MoViz: A Visualization Tool for Comparing Motion Capture Data Clustering Algorithms

Lucas Liu
Expressive Machinery Lab
Georgia Institute of Technology
Atlanta, GA
lucasliu@gatech.edu

Duri Long
Expressive Machinery Lab
Georgia Institute of Technology
Atlanta, GA
duri@gatech.edu

Brian Magerko
Expressive Machinery Lab
Georgia Institute of Technology
Atlanta, GA
magerko@gatech.edu

ABSTRACT

Motion capture data is useful for machine learning applications in a variety of domains (e.g. movement improvisation, physical therapy, character animation in games), but many of these domains require large, diverse datasets with data that is difficult to label. This has precipitated the use of unsupervised learning algorithms for analyzing motion capture datasets. However, there is a distinct lack of tools that aid in the qualitative evaluation of these unsupervised algorithms. In this paper, we present the design of *MoViz*, a novel visualization tool that enables comparative qualitative evaluation of otherwise "black-box" algorithms for pre-processing and clustering large and diverse motion capture datasets. We applied *MoViz* to the evaluation of three different gesture clustering pipelines used in the *LuminAI* improvisational dance system. This evaluation revealed features of the pipelines that may not otherwise have been apparent, suggesting directions for iterative design improvements. This use case demonstrates the potential for this tool to be used by researchers and designers in the field of movement and computing seeking to better understand and evaluate the algorithms they are using to make sense of otherwise intractably large and complex datasets.

CCS CONCEPTS

• **Applied computing** → *Performing arts*; • **Computing methodologies** → **Cluster analysis**; *Dimensionality reduction and manifold learning*;

KEYWORDS

unsupervised learning, visualization, explainable AI, movement improvisation, gesture clustering, motion capture data

ACM Reference Format:
Lucas Liu, Duri Long, and Brian Magerko. 2020. MoViz: A Visualization Tool for Comparing Motion Capture Data Clustering Algorithms. In *7th International Conference on Movement and Computing (MOCO '20), July 15–17, 2020, Jersey City/ Virtual, NJ, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3401956.3404228

1 INTRODUCTION

Motion capture (MoCap) data is useful for machine learning applications in a variety of domains (e.g. animating characters in video games [6], creating more engaging physical therapy experiences [8], enabling tangible/embodied interaction with technology [20], and supporting and augmenting artists' creative movement improvisation in fields such as dance or theater [9, 15]. Many of these domains (e.g. movement improvisation) require large open-ended datasets with data that is difficult to label or classify. Consequently, researchers working with diverse datasets often choose to analyze gestures using unsupervised learning, which does not require labeled data [3, 16]. Unsupervised learning algorithms can aid in clustering unlabeled gestures together based on similarity, but the meaning and logic behind the behavior of these algorithms is usually "black-box" (i.e. their inner operations are difficult to inspect). Therefore, it can be challenging to compare the applicability of different algorithms for a given context. Although statistical analyses can be run, the evaluation of algorithms applied to MoCap data tends towards subjectivity and oftentimes necessitates a human eye (particularly in open-ended domains such as dance).

In this paper, we ask: how can we design a tool that expresses the unique semantic properties of motion capture data, enabling comparison and evaluation of the ways in which unsupervised learning algorithms cluster data? We explore this in the context of *LuminAI*, an AI system that can improvise movement with human dancers. We hope to use our tool to answer questions from the perspective of human visual understanding such as "what makes us consider two gestures similar?" and "what does a low geometric distance between two points in this space mean?" Due to the inherently exploratory and open-ended nature of these questions, we believe that a rigorous quantitative analysis for an end user would not provide the human-intuitive insight regarding the semantic properties of the motion capture data. The task of comparative evaluation of human motion is instead more well-suited for an interactive data visualization tool. A variety of other works have explored ways of visualizing MoCap data [5, 10, 18], but none enable a qualitative and embodied comparative evaluation of algorithmic pipelines for clustering large and diverse datasets.

2 *LUMINAI*

LuminAI is an interactive art installation that facilitates a space for embodied co-creativity and improvisation between a human user and a virtual agent. The system consists of a virtual agent projected onto a screen next to a virtual shadow of the user. A Microsoft Kinect 2 sensor is used to collect the user's continuous motion and

MOCO '20, July 15–17, 2020, Jersey City/ Virtual, NJ, USA

ACM ISBN 978-1-4503-7505-4/20/07…$15.00
https://doi.org/10.1145/3401956.3404228

the agent segments this motion into discretized gestures [9]. The agent's ability to intelligently respond to gestures and the user's movements comes from its pre-processing and clustering pipeline [14]. After a gesture is detected, it is pre-processed into a lower-dimensional subspace and then clustered with learned gestures that are "similar" to the novel gesture. This enables *LuminAI* to respond to the user's movements quickly with a contextually relevant gesture by selecting at random some gesture from the cluster.

The ability to iteratively tweak and refine the design of the pre-processing pipeline and clustering algorithm is key to improving the relevance of *LuminAI*'s dance moves and creating a more engaging user experience. However, evaluation of whether the pipeline improves *LuminAI*'s ability to co-create and improvise requires the ability to deeply, intuitively and precisely understand the specific behaviors and characteristics of the pre-processing pipeline. *LuminAI* as an exemplar use case demonstrates *MoViz*'s viability in addressing the difficulty of understanding the properties of "black-box" algorithms, a challenge not unique to *LuminAI*.

