcus G. Pandy, in his 2001 "Computer Modeling and Simulation of Human Movement", explains the basics of computer models of movement and what they should include. There are also other questions such as what makes these computer models useful in understanding movement, which is heavily related to the questions I'm asking in my own project to understand how movement data is collected. Pandy argues for the use of these models to both better understand the mechanics of movement but also to produce such movements on a different screen. One of the arguments he makes for computerized modeling is the potential for its use in the orthopedic and general medical fields to better understand disruptions in movement for patients. Not only does he mention the benefits, but there is also some explanation for the challenges facing the field as well in order to promote the importance of this implementation of technology. One struggle Pandy highlights is the need for faster computer speeds at the time to enable deeper complexity of the model and therefore more accuracy. Another includes the ability to even more accurately provide a representation of the musculoskeletal system (266-267, Pandy).[3]

Putting this information into practice, some experiments take advantage of this modeling technology and even allow it to serve a purpose as a real-life application. Paul Soumyadeep et al, for example, describe in their "Virtual Kathakali: Gesture-Driven Metamorphosis" how they were able to create a virtual avatar to follow the movements and instructions of a coach or instructor and teach the students in real-time. This paper explains in detail the step-by-step process of what it took to render this virtual dancer mimicking the motion of someone who was not present. They break down the three modules of the system as well as the successes and pitfalls of it in order to eventually reach their goal. The three modules include "Real-time Detection of arm movements of the user, modeling of the Kathakali dancer, and reproduction of the pose in the Kathakali dancer's model. While the main success consisted of getting the model to work effectively, other successes included changing cameras and using a dark background for the user. With the parameters and environment given, however, the user may potentially feel constrained.[4]

4
Movement Literacy and Quantification

4.1 Movement Language and Its Importance

In Dance, there is a language that is definitely not completely understood by the general public. This includes terms like "marking", "kick-ball-change", "count", "pointe", "isolation", and even the sound effects a choreographer makes whenever hitting a move instead of using counts. While these terms all have specific meanings assigned outside of the Dance world, they also have completely different meanings within it. This exhibits the importance of movement language within the Dance world, which also translates to its importance in motion capture. Part of understanding movement input data is recognizing the attributes of movement that would be useful as computational information that allow the computer to manipulate it.

One of the first steps to understanding movement language is to grasp the idea of movement notation. This stems from the desire to record movement in the same way we record music. Because of the complexity of movement in three-dimensional space, however, this becomes difficult. Some examples of dance notation that provide a strategy for naming different aspects of movement include Labanotation (the most popular), Benesh Movement Notation, Eshkol-Wachman Movement Notation, and DanceWriting.[5] Although these aren't the most accurate forms of notation yet, they do provide a system of written language that is able to be understood universally.

There is also the idea of movement analyses and theories, such as Laban Movement Analysis, that also provide a spoken vocabulary to discuss aspects of movement in the same way.[6]. Overall, the most common attributes in dance language consist of the paths a person and their limbs make, the speed at which they make these paths, the direction that each of their parts face, and the position or distance the parts are with respect to the person. With these symbols that place labels on specific aspects of movement, we can now get a clearer picture of what is important to observe when it comes to movement, at least in the eyes of a dancer.

According to Manish and Chakrabarty, dance grammar is another aspect of dance and movement literacy.[5] This consists of a plethora of rules that set the possibilities of certain movements, similar to that of math theorems or music theory. These rules take into account the capabilities of the human body, along with the possible efforts one can muster, and creates limitations on what it can and can't do. Joshua Stuart and Elizabeth Bradley explain this concept in the context of their Computer Science background. The majority of dance grammar, they describe, is mathematical figures and representations, which will be discussed later, but these are used to represent things such as posture, joint orientation and movement, and position.[7] Of course, similar to the general ideas of dance notation, but with more concise boundaries instead of depending on the use of symbols and illustrations. But to understand these boundaries and also get closer to our goal of turning movement into data, we must quantify the aspects and symbols that continuously coming up.

4.2 Quantification of Movement

Establishing grammar for dance is pretty similar to making rules for a game, and it's just a matter of understanding what your character or token is capable of doing. Here is where we start getting a better depiction of what computational movement means as we start using math to find formulas and symbols to provide more insight into these aspects of movement. In order to provide accurate limitations for a specific dancer, we must use the basic foundations of mechanics and physics by using vectors and scalars, items that are definitely used in computer science, that allow for the use of quaternions. In the words of Stuart and Bradley, a "quaternion $q = (r, \vec{u})$ consists of an axis of rotation $\vec{u}$ and a scalar r that specifies the angle of rotation about $\vec{u}$."[7] So with this information, along with the number of joints and the information considering the positions and orientations of these joints, there seems to still be complexity, but at the very least more clarity as to how we can represent these movements.

