# Learning Movement through Human-Computer Co-Creative Improvisation

Lucas Liu
Georgia Institute of Technology
Atlanta, Georgia
lucasliu@gatech.edu

Duri Long, Swar Gujrania
Georgia Institute of Technology
Atlanta, Georgia
{duri,swar.gujrania}@gatech.edu

Brian Magerko
Georgia Institute of Technology
Atlanta, Georgia
magerko@gatech.edu

## ABSTRACT

Computers that are able to collaboratively improvise movement with humans could have an impact on a variety of application domains, ranging from improving procedural animation in game environments to fostering human-computer co-creativity. Enabling real-time movement improvisation requires equipping computers with strategies for learning and understanding movement. Most existing research focuses on gesture classification, which does not facilitate the learning of new gestures, thereby limiting the creative capacity of computers. In this paper, we explore how to develop a gesture clustering pipeline that facilitates reasoning about arbitrary novel movements in real-time. We describe the implementation of this pipeline within the context of *LuminAI*, a system in which humans can collaboratively improvise movements together with an AI agent. A preliminary evaluation indicates that our pipeline is capable of efficiently clustering similar gestures together, but further work is necessary to fully assess the pipeline's ability to meaningfully cluster complex movements.

## CCS CONCEPTS

• **Applied computing** → **Performing arts**; **Media arts**; • **Human-centered computing** → *Human computer interaction (HCI)*;

## KEYWORDS

clustering, movement, dance, machine learning, lifelong machine learning, co-creative, pre-processing, dimensionality reduction, dynamic programming, motion capture, Kinect

**ACM Reference Format:**
Lucas Liu, Duri Long, Swar Gujrania, and Brian Magerko. 2019. Learning Movement through Human-Computer Co-Creative Improvisation. In *6th International Conference on Movement and Computing (MOCO '19), October 10–12, 2019, Tempe, AZ, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3347122.3347127

## 1 INTRODUCTION

Humans collaboratively improvise movement in situations ranging from dance performances to pretend play to sports games. Computers with the ability to participate in these collaborative movement improvisations could have an impact on a variety of application domains, including improving naturalistic procedural animation in game environments [3], fostering human creativity in gesture-based domains like dance or theater [8, 9], and creating more engaging contexts for physical therapy and training [5].

One particular domain that has made advances in understanding embodied human-computer improvisation is the study of co-creative AI agents. A variety of recent research investigates how humans and computers may be able to create together in gesture-based domains including co-creative dance [9] and collaborative movement improvisation [8]. However, an obstacle that is pervasive throughout these projects is that humans and their AI collaborators bring significantly different sets of experiences to the co-creative interaction. Humans possess a vast amount of real-world knowledge, in contrast to AI agents, which draw their knowledge from comparatively small datasets. This contrast creates an imbalance during a co-creative interaction, since the humans are required to give more than they receive.

Unfortunately, many embodied creative domains like dance, pretend play, and theater are notable for their lack of large-scale, diverse, annotated datasets since motion-capture data can be time-consuming and expensive to collect. Agents capable of *lifelong learning* (c.f. [12]) are particularly well-suited for embodied creative domains since they can learn interactively from human collaborators without supervision. However, the agent needs some way of reasoning about newly learned gestures in order to respond intelligently to its human partner. One intuitive way to reason about gestures is based on their similarity, a technique that is frequently used in improvisation in a variety of domains, such as theater and jazz [11]. Discerning gesture similarity in movement improvisation requires the ability to both cluster gestures based on different metrics on-the-fly and identification of which cluster a gesture belongs to in real-time.

Most existing research on gesture understanding focuses on gesture *classification* (i.e. identifying and categorizing different clips of human motion) (e.g. [6]). However, this is not particularly useful for lifelong learning in creative domains, since human collaborators can perform a seemingly infinite number of novel gestures while classification systems try to label these gestures according to only a finite number of known categories. As a result, new gestures will not be incorporated into the agent's knowledge base, making it difficult for the agent to learn-through-interaction and thereby limiting its long-term ability to contribute to creative collaborations.

In contrast, a system capable of unsupervised gesture *clustering* would be able to learn novel gesture types. Such a system could compare novel gestures to previously seen gestures and add new gestures to existing clusters, learning through interaction. An

*MOCO '19, October 10–12, 2019, Tempe, AZ, USA*

ACM ISBN 978-1-4503-7654-9/19/10…$15.00
https://doi.org/10.1145/3347122.3347127

unsupervised gesture clustering system could also dynamically respond to novel gestures by drawing on past experiences and finding a similar gesture it has seen before without needing a preprogrammed label (in effect, creating its own knowledge of gesture categories rather than relying on pre-programmed knowledge). Systems capable of gesture clustering also have the potential to be domain-independent, whereas existing classification algorithms can often only classify gestures based on a very domain-specific set of categories.