3 CHALLENGES AND SOLUTIONS FOR VISUALIZING MOCAP DATA

The most immediately apparent issue when visualizing motion capture data is its high dimensionality. The Kinect 2 motion sensor uses a frame-based system for tracking and recording geometric information, such as a cartesian coordinate vector or a quaternion vector, from selected "joints" of the human body. Consequently, the space complexity of motion capture data is expected to scale polynomially [14]. There are a variety of theoretical and pragmatic concerns that are born from using this arrangement, such as overfitting [14], a particularly relevant phenomenon that adversely affects the performance of machine learning models. Furthermore, high dimensional multivariate data is difficult to visualize.

Pre-processing is standard practice for working with high dimensional data and constitutes a pre-emptive "treatment" of the data before it is fed into the main model [4, 13]. Dimensionality reduction is a subset of pre-processing that focuses on decreasing the dimensionality of data while preserving its salient properties [4, 13]. Principal Components Analysis (PCA) and keyframe extraction are often used in dimensionality reduction for MoCap data [13]. The former identifies linear combinations of dimensions that best express linear variance whereas the latter isolates individual frames from a gesture that best convey its important qualities [4].

Motion capture data is information dense at both micro and macro scales. Bernard et al. specifies three different levels of granularity: features, single objects and groups of objects [4]. In the micro scale, we consider the fine-grained and precise movements of individual limbs in each gesture as a meaningful way to differentiate between them. On a macro scale, we may consider a collection of gestures' perceived overall shape and style of motion to be the key determinant in differentiation. As such, another difficulty with visualizing MoCap data is managing different levels of granularity without causing information overload.

One solution for alleviating this problem could be interactivity [4, 13]. By letting the user customize the interface for a desired level of granularity, we limit the amount of information loss incurred with a "one size fits all" abstraction level. Indeed, interactive visualizations afford the user flexibility in applying a single tool to different fields such as "education, healthcare, movement annotation, dance, and other creative applications" [13]. However, customizability will likely increase an interface's complexity and pose a barrier to entry for new users [13].

4 EXEMPLAR VISUALIZATION TOOLS

Several papers provide a comprehensive review of existing visualization tools for MoCap data [4, 13]. Some of these tools have designs and purposes are similar to that of *MoViz*. *Motion Map* created by Sakamoto et al. proposes a graphical representation of a MoCap dataset using a self-organizing map and visually resembles a two-dimensional grid embedded with icons showing an avatar holding some static pose from a gesture [18]. Bernard et al. also uses static icon representations, or a "glyph", for gestures in his work on *MotionExplorer*, an "interactive dendrogram visualization" that supports "exploration of a hierarchical clustering" [5]. Hierarchical clustering produces hierarchies of clusters, whose display is user customizable per the desired level of granularity. *GestureAnalyzer* by Jang et al. also employs "a hierarchical clustering algorithm to aggregate similar gestures into several groups" [10]. The main user interface of *GestureAnalyzer* is similar to Bernard et al.'s in that it provides a visualization of the hierarchical structure of the clusters with the level of granularity being user-customizable [10].

5 MOTIVATIONS FOR *MOVIZ*

MoViz was designed for the evaluation of a pre-processing pipeline on motion capture data in the context of large and diverse datasets. To accomplish this, a robust and accurate overview that allows for rapid intuition about individual gestures and the clusters they belong to is necessary. The exemplar visualizations we discussed rely predominantly on static representations of gestures. The intricacies and subtleties of dance motions are not well conveyed with icons. Furthermore, the dataset we are working with is much more varied and diverse than the ones used in the exemplar visualizations, which were focused more on a micro-level analysis of a smaller selection of gestures. We designed *MoViz* to be as intuitive as possible yet powerful with a minimally intrusive user interface to encourage the embodied exploration of motion capture data. Other works have shown that embodied interfaces provide substantial benefits for interaction design and information visualization [7, 12, 17]. We have not encountered any visualizations that comprehensively address our aforementioned objective in an embodied context.

6 *MOVIZ* DESIGN

In this section, we summarize the characteristics, behaviors and user interface of *MoViz* followed by a discussion of data visualization design principles incorporated in its design. The visualization tool is currently viewed on a computer display with keyboard and mouse controls, although we are working on adapting the tool to work with a virtual reality (VR) headset in order to facilitate a more embodied visualization experience.

6.1 Design Overview

We refer to the main screen that the user spends most of their time interacting with as the overview screen, which consists of a black

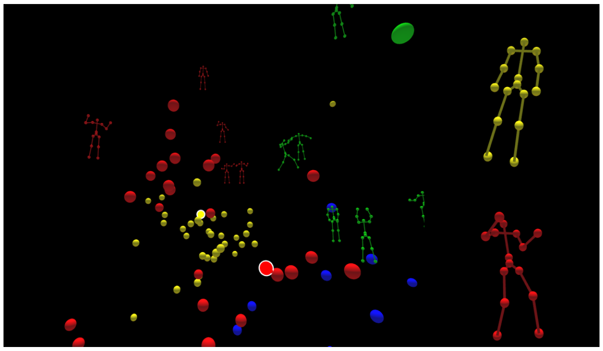

Figure 1: *MoViz* interface. When zoomed out, gestures appear as colored spheroids in clusters. Animations of gestures closer to the user in 3D space are displayed. The user can select two gestures to compare, which are shown on the right side of the screen.

void populated with numerous multi-colored spheres (Fig 1). Each sphere–or gesture point–represents a gesture. After a gesture is preprocessed, it is returned as a three-dimensional vector. The position of the corresponding gesture point is determined by interpreting this vector as a Cartesian coordinate in 3D space. A gesture point's color encodes its cluster assignment. Because we use K-means for clustering the gestures, we expect clustered datasets to ideally yield roughly spherical clusters (due to K-means' property of minimizing within-cluster distances) [1].