Movement, in general, is not only about what movements one can do but also the realistic possibility of what movements precede or follow each action. When we consider the order in most situations, we would then like to get into the idea of set theory, another concept that is definitely exploited when coding. Stuart and Bradley continue by discussing the set of *allowed* orientations versus the set of *actual* orientations of joints by utilizing the closest quaternion in the set Q^λ (holding a finite length of M^λ) for each joint λ. They also explain that although joint orientations and movement are continuous, the discreet versions of their values better engage with the complexities of their algorithm. This involves setting the components of the discretized vector for the body position s to each quaternion in Q^λ that is closest to the continuous vector component of b by using K-D trees as their data structure of choice.[7] What all of this means is that each joint will have its own set of possible movements and orientations that will be described using a set of quaternions as mentioned before to create specific frames of dance that have the potential to be observed computationally, further diving into this idea of computational movement.

One last important feature of dance grammar is the concept of transition graphs or, in this context, a joint orientation graph as Stuart and Bradley call it. They describe it as "a weighted-directed graph that captures the transition probabilities in a symbol sequence." Within this graph, the possibility of a joint moving to a certain orientation is determined by the weighted edge between vertices and makes use of its discretized values. Joint Orientation graphs also make use of the formally mentioned sets of quaternions by having every element represent a vertex in the graph. This way, we are able to manipulate this graph in any way needed computationally. With these rules and computations, the movement of the human body has specific limitations that may be captured by not only the audience and observer but also the motion capture technology.[7]

4.3 Relevant Movement Information for Motion Capture

With an understanding of what it takes for one to record a dance or sequence of movements, we can now observe what is most desirable and noticeable in movement. With dance notation, there is a goal to accurately capture the general movement of an individual, while also keeping the three-dimensional space in mind. Things like space, path, speed, and orientation are the main concerns when observing movement. In contrast, while notation takes note of what the human body *can* do, dance grammar emphasizes what the human body *cannot*. The rules put heavy restrictions on the orientation and sequence of events involving the joints and posture of the body. What all of that does for us is give us a clear picture as to what might be highlighted and used for computation when recording movement for input.

5
Processing Movement as Input

5.1 Desirable Units of Movement for Input

A lot of this discussion consists of the idea of movement as unit of measurement. What would the computer specifically desire out of a moving person in order to manipulate it later? This question mostly relies on what type of motion capture we are dealing with because the process in which they accomplish the main task of tracking motion can be incredibly different. Some methods rely on the use of body heat while others use specific markers to identify specific movements. One method, and probably the one we are most interested in, lacks any markers and strictly relies on software to identify movement. In order to understand motion capture's mechanics and data, we first look at how the movement is recorded to then translate it into computational information.

5.2 Motion Capture Technology: Hardware

One device that leaves room to observe movement, but not necessarily for capturing it are sensors. According to Manish and Chakrabarty, a sensor "receives and responds to a signal on a certain preset event (say 'when touched')" and allows for any specific action on the body to be detected and recorded.[5] Sensor markers are placed on the body that would be most ideal to be identified for movement. Most sensors are different in the sense that they are all programmed to detect different actions. One may require a touch while another may need something to move in front of it to be triggered. Once the sensors are tipped off, they alter their electrical state and are capable of sending signals to desired locations through the connected software.

Monocular optical motion capture allows for the combination of one camera and the use of markers. The camera used is infrared, meaning that it's sensitive to certain wavelengths of light and heat that allow for the recognition of human bodies moving.[8] As opposed to solely video motion capture, where multiple cameras are used to observe every angle of a figure in three-dimensional space. This is usually what is used specifically in motion data studies to observe details in motion capture.

One particular piece of hardware equipment for motion capture is the depth camera. These cameras send out pulses of light that should reflect back at the lens to identify the distance something in the frame is from the camera. What this does is render a three-dimensional map of the frame to capture the distance each pixel of the image is at.[9] With all of this equipment, however, the information this hardware observes isn't what a computer processes, so now we must look at what is actually inputted as movement information.