There is some existing literature on unsupervised gesture clustering, though it is focused primarily on only hand gestures [2, 14, 15]. Work is still needed to understand how to approach unsupervised gesture clustering with a full-body skeleton, which differs significantly from hand motion both anatomically and in terms of gesture duration (i.e. hand gestures tend to be shorter in length with less freedom of movement than full-body gestures).

In this paper, we investigate the following central research question: *How can we implement a co-creative agent that can cluster arbitrary full-body motion data in real-time, thereby enabling the agent to draw on a breadth of learned experiences when responding to its human collaborator?* In the rest of the paper, we look at other related work in this area, discuss a particular use case for which we designed our clustering pipeline, detail the implementation of the pipeline, and discuss a preliminary evaluation of our pipeline, ending with conclusions and plans for future work.

# 2 RELATED WORK

## 2.1 Gesture Clustering Based on Skeletal Similarity

There is some existing research that has explored how to cluster gestures based on skeletal similarity. There is also some classification work that can inform our research on unsupervised gesture clustering. In this section we will highlight key takeaways from the existing work in this space.

The limited existing work on gesture clustering primarily uses k-means [16] as a clustering algorithm. Balci et al. use k-means to cluster individual poses/frames of human motion [1]. O'Hara et al. also use k-means to cluster video clips of human motion [10]. There are a variety of different motivations for using k-means for gesture clustering, including its computational efficiency [16].

One of the challenges with motion capture data is its high dimensionality, which can make running a clustering algorithm like k-means on unprocessed data intractable in terms of run-time (see Challenges for more detail) and prone to overfitting. Several existing projects use Principal Components Analysis (PCA) as a method for dimensionality reduction, suggesting that it is a suitable candidate for datatypes depicting human motion. Srivastava et al. apply PCA to video clips of human hand gestures as a pre-processing step for a classification algorithm [14]. Balci et al. also use PCA in an application that clusters motion poses [1].

Other techniques for dimensionality reduction have been explored in the literature as well. Kim et al. extract key joints from gestures as a way of reducing dimensionality, focusing on joints that have more impact on the visual appearance of the dance motion [6]. Yang et al. also propose a novel approach to reducing the dimensionality of large frame-based representations of the human body [17]. Their approach, called *temporal clustering*, takes a gesture of some number of frames and from them identifies a subset of frames, much less than the original, which well-approximate the gesture. We draw on these two approaches as well as PCA in the implementation of our pipeline.

While there is a variety of existing work that suggests useful strategies for clustering arbitrary motion data, none of them directly answer our research question. Kim et al. propose a pipeline for classifying Korean Pop style dance gestures, but as a result of being a classification and not a clustering pipeline, many aspects of their implementation rely on labeled data in order to function, which is not feasible in applications that require lifelong learning [6].

Balci et al. suggest an approach for reducing the dimensionality of the human skeleton that takes the average, or "centroid", of several joint positions of a limb in a skeleton and then attempts to cluster that processed data [1]. However, their chosen input data is not continuous and consists instead of still poses taken from a single recorded gesture. Our work attempts to cluster entire gestures rather than individual poses or still frames.

Finally, O'Hara et al. attempt to pre-process and cluster data recorded from various parts of the body such as the face or the hands, including full-body motions [10]. However, the algorithms used in their approach, namely Product Manifolds and Bag of Features, were designed explicility for video data and thus do not capture well the unique characteristics of motion capture data.

## 2.2 Non-skeletal Similarity Metrics

The previously discussed work is all heavily focused on understanding gestures in relation to the position of joints in the human skeleton. There is also research looking into how to compare gestures based on non-skeletal measures, such as Laban movement analysis [7], which is a framework created by dancer/choreographer Rudolf Laban that characterizes movement based on four different paradigms–*Body*, *Space*, *Effort*, and *Shape*. *Effort* (i.e. the intrinsic quality of a movement) is the element of this framework that has been explored the most by practitioners and researchers working in computational movement science. Existing research largely concerns itself with analyzing movements to discern the four parameters of *Effort–Time*, *Weight*, *Space* and *Flow*. This includes work focused on identifying parameters from features such as velocity, acceleration, joint position/orientation, and muscle tension and classifying movements accordingly (e.g. [4]). This is a different approach to understanding what defines two "similar" gestures. We seek to develop a pipeline for gesture clustering that can accommodate both skeletal similarity and other metrics of similarity, such as the metrics identified in the Laban effort system.

# 3 LUMINAI

*LuminAI* is an interactive art installation in which humans can collaboratively improvise movement with a virtual dance partner [9]. A Microsoft Kinect 2.0 depth sensor is used to detect the human participant's motion, which is visualized as a virtual "shadow" on a projection screen. Next to the shadow is a humanoid "agent", which dances by analyzing the participant's movement and responding with a movement that it deems to be similar in terms of parameters such as energy, tempo, or size (adapted from Viewpoints movement theory [9]). The agent interactively learns gestures from the participant as they dance together.