The camera position is controlled by the user using six degrees of motion input using keyboard keys and the mouse. By using the two together, the user can easily traverse the 3D space and decide whether they want to focus on one smaller region of the clustering or zoom outwards and get a broad overview. As the user zooms in close to a gesture it is replaced with a small avatar that is animating the gesture. This animation display allows users to inspect multiple clustered gestures of interest at once. It also helps prevent the gesture point from obscuring points behind it, as the avatar has a much lower surface area than the sphere.

These animated avatar displays are of the same color as the gesture points and will play the corresponding motion capture animation on loop until the user leaves the threshold distance. This threshold behavior is implemented using a spherical collider centered on the camera's position, where gesture points that are within the collider are activated to play the animation. The size of this spherical collider–or the threshold distance–can be controlled by the user using the scroll wheel. The size of the gesture points can also be scaled by the user using the keyboard. Making these gesture points very large allows them to better convey the general shape of a cluster, whereas making them smaller allows for easier in-cluster exploration and comparison.

There is also a "detail" overlay on top of the "overview" screen. By pressing the "alt" key, the user can unlock the mouse from controlling the camera and is free to move the cursor around. By clicking on different gesture points, the user controls the two small sub-views and the avatars displayed, located on the bottom left and right of the screen. Screenshots of *MoViz* in this paper (e.g. Fig 7) have the subviews re-positioned to occupy the right third of the screen for presentation clarity. Left (right)-clicking on a gesture point allows the user to view the associated motion capture data on the bottom left (right) corner of the screen. These gestures are shown in the detail overlay until new gestures are selected. The selected gesture points are identified with a white aura around them in the overview screen (Fig 1).

6.2 Design Principles

We utilized several information visualization design principles in the development of *MoViz* to design a system that is intuitive, informative, and accurate in its portrayal of data. The first set of these principles are outlined in Scheiderman's mantra for information visualization design: "overview first, zoom and filter, then details on demand" [19]. This phrase suggests that a tool should first present the user with a comprehensive overview of the dataset, then encourage the user to identify a subset of data they are interested in, and finally display details regarding this subset per user demands.

Our data visualization abides by the design considerations put forth by Schneiderman. The overview screen provides a high-level overview of the data that summarizes the shape of clusters and the associated distribution of gesture points. By controlling the positioning of the camera, the user can interactively and iteratively display a subset of gesture points that they're interested in. The mouse clicking and detail overlay both provide details on demand and assist in the zoom and filter step by making it easier to investigate the individual gestures of a cluster without changing the current subset of points on display. Being able to change the activation threshold for gesture animation display also aids in facilitating "details on demand". Making all the points display their avatars allows the user to compare individual animations within a subspace and gain insight to what kind of gestures characterize it. Showing only the spherical gesture points gives the user knowledge about the shape of this subset of gesture points. This feature is augmented by allowing users to scale the gesture points or avatars up or down on demand.

MoViz also adheres to several information visualization design principles suggested by Tufte [21]. Tufte's conception of *graphical integrity* emphasizes "telling the truth"—visual representations of data should neither over nor under-represent its effects and phenomena [21]. Graphical representations of numbers and objects must be directly proportional and commensurate with the data's quantitative elements. *MoViz* ensures that the position of the gesture points is linearly proportional to their respective output values after pre-processing. Tufte's *data-ink ratio* is a principle claiming that effective information visualizations minimize the amount of 'ink' used while maximizing the amount of meaningful data conveyed [21]. Our implementation accomplishes this by displaying only colored gesture points and animated avatars amidst black space. We chose to omit any axes or axis markings since gesture point positions are determined by PCA, which often produces output values with no consistent interpretable meaning. Any relevant information regarding a point's position is to be inferred by the user from neighboring points. Any emergent behavior regarding the semantic meaning of axes values is nonetheless preserved by this approach while avoiding visual noise. This design choice also avoids

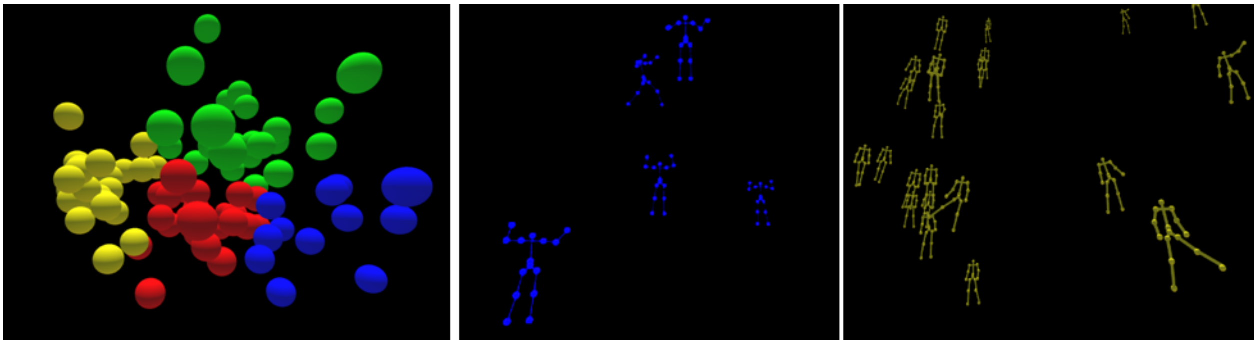

Figure 2: Left - Visualization of Temporal Clustering pipeline results in MoViz. Center - Zoomed in view of the bottom right blue cluster, consisting of arm movements with elbow joints above the shoulder. Right - Zoomed in view of the leftmost yellow cluster, consisting primarily of tightly clustered leg movements.

chartjunk, which is described by Tufte as unnecessary illustrations and graphical effects [21].