5.3 Motion Captured Information

For the sensors and markers placed on a human body, the mechanics of collecting data work in the same capacity as tracking an object's position in space. As the tracker moves further from the camera or from a starting position, the distance information is tracked as well as the specific path these trackers make in order to replicate that virtually, leaving room for direct manipulation of certain values and images for graphing onto another medium. This data is also made sure to be as continuous as the movement in actual reality, and so the discretized values of the orientation of the joints and body as mentioned before become continuous once again so they may become similar to one's movement.[10]

In contrast, markerless motion capture, whether using one or multiple cameras, requires a more complex approach in order to achieve the same goal and this is by using machine learning.

How this works is that the cameras will record one's normal movement in whatever angles are necessary to achieve the researcher's goal, and search through a pre-trained library of data that holds a collection of similar images and sequence of images that allow for a two-dimensional "skeleton" of the person moving. This is called pose estimation and with this skeleton, the same assumptions the sensor and tracker-based motion capture technology make concerning distance and path are made by this skeleton's movement and distance from the camera. Some approaches have also allowed for three-dimensional pose estimation and even without a depth camera but the same process still applies.[11]

5.4 Motion Capture Technology: Software

After understanding what goes into a computer when capturing movement as a source of data, the last question to answer about movement input is how software systems read and interpret this data in order to accomplish other things. We can determine how this software's motion capture technology functions and then apply it to our current understanding of motion capture hardware and connect how the translation happens.

As of right now, we understand motion-captured data to be values that represent distance, path, and orientation as the units of movement data. This is a lot of information that seems a bit disjointed to collect into a dataset used for deep learning and neural network techniques. So in order to combine and consolidate this information, the motion-captured data is placed in something called a BVH file. Written in ASCII code, this specific file references and encodes the starting position of a figure and the sequence of poses that follow.[12] These files also provide a number of frames, frame time, and coordinates that allow for even deeper computation. Within these files, there also lies specific vocabulary such as offset and channel that are described to be the parts of a joint, where offsets are the relative distance from their parent joints and channels are the number of routes towards other joints.

Some examples of software that utilize this file format include Blender, Poser, and Maya. Using our human computational abilities, the way we would interpret the data placed in a BVH file would be to use a transformation matrix for the local translation and rotation data for the segment of the body the file refers to so that we would be able to find the position of a certain body segment of the skeleton, according to the research done in "Biovision BVH".[13] For these software programs to read these files, they are organized into what the name of the file refers to: a hierarchy of joints. This provides information on every joint including the root as well as its offsets and channels.[14]

5.5 Relevant Movement Input Information for Motion Capture Use

So now understanding the basic mechanics and processes that occur during motion capture, we can now bring that into our search for answers when it comes to movement input and decide how to approach the questions asked. The hardware in particular, as we now understand, uses different forms of language and uses different aspects of recording to ultimately achieve the data desired. This data ultimately consists of a record of distances and paths from the camera that allow the software to easily manipulate it. We then learned that there is even more complexity in the information picked up from motion capture technology when discussing the idea of motion capture data files, the BVH file in particular where even more precise descriptors of the body's movements are written in an organized fashion for computers to make use of. Now we can apply this to our own attempt at using motion capture information as input.

6 Experience with Motion Capture Technology

6.1 Goals

My goal for this project is twofold. Conceptually, as a reminder, we are attempting to explore the machinations of motion capture by understanding what it means for movement to be input and how it's manipulated computationally — hence the idea of computational movement. This is part of an even larger goal which is to then understand how one may use their own movement input in a way that would enable the computer to output its own choreography if I so choose to continue this research further. Using the information we've discussed concerning movement literacy and motion capture, I would like to manipulate movement input data to fully understand what the software takes into account while coding movement information.

My practical goals for this project consist of putting to use my knowledge of motion capture and movement input and applying this understanding to code. My ultimate goal is to run and understand code that manipulates movement input data, so in this project, I will be attempting to generate two-dimensional skeletal models for multiple figures in view as a means to explore what it takes for movement input to be manipulated. For this to happen, I need to find certain hardware and software that are not only functional on their own but also compatible with each other. I will also be using code that has accomplished a similar task of generating three-dimensional skeletal models and applying that to my own goals. Finally, to demonstrate and test this code, I would like to explore the complexities of this data and attempt to create a dance that would trigger an animation based on the movement input that is being provided.