We used *LuminAI* as a context for developing an unsupervised gesture clustering pipeline because, while *LuminAI* is capable of lifelong learning, it simply remembers every gesture that it recognizes, and does not cluster gestures based on similarity. The current version of *LuminAI* can calculate certain similarity metrics between gestures (e.g. whether two gestures have the same tempo) using mathematical heuristics developed based on Viewpoints movement theory [9], but the current system is not capable of comparing gestures using other parameters such as visual similarity or other movement theories such as Laban movement analysis [7]. A gesture clustering pipeline would enable *LuminAI* to respond more intelligently to its human collaborators’ dance moves.

## 4 CHALLENGES

There were three main difficulties we encountered when trying to develop a pipeline that could cluster motion capture data during real-time participant interactions with *LuminAI*:

*Defining Meaningful Similarity:* The first challenge stems from the disconnect between quantitative similarity measures and how human beings perceive motion. A clustering pipeline that allows for meaningful co-creative experiences must present users with clusters that are not only quantitatively similar, but also visually and intuitively understandable . The challenge of defining what constitutes meaningful similarity is compounded by the many metrics that human beings use to interpret similarity (e.g. skeletal similarity vs. the Laban metrics discussed in Related Work). A similarity-based gesture clustering algorithm should be able to accommodate different similarity metrics depending on the context.

*Need for Dimensionality Reduction:* The second challenge we encountered when developing our gesture clustering pipeline was that conventionally recorded motion capture data using a frames-per-second approach caused the complexity of a motion to grow in polynomial time. In the *LuminAI* setup, the growth scaled with the number of features per frame by the total number of frames, scaled once more by the total number of motions in the knowledge-base. This more or less requires us to implement a pre-processing approach (which reduces the efficacy of use in real-time improvisation) or dimensionality-reduction to make the clustering tractable at larger motion-library sizes. In addition, data that is too high dimensional, like motion capture data, is extremely prone to overfitting and would adversely affect the accuracy of our clustering.

*Real-Time Response:* The third challenge we faced is that the dimensionality reduction steps must be efficient enough to run in real-time so that the pre-processing of novel gestures does not interfere with the system’s response time.

## 5 IMPLEMENTATION

The aforementioned challenges and related work [1, 6, 14, 17] informed our development of a three-stage pipeline for unsupervised gesture clustering of arbitrary full-body motion data. In this section, we provide an overview of the pipeline architecture, followed by a more detailed description of each stage in the pipeline.

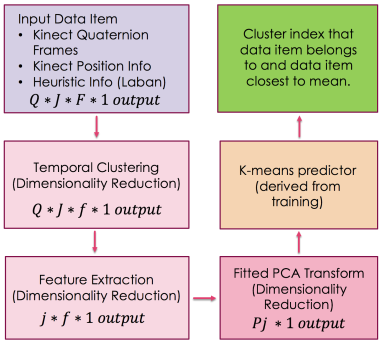
Input Data Item
• Kinect Quaternion Frames
• Kinect Position Info
• Heuristic Info (Laban)
Q * J * F * 1 output
Temporal Clustering (Dimensionality Reduction)
Q * J * f * 1 output
Feature Extraction (Dimensionality Reduction)
j * f * 1 output
Fitted PCA Transform (Dimensionality Reduction)
Pj * 1 output
K-means predictor (derived from training)
Cluster index that data item belongs to and data item closest to mean.

**Figure 1: Flow-chart of the real-time operation pipeline.**

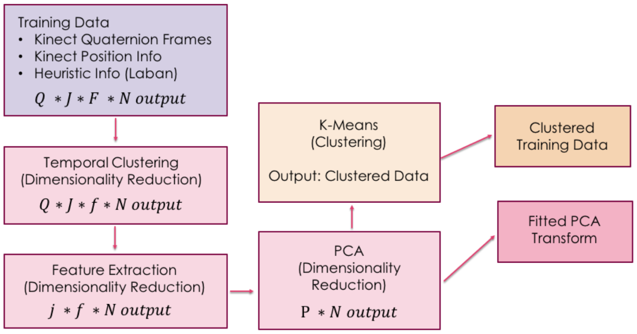

**Figure 2: Flow-chart of the training pipeline.**

### 5.1 Overview of Pipeline

Our proposed pipeline consists of two similar implementations, one for real-time operation (Fig. 1) and one for training (Fig. 2). The training component consists of three main parts: the pre-processing dimensionality reduction steps, the clustering and model fitting steps, and the export of a trained k-means model [16] with a fitted PCA transform model [13]. The pre-processing step reduces the dimensionality of motion data considerably using temporal clustering [17], then reduces dimensionality even further using a joint angle extraction technique [6]. Once this is applied to every item in the motion library, the data is then used in the model-fitting and clustering step, in which a PCA model is first fitted on the reduced motion library.