7 PIPELINE EVALUATION

We applied *MoViz* to the comparison of three gesture clustering pipelines under consideration in *LuminAI* to demonstrate a use case. The dataset we evaluated these pipelines on consists of 105 gestures recorded over an interval of three days by researchers in our lab. The dancers were not given any instruction on what type of gesture to perform, resulting in few directly "similar" gestures in this dataset, thus serving as a good practical example. Upon examination, it is clear that many gestures emphasize a particular limb or half of the human body, such as the left arm or the lower two legs. Such gestures constitute a rough majority of the dataset; the remaining minority had movements and motions which could not be easily localized to one part of the body. We also noticed that gestures involving waving motions with the arms or the raising and lowering of the knees were particularly common. The vast majority of the gestures were performed with the body facing the *Kinect*, with a few gestures where the body faced the side or rotated along the vertical axis.

7.1 Temporal Clustering Pipeline

7.1.1 Temporal Clustering Pipeline Description. The first pipeline developed for pre-processing and clustering motion capture data uses a technique called *Temporal Clustering* to identify a user-specified count of keyframes to lower gesture dimensionality. Temporal Clustering works by using dynamic programming to optimize locations of various consecutive and contiguous partitions of a gesture. The metric being optimized is referred to in Yang et al. as the "within segment sum of squared error" [22], a quantitative measure of how much each individual frame within the partition deviates from the mean frame, i.e. the average of all the frames in a partition. This minimizes the variance within the partition and indirectly maximizes the variance between the mean frames of different partitions. The set of keyframes is extracted from the partitions by finding their *mean frame.* As such, the number of partitions is equal to the number of user-specified keyframes.

The pipeline's next step for dimensionality reduction involves extracting scalar angles from important joints. In a typical gesture frame, three sets of Cartesian coordinates might be used to describe the position of the shoulder, elbow and wrist joints. The Temporal Clustering pipeline reduces this set of coordinates to the angle held by the elbow. More formally, given three joints A, B and C, the system computes the angle ABC held by joint B where [ABC] defines a triangle. This approach to feature extraction build on Kim et al.'s work on motion data classification, in which the authors obtained impressive accuracy using a similar method [11]. Once this is done, the pipeline will have computed a fixed number of angles for each keyframe. The final step in dimensionality reduction is the application of Principal Components Analysis to these angles and, having specified three principal components, results in a dimensionality of three per gesture. The reduced gestures are then clustered using the K-means clustering algorithm [14]. See [14] for more details on the implementation of this pipeline.

7.1.2 Evaluation of Temporal Clustering Pipeline using MoViz. As expected, due to K-Means' tendency to reduce within-cluster distances, the clusters produced by the Temporal Clustering approach take on a roughly spheroid shape (Fig 2). The distribution of gestures in this subspace is roughly spheroid as well. Individual gestures are somewhat well spread out and spaced. The few outliers are within reasonable distance of the "center mass".

Of particular note is the yellow cluster (shown in more detail on the right in Fig 2). It is unusually dense compared to the red, green and blue clusters. It constitutes 37% of the gestures in the dataset and consists mostly of leg motions.

As hypothesized in previous work on *LuminAI* [14], this uncharacteristically tight grouping is likely the result of the lowered flexibility in the lower half of the body (at least for researchers in our lab, who are not trained dancers). Specifically, the degree of freedom for the shoulder joint, for example, is much greater than that of the knee joint, and as the Temporal Clustering pipeline relies on the angles occupied by certain joints, it is expected that exclusive leg motions would be harder to differentiate.

The blue cluster on the bottom right in Fig 2 is noticeably sparser than the other clusters. Using *MoViz*, we zoomed into this cluster (center, Fig 2) and found that it was composed not only of motions

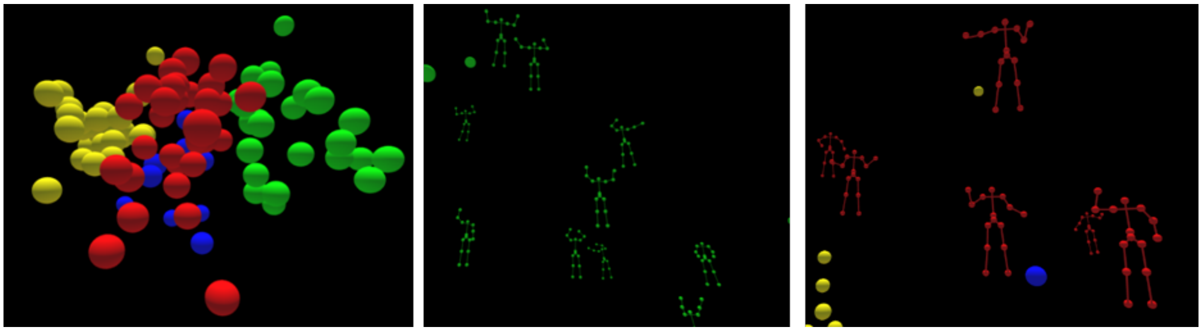

Figure 3: Left - Visualization of Limb Centroid pipeline clusters using MoViz. Center - Zoomed in view of the green cluster containing arm movements where the elbow is above the shoulders. This cluster similar to the blue cluster generated from Temporal Clustering. Right - Zoomed in view of a red cluster which contains arm motions where the elbow joint is below the shoulder.