6.2 Influence on Choreography

Although this project mainly focuses on the computation of movement, I still think it's important to discuss the experience I had while my cast and I choreographed this dance in order to demonstrate my findings. A lot of the concept for the piece had already been premeditated before working on the choreography, but I had always struggled with finding the right movement sequence as well as the genre of the phrases I would want to use that would accurately describe the emotion I would want to put behind this piece I had been thinking about. But after getting deeper into this research, I was inspired to challenge myself by basing my movements on what I've learned about motion capture and this heavily influenced the ideas for the movement I wanted to use for this piece while still maintaining the integrity of the original idea and storyline.

My main goal with the movement was to ultimately test out the range of capabilities the motion capture technology would have and see if the end result would remain —- keeping the skeletons on those that are moving in the frame. One example of a movement that was inspired by this goal was the use of different shapes that both dancers would make during the piece. One question I had early in the process was "How does the motion capture technology work if someone is partially blocking someone?" "Does it assume that a figure is still there?" Other questions arose in the same fashion where I wanted to test out the difference in distance from the camera between figures as well as the pose estimation's interpretation of floorwork. Finally, after everything I've learned about how much motion capture values and makes note of rotation and orientation, I wanted to exploit that in a lot of parts of the dance in different contexts and with different joints. Ultimately, this produced a great product of variety in movement as well as a preserved storyline of what I had originally had in mind.

6.3 Experience with Hardware

In my journey attempting my own version of motion capture to demonstrate my understanding of movement input, my first step was to find adequate hardware to observe movement. If necessary, I would use my simple camera on my laptop. However, I did want to use technology that would provide more information about the movement in the frame and so I was determined to find some type of genuine motion capture equipment. Realistically, the mocap suits and sensors are simply too expensive and mostly used in larger-scale projects, but even so, the main focus in motion capture that we have explored is markerless figure recognition, so optical motion capture equipment seemed to be more of the goal. With this in mind, one of the best options that would aid me in my research on movement input was the depth camera. So the decision was made, and I ended up with an Intel RealSense Depth Camera.

6.4 Experience with Software

Next was to find some software that would allow me to not only accomplish my goal but also be compatible with the depth camera I decided to use. The depth camera also connects to the USB jack on my laptop and has the same abilities as the laptop's camera, so this definitely made my life easier. So while on my journey, I found this article that accurately described what I was attempting to accomplish. While reading this article, I noticed that they also had a GitHub repository for their open-sourced code as well as a video explaining the project.[15] This is what I wanted to replicate and explain the process of their algorithm that allowed for this to be possible.

Looking at the repository, I noticed that the code was in Python. The code also used PyTorch as the framework and CUDA as the main model for deep learning operations. With frameworks, the methods are already provided for you to base your work upon them. Specifically with PyTorch, this framework is best for deep learning. CUDA almost acts as an add-on that allows for more operations to run for deep learning models. Because my laptop is unable to support this code, I used Google Colab to run this algorithm. The main Python file I was able to run was called evaluate_timing.py which evaluates "the forward pass time and the accuracy of the trained model on ImageNet validation set".[16] Although this isn't exactly my goal of replicating the pose estimation that I desired, it was still interesting to see the efficiency of the researchers' model in order to accomplish the goals discussed.

This code exhibits the training of the SelecSLS model the researchers created through their neural network ImageNet. With their algorithm, I was still able to understand how movement input was able to be put to use and manipulated after being taken as data.

Most of what this algorithm depends on are the "Net" class from "selecsls.py" that "evaluate_timing.py" references. After initialization, this code defines the stem as the sequential container of modules that applies a two-dimensional convolution, batch normalization over a four-dimensional input, and the rectified linear unit function. Because the "config" attribute of the Net class is "SelecSLS84", the if statement defines the configuration of the network after the initial stem. The code then defines the head as the sequential container of three modules of sequential containers that apply different values than what was done to the stem including one different sequential container that does a similar job but with a normalized convolution. The if statement then sets the number of features to 1280. Building on SelecSLS's core, the Net class also has a two-dimensional "features" array that gets filled with a for a loop by appending a SelecSLSBlock object that holds its own six sequential containers. This then allows nn.Sequential to be applied to the features array. Back to "evaluate_timing.py", the function "benchmark" that references the Net class starts evaluating two different batch sizes, 1 and 16, on GPU and then on CPU with batch size 1. These evaluations use the functions "measure_gpu" and "measure_cpu" which both put the model in evaluate mode as opposed to training mode, to then time the Net objects' efficiency either using CUDA (GPU) or not (CPU). During the evaluation, the code uses a for loop to measure each in range 10 and then 50 for the model the researchers created and then using the JIT (Just-In-Time) compiler. The times are measured using the Net object and a random tensor for the model and torch.jit.trace() as the JIT's Net with a random tensor as well. When the times are collected into an array, the mean is taken and then multiplied by 1000 to reveal the results in milliseconds.