Once the PCA transform model is obtained, the dimensionality of the data is further reduced using the PCA model. Finally, the newly transformed data is clustered using a k-means model. The products of the model-fitting and clustering step are a PCA transform model, which can be used to apply PCA to novel data items, and a k-means model, which contains the clustering of the pre-processed motion library and can be used to place novel gestures in their appropriate clusters. In the final stage of the pipeline, these two models are exported for future use.

The pipeline running in real-time uses the pre-trained PCA transform model and k-means model and consists of three steps: motion recording, motion pre-processing, and motion clustering. In the first step, a participant is prompted to record themselves performing a gesture using the Microsoft Kinect 2.0 depth sensor. In the motion pre-processing step, the dimensionality of the novel gesture is reduced using temporal clustering, angle extraction, and the fitted PCA transform model. In the final step, the transformed gesture is placed into an appropriate cluster by the fitted k-means model. A gesture randomly selected from the target cluster of the novel gesture will be played back to the user (this step is specific to the *LuminAI* use case, in which we want the agent to respond with a gesture that is similar to the participant's gesture).

We decided to focus heavily on the principle of maximizing variance between different gestures when designing our pipeline. Temporal clustering, PCA, and k-means were chosen as a starting point from pre-existing papers [1, 6, 14, 17] specifically due to the way that all three incorporate elements of variance maximization in their design. In the remainder of this section, we will describe the implementation of each stage of the pipeline in more detail.

## 5.2 Input Data

The input data for the pipeline can consist of any feature vector where the geometric distance between any two feature vectors is a quantitative measure of the dissimilarity between them. This means that the pipeline can cluster gestures based on a feature vector consisting of joints-based skeletal data or a feature vector of other movement qualities such as *Time*, *Weight*, *Space*, or *Flow* [7]. We focus on joints-based skeletal data in this paper, but plan to incorporate Laban feature vectors in future work as an alternative way of understanding meaningful similarity between gestures.

The joints-based input data for the pipeline consists of gestures gathered using the Microsoft Kinect depth sensor–although this pipeline could be adapted to accommodate other motion capture devices such as a motion capture suit. The dimensionality of a single gesture recorded from the Kinect is *Q x J x F*, where Q is the dimensionality of the geometric information associated with each joint, J is the number of joints in a Kinect skeleton, and F is the number of frames in a gesture.

*Positional and Rotational Data:* Q is the dimensionality of the geometric information associated with each joint. The Kinect can record either the rotation of a joint or its position. In our implementation, we recorded the data using Cartesian coordinates, making the dimensionlity of Q three.

*Joints:* J is the number of joints in a Kinect skeleton. A single frame in a Kinect-recorded gesture consists of an abstract skeletal representation with a pose and orientation that approximates the human pose. The Kinect does this by identifying the "joints" on a human user, such as the knee or the elbow, and where they are in R3 space. As a result, the "skeleton" is composed of a set number of joints J. Joints in this skeleton follow a tree structure so as to emulate a human's physiological makeup. A hand is the child of an elbow, which in turn is the child of the shoulder.

*Frames:* F is the number of frames in a gesture, which can be considered "still frames" of movement that approximate its kinetic and spatial qualities. Strung together quickly enough, frames can emulate continuous motion. The Kinect sensor captures movements at 24 FPS (frames-per-second).

## 5.3 Temporal Clustering

The size of F can easily grow into the hundreds with longer gestures, making a reduction in the number of frames necessary in order to facilitate real-time data processing. The objective of *temporal clustering* [17] is to find a user-specified number of "keyframes" that best approximate the input motion. Temporal clustering achieves this by expressing the problem of finding representative "keyframes" as optimizing the placements of consecutive contiguous partitions. Each partition is evaluated using a measure described in Yang et al. as the "within-segment sum of squared error" which quantifies how "different" the frames in each partition are from the partition's mean frame [17]. This creates partitions consisting of frames that are as similar to one another as possible, thus indirectly maximizing the difference or variance between one partition and all other partitions. In order to make this approach computationally tractable, Fisher's optimal partition algorithm [6], a dynamic programming approach, is used to identify these partitions, and an average of all the frames in one partition is used to produce a "keyframe" [1].

Suppose we are given an gesture that is 300 frames in length. We can reduce the number of frames in the representation to less than one twentieth of its original size by setting the number of keyframes to 15. In addition, the implementation of temporal clustering makes the pre-processing approach invariant to the length of each input gesture, as the number of frames output is user-specified as the keyframe number. This is particularly important for our system as not all recorded gestures are of equal duration. If we did not use temporal clustering, we would have to pad shorter gestures before use with models requiring a uniform input size over all data points, such as PCA or k-means. This would increase the average representation size with no information gain. Supposing that the user has set the desired number of keyframes to f, then the dimensionality of a single gesture after temporal clustering will be reduced to Q x J x f.