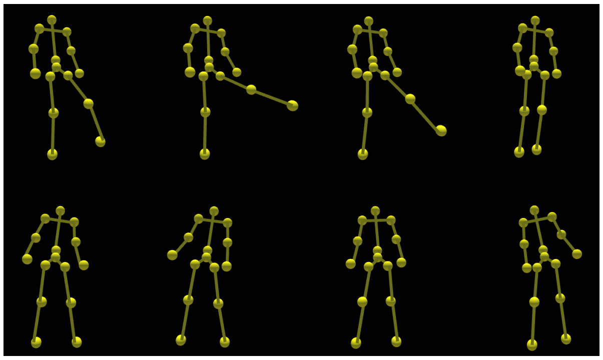

Figure 4: Selected frames from two gestures that demonstrate the Temporal Clustering pipeline's difficulties with separating hip and leg motions.

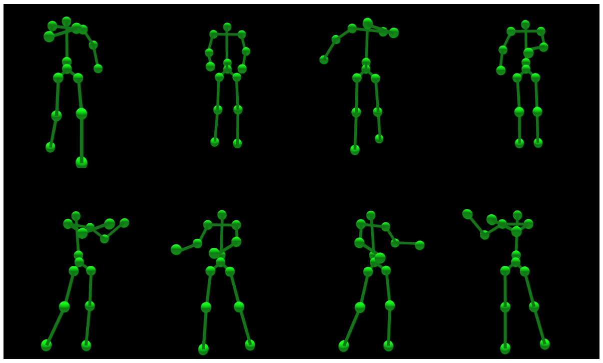

Figure 5: Selected frames from two gestures with similar rhythmic hand swaying motions.

that involve the movement of both arms, but also that these motions all position the elbow joint above the shoulder. Prior work on *LuminAI* noted that the Temporal Clustering pipeline identifies, without supervision, the limbs of the human body that share a similar pose and uses that information to place gestures into their appropriate clusters [14]. This phenomenon is made immediately apparent via interaction with *MoViz*.

Another interesting property of the Temporal Clustering pipeline can be observed by comparing individual gesture pairs using the "detail overlay"/gesture viewer in *MoViz*. As shown in Fig 5, two gestures were found in close proximity to each other and depict a regular "swinging" of both arms from one side to another. We did not expect the Temporal Clustering pipeline to identify gestures that were rhythmically similar. We hypothesize that this phenomenon is born from Temporal Clustering's variance maximizing property. In retrospect, the definition of rhythm as the speed at which a dancer regularly transitions from one stage in the gesture to another is intuitively similar to the process through which Temporal Clustering computes optimal partitions [22].

The pair of gestures in Fig 4 reveals unexpected behavior exhibited by the Temporal Clustering pipeline, caused by the implementation of joint angles. The gesture shown above in Fig 4 moves its leg up and down by rotating the hip joint. The gesture shown below in Fig 4 sways left to right by rotating the upper body. The latter gesture causes a similar rotation in the hip joint, which has caused it to be considered similar to the former gesture. Intuitively, however, we can tell that the performer of the former gesture deliberately emphasized the movement and rotation of the left leg. Because the gestures are, in terms of their joint angles, similar, their perceptual differences lead to serendipitous insight into the nature of human motion.

7.2 Limb Centroid Pipeline

7.2.1 Limb Centroid Pipeline Description. As noted in our prior work, the computation of optimal partitions using Temporal Clustering resulted in *LuminAI* experiencing some performance difficulties[14]. This motivated our investigation of a new approach that would discretize a gesture into a user-specified number of equally sized and evenly spaced contiguous and consecutive partitions. From each of these partitions, the pipeline will compute what Balc et al. refers to as a Limb Centroid by first finding the mean frame of the partition [2]. Once this has been found, the pipeline computes the centroid of a limb—using the shoulder, elbow and wrist as an example again—by calculating the mean cartesian coordinate of the three joints by adding them together and dividing by three. This is done for the left arm, right arm, left leg, right leg, and spine. By the end of this process, there will be five Limb Centroids, which are then fed into PCA to obtain a three-dimensional output vector for the input gesture. The final set of reduced gestures is again clustered using a K-Means clustering algorithm.

7.2.2 Evaluation of Limb Centroid Pipeline using MoViz. As shown in Fig 3, the clusters computed using the Limb Centroid pipeline are more uniform and well spread out than those of the Temporal Clustering pipeline. Though the yellow cluster is still comparatively dense, it is not significantly more so than the red or green clusters. The shape of the clusters produced is not as spherical compared to the Temporal Clustering pipeline, but the size and space occupied appear to be roughly equal across the yellow, red, and green clusters. Interestingly, the blue cluster is relatively underpopulated, as it was with the Temporal Clustering pipeline.