6.5 Results

From this algorithm, we get "Model FP: 11.904783248901367ms" and "JIT FP: 11.484670639038086ms" while the GPU Batch Size is 1. While the Batch Size is 16, "Model FP: 136.40655040740967ms" and "JIT FP: 126.87216281890869ms". And finally, while the CPU Batch Size is 1, "Model FP: 11190.253763198853ms" and "JIT FP: 11093.708634376526ms".

What this tells us about the model is the time it took this model to train and execute when running. When using a library of movement data, this is what we are able to test when considering models that would desire to experiment with it.

7 Conclusion

In this project, I explore the intersection between Computer Science and Dance through the concept of movement as data, honing in on movement input. After reviewing some other works on the subject, we were able to see the similarities of how others utilized movement input and apply it to our own study of it. After exploring the different avenues of experimentation and research, we then discuss the ideas of Dance semantics and quantification using motion capture. According to our research on notation and motion, the most important aspects of movement include the distance from the observer, the path of body parts, the orientation of joints, the sequence of poses, and the speed of motion, which would serve as important recorded traits for movement input. We then learned to quantify these aspects of movement using quaternions, which gave us a direct value to represent movement and be able to manipulate this data.

With the discussion of the different types of motion capture and what is being observed when in use, we learn that movement input requires specific formatting to be used for computer code. This formatting comes from BVH files which allow us to understand the values of movement that we are able to manipulate when considering movement data. From my own experience, we were able to understand how movement input was able to be used and manipulated once it was recorded.

8
Future Work

So as we know, part of this project's purpose was to get a better understanding of movement input so that we could then potentially move toward movement output. This way we can understand the mechanics of how to generate movement data instead of recording it. Some goals for this project would include finding software that would promote the application of movement output data on a figure and understanding how a computer would generate random movement as opposed to directed movement from the user. Once these goals are accomplished, our understanding of a computer's ability to interpret movement data would be complete.

Bibliography

[1] Victoria Palacin et al., *What happens when we dance our data?*, https://civic.mit.edu/index.html%3Fp=2569.html.

[2] Dance Data Project, *About DDP*, https://www.dancedataproject.com/about-ddp/.

[3] Marcus G. Pandy, *Computer Modeling and Simulation of Human Movement*, Annual Reviews **3** (2001), 245–273.

[4] Soumyadeep Paul et al., *Virtual Kathakali: Gesture-Driven Metamorphosis* **1** (1998), 1–12.

[5] Joshi Manish and Sangeeta Chakrabarty, *An extensive review of computational dance automation techniques and applications*, Proceedings of the Royal Society A **477** (July 2021), 1–21.

[6] Robin Konie, *A BRIEF OVERVIEW OF LABAN MOVEMENT ANALYSIS*, http://psychomotorischetherapie.info/website/wp-content/uploads/2015/10/LMA-Workshop-Sheet-Laban.pdf.

[7] Joshua Stuart and Elizabeth Bradley, *Learning the Grammar of Dance*, https://www.researchgate.net/publication/221345975_Learning_the_Grammar_of_Dance.

[8] Rokoko, *What is Motion Capture, and How Does it Work in 2022?*, https://www.rokoko.com/insights/what-is-motion-capture-and-how-does-it-work-in-2022.

[9] Jamie Kelly, *How Does a Depth Sensor Camera Work? (The SIMPLE Answer)*, https://www.lapseoftheshutter.com/how-does-a-depth-sensor-camera-work/.

[10] Film Lifestyle, *What Is Motion Capture (Mo-Cap): The Definitive Guide*, https://filmlifestyle.com/what-is-motion-capture/.

[11] Nobuyasu Nakano et al., *Evaluation of 3D Markerless Motion Capture Accuracy Using OpenPose With Multiple Video Cameras*, Front. Sports Act. Living **2** (2020), 1–9.

[12] CHAN ka Chun et al., *BVH Motion Capture Data Animated.*

[13] Biovision, *Biovision BVH*, https://research.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html.

[14] M. Meredith and S. Maddock, *Motion Capture File Formats Explained*, 1–36.

[15] Dushyant Mehta et al., *XNect: Real-time Multi-Person 3D Motion Capture with a Single RGB Camera*, ACM Transactions on Graphics **2** (Apr 2020), 1–24.