## 5.4 Feature Extraction

Certain joints do not contribute as much to the overall representation of a gesture or dance as much as others do–for example, shaking your leg will have a larger effect on a gesture than shaking your foot. The significance of certain joints and their associated angles in different kinds of dance was noticed by Kim et al. [6]. Kim et al. achieved remarkable accuracy in their classification model by extracting the scalar angles created from the positions of important joints and the positions of their neighboring joints, then discarding joints that were deemed insignificant [6].

Our implementation borrows from Kim et al.'s technique and extracts angles in the same way, but because our representation uses a reduced set of frames and therefore has lower dimensionality, we are able to keep more joint angles without reducing performance. The joints that are deemed significant are selected by the programmer before the system begins training (this also allows the joints under consideration to be modified according to a particular dance style or culture). In our current implementation, the joints

[1] In Yang et al.'s original paper on temporal clustering [17], there appears to be an error in the pseudocode in which the diameter calculation is calculated over all n rather than all j, j being the iterator for a for loop. Our implementation uses our modified pseudocode instead of the original.

that we have kept are the middle spine, left shoulder, left elbow, right shoulder, right elbow, left hip, left knee, right hip, right knee. Once the joints have been decided, our pipeline then extracts the angles of important joints, further reducing dimensionality. We use Kim et al.'s technique of computing the angle of rotation between the parent joint and the child joint of any "important joint", thus producing the vectors from the "important joint" to "parent joint" and "important joint" to "child joint" [6] . This process is much like placing any point A in 3D space, placing two other points B and C in arbitrary locations, and measuring the angle BAC created, oriented in the plane created by the vectors AB and AC. This step reduces the dimensionality of a single gesture to j x 1 x f, or j x f, where j is the number of important joints.

## 5.5 PCA

PCA is one of the most widely known approaches to dimensionality reduction available. It is considered a "standard technique for finding the single best (in the sense of least-square error) subspace of a given dimension " [13]. The mathematical principle behind PCA is the creation of a set of principal components that best express or explain the linear variance present in the data. A principal component is found by creating linear combinations of existing axes, with the first principal component exhibiting the most variance among data points, and the second principal component less so, and so on.

In addition to the motivation provided by the experimental success of PCA used in gesture-related domains (see Related Work), we were also inspired to use PCA in our pipeline because its mathematical principle is similar to that of temporal clustering, which also focuses on maximizing variance. The number of principal components is programmer-specified. Suppose that it is set to P, then the dimensionality of a single gesture will be reduced from j x f to simply P. In addition to producing a transformed lower-dimension data set, the PCA model will also be fitted to the data set and will be able to transform novel data points into the same subspace as the data that was used to train it. This step exports the transformed data and the fitted model for future use in the clustering pipeline.

## 5.6 K-means Clustering

K-means belongs to the family of partition based clustering algorithms, whose key principle is the definition and characterization of a cluster by its "center point", where the center point of a cluster is the "average" or the point that minimizes distance between it and all other points in the cluster [16]. K-means updates the centers of clusters iteratively until the clusters eventually converge and each data point is placed into its appropriate cluster [16]. K-means' biggest advantage is that it is relatively computationally efficient, but it suffers from several other issues such as requiring a pre-set number of clusters and being sensitive to outliers [16].

K-means' usage is nonetheless widespread, and the algorithm has been shown to work well in gesture-based domains [1, 10]. Balci et al.'s use of k-means alongside PCA also indicates that the two work well together [1]. Due to k-means' heavy reliance on a distance metric when comparing data-points, we find it intuitive to use for a pipeline that maximizes variance. Given N data items of dimensionality P, the dimensionality of the input is P x N. After k-means is fitted to this data set, it produces a clustering that assigns an index to each data point corresponding to the cluster it belongs to, and a trained k-means model that is able to predict what cluster novel data items belong to, provided that the novel data item goes through the appropriate pre-processing.

# 6 EVALUATION

We conducted a preliminary evaluation of the three-stage pipeline for unsupervised gesture clustering of arbitrary full-body motion data that we developed in order to better understand its ability to cluster skeletally similar gestures and identify limitations. We initially set the number of principal components for the PCA model to two and the number of k-means clusters to three for our evaluation. We initially chose two as the number of dimensions because it allows for easy visualization and inspection of the data, but with a downside of a decrease in accuracy. We later changed the number of k-means clusters from three to four after observing four clear clusters in the visualization. This section details the findings from our evaluation which, while preliminary, offer insights that can inform future research.

## 6.1 Dataset

We gathered a dataset of 104 unique gestures in order to develop an initial understanding of how well our pipeline clustered gestures based on skeletal similarity. Four different members of our lab danced in front of a Microsoft Kinect sensor placed at waist level in order to record the gestures. The participants were prompted to cover a wide variety of motions that each differed greatly from one another. Participants alternated between isolated motions that engaged only one body part and whole body dances or motions that engaged all four limbs. The participants were told not to perform certain gesture types due to the difficulties the Kinect sensor has with tracking them, such as motions that involve rotating the body along the upwards Y axis at rapid speeds (e.g. spinning) or gestures in which body parts were occluded (e.g. laying on the ground).