Re-orienting the camera in *MoViz* reveals a few outlier gestures. Upon closer examination in *MoViz*'s gesture subview, it becomes clear that these selected gestures exhibit high amounts of lateral translation. Due to the Limb Centroid pipeline relying on the Cartesian coordinates of joints to determine the position of Limb Centroids, the computed Limb Centroids also experience a degree of lateral translation not present in other gestures, thus their "outlier" positioning.

The Limb Centroid approach also exhibits the emergent property, mentioned in 7.1.2, of grouping gestures together based on individual limb positioning. However, the Limb Centroid approach produces clusters that are more visually coherent, with distances within a cluster well reflecting similarity. For example, in the green cluster shown in the center in Fig 3, the gestures in the top left have the elbow joints horizontally aligned with the shoulder joint, whereas the gestures below towards the middle-center raise the elbow joints above the shoulder. This property of semantically significant information being encoded in micro scale positioning of gestures within a cluster is exhibited across all clusters (Fig 3). The issue the Temporal Clustering approach experienced with leg motions, in the yellow cluster, is alleviated as Limb Centroids do not rely on joint angles (Fig 6). Interestingly, both Temporal Clustering and Limb Centroid approaches place leg motions clusters on the left of the space in the yellow cluster.

In the gesture pairing shown in the center in Fig 6, the Limb Centroid approach can be shown abstracting away from the precise orientation of the shoulder joints and elbow joints. Though the pose of the rest of the body is somewhat different in the low gesture, as shown by its hip movement and arm orientation, both gestures are recognizably waving motions. This example demonstrates the Limb Centroid's ability, like that of the Temporal Clustering approach, to abstract away from the specific orientation of active body parts and to ignore minute differences in static body parts.

As we were exploring the gestures using *MoViz*, we notice an unexpected gesture pairing that suggests a means for future improvement. This gesture pairing—shown on the right in Fig 6—consists of a simple waving motion paired with gesture in which the user leans backwards with both arms oriented upwards and shifting left to right. Though the right arms of both gestures are in a similar pose, it was unexpected t hat they would be grouped so closely. We believe this is caused by the way that the Limb Centroid approach calculates zero-meaning of centroids.

Specifically, for any collection of centroids that describes the pose within some partition of the original gesture, the system adds or subtracts the same vector to all centroids such that the average position of these centroids is zero. This approach might not preserve the information about the relative positions of each Limb Centroid in relation to some fixed frame of reference, such as the hip, leading to unexpected results. Indeed, because of zero-meaning, even if a Limb Centroid remains static, should another Limb Centroid move, the former Limb Centroid's position might be affected as well. An example of this behavior is described as follows: supposing that we extend the left arm outwards from the center body while leaving all other centroids unmoving, then after zero-meaning, an offset would be applied to all other Limb Centroids and altering their position relative to the previous partition's centroids, creating a misrepresentation of the motion. A solution to this issue might involve applying a vector offset to the gesture before centroid calculation such that the hip joint, for example, is always at [0,0,0], thus preserving the limbs' relative position information.

7.3 Difference Vector Pipeline

7.3.1 Difference Vector Pipeline Description. The final pipeline we tested, for each consecutive and contiguous partition, extracted and then concatenated together a covariance matrix and difference vector from each of the centroids. The difference vector is found by finding the average amount of translation, in the form of a three-dimensional vector, the centroid would experience going from one frame to the next within the partition. This pipeline design was motivated by an interest in expressing the "magnitude" and "quality" of movement in more detail. It was hypothesized that using a covariance matrix per Limb Centroid, along with the difference vector, would better express the semantic characteristics of that Limb Centroid's motion.

7.3.2 Evaluation of Difference Vector Pipeline using MoViz. The unusual shape of the clusterings generated using this approach is immediately apparent (Fig 7). We were not able to find a geometric primitive that describes its shape adequately. There are several obvious outliers, the most obvious of which is a single green gesture on the right in Fig 7 which has deviated so far from the central mass that it is its own cluster. The blue cluster shown on the left in Fig 7 is extremely dense. Furthermore, the positionings of individual gestures do not appear to reflect any sort of consistent similarity, as gestures which are placed together are frequently wildly different. For example, in the gesture shown on the right in Fig 7, the gesture depicted on the top mainly moves the left arm, specifically extending the elbow joint, whereas the one on the bottom raises the left knee by rotating the hip. Our exploration of this space using *MoViz* has suggested that this poor behavior is caused by the same zero-meaning issue the Limb Centroid pipeline experienced. Specifically, the positions of Limb Centroids at any given frame are inter-related. A translation of one centroid will also affect the relative position of all other centroids after zero-meaning, resulting in unreliable covariances and poor differentiation between individual limb activity.