[16] ______, *On Implicit Filter Level Sparsity in Convolutional Neural Networks*, https://github.com/mehtadushy/SelecSLS-Pytorch.git.

[17] Valeria Lo lacono, *Presentation of the Dance Research Data (4.6.4)*, https://worlddanceheritage.org/dance-research-data-model/.

[18] Jessica Rajko, *Critical Study of Computing Research Involving Dance*, http://www.jessicarajko.com/computingdance.

[19] Benesh in Action, *What is Benesh Movement Notation?*, https://beneshinaction.wordpress.com/what-is-benesh-movement-notation/.

[20] The Noa Eshkol Foundation, *About Eshkol Wachman Movement Notation*, https://noaeshkol.org/about-eshkol-wachman-movement-notation/.

[21] OptiSol, *Depth Perception Using Intel Realsense*, https://www.optisolbusiness.com/insight/depth-perception-using-intel-realsense.

[22] Akkshaj Singh, Ramon Qu, and Rosario Scalise, *Using the RealSense Camera*, https://mushr.io/tutorials/realsense/.

[23] Adobe, *What is motion capture and how does it work?*.

[24] BALLUFF, *Function, construction 'I&' technologies of sensors*, https://www.balluff.com/en-us/basics-of-automation/fundamentals-of-automation/function-construction-technologies-of.

[25] Peter Terren, *Electricity and sensors*, https://www.sciencelearn.org.nz/resources/1602-electricity-and-sensors.

[26] FILExt, *Opening a BVH File*, https://filext.com/file-extension/BVH.

[27] Tomas Šinkunas, *BVH Importer for After Effects*, https://www.rendertom.com/downloads/web/bvh_importer_help.pdf.

[28] Nuno N Correia et al., *Designing Interactive Visuals for Dance from Body Maps: Machine Learning and Composite Animation Approaches*, DIS '22: Designing Interactive Systems Conference (June 2022), 204–216.

[29] Sarah Fdili Alaoui et al., *Interactive Visuals as Metaphors for Dance Movement Qualities*, ACM Transactions on Interactive Intelligent Systems **5** (Sept 2015), 1–24.

[30] Andrew Myers, *AI-powered EDGE Dance Animator Applies Generative AI to Choreography*, https://hai.stanford.edu/news/ai-powered-edge-dance-animator-applies-generative-ai-choreography?utm_source=Stanford+HAI&utm_campaign=fed781b8c4-Mailchimp_HAI_Newsletter_April+2023_3_General&utm_medium=email&utm_term=0_aaf04f4a4b-fed781b8c4-214102738.

[31] Miriam Felton-Dansky, *PROCEDURAL DRAMA: Miriam Felton-Dansky on the theater of Annie Dorsen*, https://www.artforum.com/print/202302/miriam-felton-dansky-on-the-theater-of-annie-dorsen-89994.

[32] Lijia Liu, Joseph L. Cooper, and Dana H. Ballard, *Computational Modeling: Human Dynamic Model*, Frontiers in Neurorobotics. **15** (Sept 2021), 1–16.

[33] Peng-zhan Chen, Jie Li, Man Luo, and Nian-hua Zhu, *Real-Time Human Motion Capture Driven by a Wireless Sensor Network*, International Journal of Computer Games Technology **2015** (Feb 2015), 1–14.

[34] Virtual Production Malta, *The History of Motion Capture Technology*, `https://virtualproductionmalta.com/blog/the-history-of-motion-capture-technology`.

[35] Codecademy Team, *What Is a Framework*, `https://www.codecademy.com/resources/blog/what-is-a-framework/`.

[36] Andrew Murphy, *Batch size (machine learning)*.

[37] PyTorch, *Fusing Convolution and Batch Norm Using Custom Function*, `https://pytorch.org/tutorials/intermediate/custom_function_conv_bn_tutorial.html`.

[38] Databricks, *Convolutional Layer*.

[39] Aman Arora, *Is MLP-Mixer a CNN in Disguise?*, `https://wandb.ai/wandb_fc/pytorch-image-models/reports/Is-MLP-Mixer-a-CNN-in-Disguise---Vmlldzo4NDE1MTU`.

[40] Pirun Hemmondharop, *A comparative study of Just-In-Time (JIT) and Theory of Constraints (TOC) systems with varying constraint systems with varying constraint locations and operational characteristics*, Doctoral Dissertation **1429** (2002), 1–122.