We attempted to label each gesture in the dataset according to a particular body part category ("hands", "hips", or "legs") in order to determine whether the clusters the pipeline created would match up with the labels we gave them as a way to measure the "intuitiveness" of the clusters generated by our pipeline. Labels were assigned based on which body parts were primarily being used in the gesture–for instance, a one handed wave and two handed wave would both be put under the label of "hands", whereas a gesture depicting a walking motion would be put under the label of "legs". Unfortunately, it proved difficult to intuitively label some of the more complex movements involving multiple body parts, so we ended up only labeling 44 of the 104 gestures (we refer to this as the *reduced dataset* in the remainder of the paper, see Future Work for future plans to improve this evaluation metric).

## 6.2 Efficiency

*6.2.1 Compact Gesture Representation.* Our pipeline is able to approximate gestures at a much lower dimensionality than regular motion capture data, while enhancing the accuracy of clustering. Our pipeline can reduce a gesture, which can consist of over 10,000 dimensions, to a two-dimensional vector. This reduced dimensionality allows us to visualize and inspect clusters more easily and

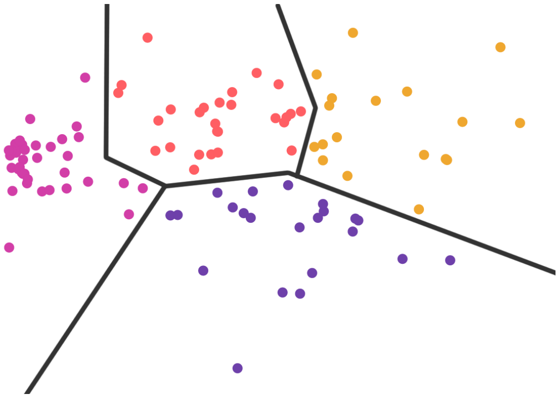

**Figure 3: Visualization of the clustering for the full dataset of 104 gestures. Each color represents a different cluster identified by the k-means clustering algorithm.**

allows us to train models more efficiently. The ability to easily visualize gesture clusters is particularly important in application domains where it is important to evaluate the agent's creative contributions (e.g. computational creativity) and/or help others to learn about the agent's reasoning process (e.g. informal learning spaces). We assessed the efficiency gains that resulted from the more compact gesture representation using using a HP Spectre x360 (2016) running Ubuntu 18.04.

*6.2.2 Real-Time Pre-Processing.* One of the driving motivations behind our pipeline implementation was the need to rapidly add new data to clusters in order to enable the *LuminAI* agent to quickly respond to performed gestures with relevant "similar" gestures. This means that the pipeline needs to pre-process gestures quickly. We found that it took a total of 157.6 seconds to pre-process the reduced dataset of 44 gestures and to train the PCA model. This averages to 3.56 seconds to process one gesture, which is 1-2 seconds longer than desired but feasible for a real-time application (especially if run on a more powerful computer and if the code is further optimized for performance).

*6.2.3 Clustering Speed.* The gesture dataset will need to be periodically re-clustered as the agent learns new data. Because gestures are pre-processed in real-time as data is gathered, clustering speed is improved using our pipeline. The runtime to cluster the full dataset of 104 gestures with no pre-processing was 1.789822 seconds, whereas the runtime for the data that was pre-processed using our pipeline was 0.038425 seconds. This speedup would be amplified for larger datasets.

## 6.3 Noise Sensitivity

We produced a duplicated gesture for each recorded gesture with a random amount of Gaussian noise added to one of the body parts in order to increase the number of total gestures and to test our pipeline's resilience to random noise. The amount of noise added to each body part was calculated by sampling from a Gaussian distribution with a mean of 0.07 and sigma of 0.02 three times. The three independent samples are concatenated into a 3D vector and added to the position of a randomly selected joint, excluding spine and hip joints due to the physical limitations of the human body, for every frame in a gesture. The mean magnitude of the noise added is 0.1212 and about half the size of the skeleton's forearm in our representation. Examination of the cluster visualization of the reduced dataset with randomized gestures added indicated that our clustering approach is fairly resistant to the effects of random noise, as almost all of the randomized gestures are placed into the same cluster as their parent.