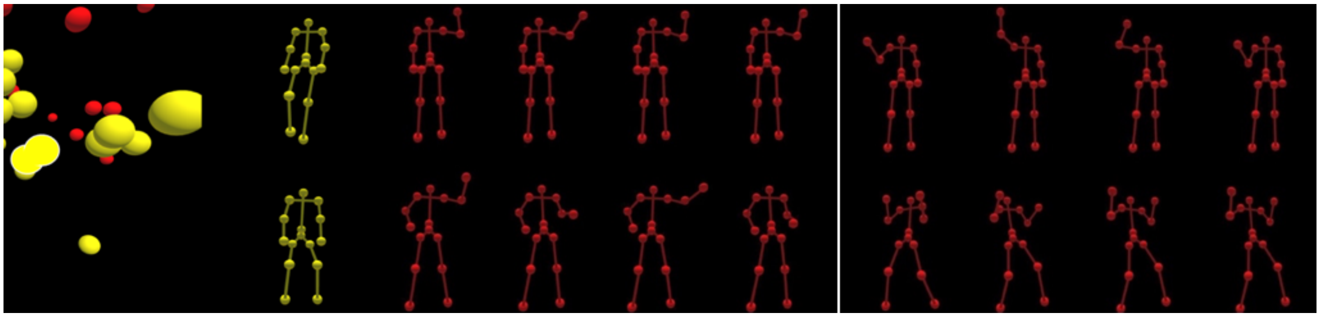

Figure 6: Left - These two gestures from the yellow cluster are close to one another relative to distances within the cluster and are reasonably similar, both depicting bent knees. Center - Selected frames from two gestures that both depicting a waving motion with the left arm, but different below-the-neck poses. Right - Selected frames from two gestures that depict arm(s) in a bent position, but wildly different below-the-neck poses and motion.

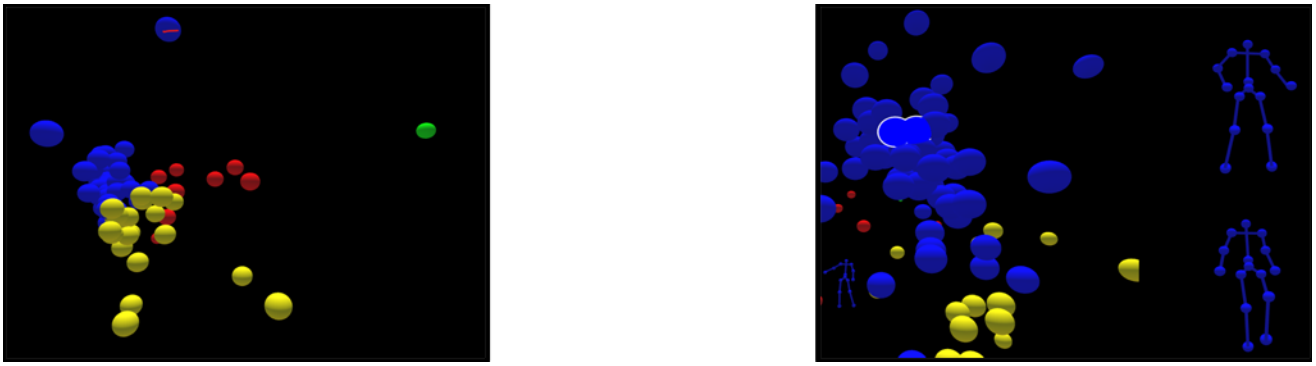

Figure 7: Left - A view of the distribution of points produced by the Difference Vector pipeline. Right - The blue cluster is extremely dense. The two highlighted gestures do not appear to be similar.

8 FUTURE WORK

One of the primary advantages of the Temporal Clustering pipeline was its invariance to the global positions of the avatar. However, it did this by retaining only angular descriptions of the avatar’s pose and discarding all positional data. The proposed modifications to the Limb Centroid pipeline achieves this invariance by centering the positions of the various joints in the avatar relative to some bone, likely the hip or base of the spine.

In looking towards future directions for research, we believe it is important to find a way to retain this previously discarded translation information while also making our system invariant to changes in *Kinect* sensor location—i.e. the translation information of an identical gesture being performed across different *Kinect* sensor configurations and placements should remain consistent.

Though the Limb Centroid approach demonstrates efficacy in identifying which limbs of the avatar are active and have a roughly similar position, it is clear to us that gestures and dances may also emphasize the movements of the hands and fingers, or distal joints. We would like to incorporate steps that extract additional information about the pose of hands and feet into our future pipeline designs.

Furthermore, we believe that research into using quaternions, which describe the rotation and orientation of different joints, instead of Cartesian coordinates or scalar angles derived from Cartesian coordinates will prove to be a fruitful avenue for future work, such as clustering by joint angle, angular velocity and angular acceleration. There are preexisting measures for distances between quaternions which we believe can enable compatibility with techniques such as Temporal Clustering or K-Means.

9 CONCLUSION

In this paper, we present the design of *MoViz*, a novel visualization tool that enables comparative qualitative evaluation of otherwise “black-box” algorithms for pre-processing and clustering large and diverse motion capture datasets. We applied *MoViz* to the evaluation of three different gesture clustering pipelines used in the *LuminAI* improvisational dance system. This evaluation revealed features of the pipelines that may not otherwise have been apparent, suggesting directions for iterative design improvements. It also allowed us to identify which pipelines worked well for our dataset (e.g. Limb Centroid Pipeline) and which performed poorly (e.g. Difference Vector Pipeline). This use case demonstrates the potential for this tool to be used by researchers and designers in the field of movement and computing seeking to better understand the algorithms used to make sense of large, diverse motion capture databases.

ACKNOWLEDGMENTS

We would like to thank the other researchers and developers on the *LuminAI* team for their hard work and guidance, including Swar Gujrania, Meha Kumar, and Cassandra Naomi. We would also like to thank Mark Riedl for his guidance and suggestions.