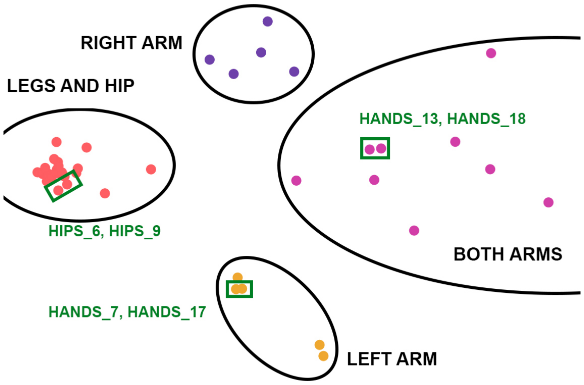

**Figure 4: Visualization of the clustering for the reduced, labeled dataset of 44 gestures. Each color represents a different cluster identified by the k-means clustering algorithm.**

| Cluster | # gestures | Legs % | Hips % | Hands % |
|---|---|---|---|---|
| Both Hands | 9 | 0 | 11.11 | 88.88 |
| Right Hand | 5 | 0 | 0 | 100 |
| Left Hand | 5 | 0 | 0 | 100 |
| Lower Body | 25 | 68 | 32 | 0 |

**Table 1: Cluster composition for the reduced dataset of 44 gestures**

## 6.4 Cluster Clarity

We visually evaluated our pipeline's ability to cluster items using the reduced dataset. Our hypothesis was that the clustering visualization would produce clearly identifiable clusters of the gestures that correspond to the "hands", "hips", and "legs" labels applied to the reduced dataset. As Fig. 4 shows, the red clustering on the left (labeled *Legs and Hip*) is the most obvious due to its density. Three more distinct clusters can be observed towards the top, bottom and right hand side of the visualization. The clusters in the full dataset (Fig. 3) are less visually apparent, but this is to be expected as participants were instructed to perform varied gestures, meaning that not many gestures in the full dataset were similar to one another. As a point of comparison, all of the gestures except for one outlier appear to be clustered in a small and very dense clump in the center of the visualization generated from the results from running only PCA and k-means on the data without our pipeline pre-processing (not pictured due to space constraints), indicating that our pipeline did a better job of separating the gestures into distinct clusters.

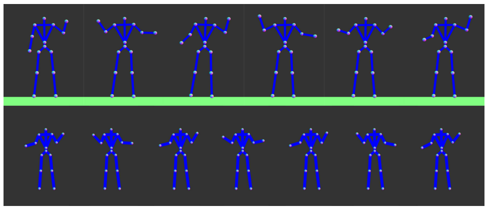

**Figure 5: A success case: Double arm oscillation (Hands-13, top) and waving (Hands-18, bottom) gestures are considered similar. This example also highlights some understanding of "rhythm".**

## 6.5 Cluster Accuracy

We initially hypothesized that we would see three distinct clusters of gestures in the reduced labeled dataset after pre-processing - one for "hips", one for "legs", and one for "hands". We actually found four distinct clusters (see Fig. 4). We evaluated the quality of these clusters based on how how homogeneous each cluster was in terms of the labeled gestures it contained (i.e. a cluster consisting exclusively of "hands" gestures was considered a better clustering than a cluster consisting of an equal mix of "hands", "hips" and "legs"). We found that the clusters on the top (blue) and bottom (yellow) of the visualization consisted exclusively of "hands" gestures (see Fig. 4, Table 1). The two clusters correspond to left and right arm motion, suggesting that our pipeline discretized the two body parts into their own individual clusters. The cluster on the far right (pink) also consists of 88% gestures labeled as "hands". This cluster appears to be formed from "hands" gestures that involve both left and right arm motion, explaining its positioning between the left and right arm clusters. The final cluster is shown on the far left (red) in Fig. 4 and is the most mixed of all the clusters present, composed of all of the "legs" and "hips" motions together. The likely reason behind this is that at the time of recording, we did not realize that moving ones' hips almost certainly involves the reorientation of the legs. In addition, in all the "hips" and "legs" motions, the performers' arms were static by their sides, causing the upper bodies in these gestures to be identical to one another, likely explaining the density of this cluster. In spite of the unexpected results, we found that according to our evaluation metric, the clusters created for the reduced dataset were agreeable.

## 6.6 Visual Inspection

We conducted a qualitative visual inspection of both the reduced and the full dataset to supplement our quantitative evaluation of the reduced dataset clusters. We present several exemplar gesture comparisons from the reduced dataset here to highlight areas where the pipeline succeeded and failed. In Figures 5 - 7, gestures are depicted as a series of keyframes and should be read left to right. Gestures are presented in pairs for comparison, with one gesture on the top row and one gesture on the bottom row. Fig. 4 shows the location of the exemplars we selected within the clustering plot for the reduced dataset (the exemplars are boxed and labeled in green).

The two gestures pictured in Fig. 5 were placed in the "both hands" cluster. In the first gesture, the skeleton moves both of its hands in a circular fashion, engaging its elbows in the motion. This is visually similar to the second gesture, in which the skeleton performs a simple wave with both hands. The keyframes shown also suggest that the system has some understanding of "rhythm" as the reduced keyframe set clearly depicts the "left arm then right arm" rhythm of the gestures.