REFERENCES

[1] [n. d.]. K-means. ([n. d.]). https://nlp.stanford.edu/IR-book/html/htmledition/k-means-1.html

[2] Koray Balci and Lale Akarun. 2008. Clustering poses of motion capture data using limb centroids. In *2008 23rd International Symposium on Computer and Information Sciences*. IEEE, 1–6.

[3] Adrian Ball, David Rye, Fabio Ramos, and Mari Velonaki. 2011. A comparison of unsupervised learning algorithms for gesture clustering. In *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 111–112.

[4] Jürgen Bernard, Anna Vögele, Reinhard Klein, and Dieter W. Fellner. 2017. Approaches and Challenges in the Visual-interactive Comparison of Human Motion Data. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SciTePress, 217–224. https://doi.org/10.5220/0006127502170224

[5] Jürgen Bernard, Nils Wilhelm, Björn Krüger, Thorsten May, Tobias Schreck, and Jörn Kohlhammer. 2013. MotionExplorer: Exploratory Search in Human Motion Capture Data Based on Hierarchical Aggregation. *IEEE Transactions on Visualization and Computer Graphics* 19 (Dec. 2013), 2257–66. https://doi.org/10.1109/TVCG.2013.178

[6] Amit Bleiweiss, Dagan Eshar, Gershom Kutliroff, Alon Lerner, Yinon Oshrat, and Yaron Yanai. 2010. Enhanced Interactive Gaming by Blending Full-body Tracking and Gesture Animation. In *ACM SIGGRAPH ASIA 2010 Sketches (SA '10)*. ACM, New York, NY, USA, 34:1–34:2. https://doi.org/10.1145/1899950.1899984

[7] Paul Dourish. 2001. Where the Action Is: The Foundations of Embodied Interaction. 256.

[8] Bernadette Hecox, Ellen Levine, and Diana Scott. 1976. Dance in physical rehabilitation. *Physical therapy* 56, 8 (1976), 919–924.

[9] Mikhail Jacob. 2017. Towards Lifelong Interactive Learning For Open-ended Embodied Narrative Improvisation. In *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*. ACM, 502–507.

[10] Sujin Jang, Niklas Elmqvist, and Karthik Ramani. 2014. GestureAnalyzer: Visual Analytics for Pattern Analysis of Mid-Air Hand Gestures. In *Proceedings of the 2nd ACM Symposium on Spatial User Interaction (SUI '14)*. Association for Computing Machinery, New York, NY, USA, 30–39. https://doi.org/10.1145/2659766.2659772

[11] Dohyung Kim, Dong-Hyeon Kim, and Keun-Chang Kwak. 2017. Classification of K-Pop dance movements based on skeleton information obtained by a Kinect sensor. *Sensors* 17, 6 (2017), 1261.

[12] Scott R. Klemmer, Björn Hartmann, and Leila Takayama. 2006. How bodies matter: five themes for interaction design. In *Proceedings of the 6th ACM conference on Designing Interactive systems - DIS '06*. ACM Press, University Park, PA, USA, 140. https://doi.org/10.1145/1142405.1142429

[13] William Li, Lyn Bartram, and Philippe Pasquier. 2016. Techniques and Approaches in Static Visualization of Motion Capture Data. In *Proceedings of the 3rd International Symposium on Movement and Computing (MOCO '16)*. Association for Computing Machinery, Thessaloniki, GA, Greece, 1–8. https://doi.org/10.1145/2948910.2948935

[14] Lucas Liu, Duri Long, Swar Gujrania, and Brian Magerko. 2019. Learning Movement through Human-Computer Co-Creative Improvisation. In *Proceedings of the 6th International Conference on Movement and Computing (MOCO '19)*. Association for Computing Machinery, Tempe, AZ, USA, 1–8. https://doi.org/10.1145/3347122.3347127

[15] Brian Magerko, Christopher DeLeon, and Peter Dohogne. 2011. Digital Improvisational Theatre: Party Quirks. AAAI Press, Reykjavík, Iceland.

[16] Stephen O'Hara, Yui Man Lui, and Bruce A Draper. 2011. Unsupervised learning of human expressions, gestures, and actions. In *Face and Gesture 2011*. IEEE, 1–8.

[17] Dheva Raja, Doug A. Bowman, John Lucas, and Chris North. 2004. Exploring the Benefits of Immersion in Abstract Information Visualization. In *In proceedings of Immersive Projection Technology Workshop*.

[18] Yasuhiko Sakamoto, Shigeru Kuriyama, and Toyohisa Kaneko. 2004. Motion map: Image-based retrieval and segmentation of motion data. *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Jan. 2004), 259–266. https://doi.org/10.1145/1028523.1028557

[19] Ben Shneiderman. 1996. The eyes have it: A task by data type taxonomy for information visualizations. In *In Ieee Symposium on Visual Languages*. 336–343.

[20] Tanu Srivastava, Raj Shree Singh, Sunil Kumar, and Pavan Chakraborty. 2017. Feasibility of Principal Component Analysis in hand gesture recognition system. *arXiv preprint arXiv:1702.07371* (2017).

[21] Edward Rolf Tufte. 1983. The visual display of quantitative information. https://doi.org/10.1097/01445442-198507000-00012

[22] Yang Yang, Hubert P. H. Shum, Nauman Aslam, and Lanling Zeng. 2016. Temporal clustering of motion capture data with optimal partitioning. In *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry - Volume 1 (VRCAI '16)*. Association for Computing Machinery, Zhuhai, China, 479–482. https://doi.org/10.1145/3013971.3014019