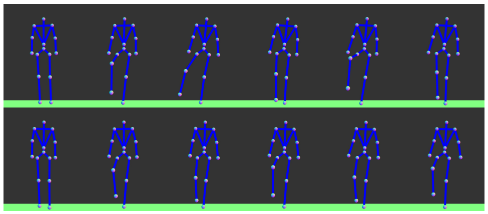

**Figure 6: A success case: two leg motions (Legs-16, top, and Legs-8, bottom) involving the raising and lowering of the knee are considered similar despite the addition of lateral motion in Legs-16 (top).**

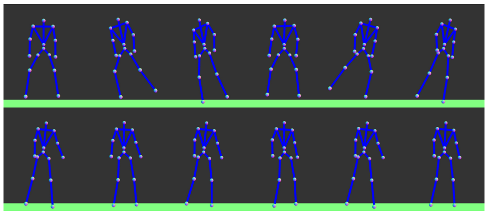

**Figure 7: A failure case. Upper body swaying and lateral translation (Hips-7, top) is considered similar to a slight swinging of the hips (Hips-9, bottom).**

The gestures pictured in Fig. 6 both depict the raising and lowering of the knee–however, the gesture shown at the top adds more lateral motion to the knee joint. In spite of the difference between the two, we found their close clustering agreeable due to their intuitive visual similarity.

The two sets of gesture keyframes pictured in Fig. 7 are an example of what we consider a failure case. The emphasis in the gesture shown at the top is clearly the lateral swaying and leaning of the upper body whereas the gesture on the bottom depicts only the lateral swinging of hips. This difference is lost because our angle extraction works only on specific body parts and does not take into account the rotation of the whole body. As a result, these two gestures are considered similar due to their close joint orientations. This effect was observed in several other gesture pairs.

We also noticed that the clustering visualization of the reduced dataset took on an unexpected emergent property–the placement of data points in this space allows one to immediately determine which body part was most active simply by looking at which quartile it lies in. This is an intriguing property because it suggests that the system, with no input from the user, has identified the body parts of the human skeleton that exhibit the most motion variance. It has learned on its own that limbs are an important part of motion and clustered data points using them.

## 6.7 Inherent Limitations of Pipeline

There are several limitations of the pipeline inherent to its implementation that need to be taken into consideration and may make it more or less suitable for certain styles of movement improvisation.

*6.7.1 Angle Invariance.* We extract angles from important joints using their Cartesian coordinates as part of the pre-processing step (see Feature Extraction). This step introduces an invariance to the actual position of the user relative to the Kinect camera, as applying a transformation to all joints will have no effect on the angle extracted. A person performing a wave to the left of the camera will have the same joint angles as a person on the right. Angle extraction also makes the system blind to the direction a rotating joint is currently facing. These consequences of angle extraction could interfere with dance styles or gestures that emphasize translational movement or the direction of angular movement.

*6.7.2 Inability to abstract motions from specific body parts.* Each of the body parts is given a unique position in the feature vector used to describe a Kinect skeleton. This means that the system has difficulty equating similar motions mirrored across the Y axis of the human body. For human users, it is apparent that a hand waving motion is a "wave" regardless of which arm it is performed with. However, our system does not view these two to be similar as it has no preconception of the symmetrical human body, nor the relationship between the left and right arms.

*6.7.3 Change Emphasis.* Our pipeline gives equal weight to body parts that remain static and body parts that are moving from frame-to-frame, but we noticed during our evaluation that we intuitively placed a greater weight on moving body parts when comparing similar gestures. We hypothesize this to be the cause of some of the failure cases observed in the reduced dataset "legs and hips" cluster. Due to the similar positions of the upper bodies in the gestures from that cluster, gestures that are sometimes visually dissimilar to humans due to movement of an angle, like the hip, are placed together due to their upper body similarities.

## 7 FUTURE WORK

We plan investigate how to mitigate some of the limitations of the pipeline highlighted in the Evaluation section. This will include technical pipeline efficiency and accuracy improvements as well as collecting a larger dataset and exploring more rigorous methods of assessing clustering quality. Further work is necessary to fully understand the ability of the pipeline to find meaningful clusters for larger datasets that contain varied gestures that involve the motion of many body parts at one time. This was challenging to assess using our preliminary approach to evaluation, both because we could not visualize clustering visualizations with more than two dimensions, and because it was difficult for us to come up with meaningful labels for full-body gestures. In the future, we might explore how the algorithm performs in relationship to labeled datasets generated by expert dancers/choreographers who are able to more accurately label complex movements and/or investigate whether or not users of the system can discern a difference in gesture responses generated using our pipeline vs. random responses. In addition, the gesture clustering pipeline we have built can support reasoning/clustering along non-skeletal metrics of similarity, such as Laban movement analysis [7]. We plan to further explore how different ways of reasoning about movement can affect co-creative movement improvisation within the context of *LuminAI*.

## 8 CONCLUSION

In this paper, we have combined multiple strategies used for gesture dimensionality reduction [1, 6, 17] with a k-means clustering technique to develop a pipeline for unsupervised clustering of arbitrary full-body motion data. We conducted a preliminary evaluation of our pipeline and found that it is able to efficiently and intuitively cluster gestures involving the movement of isolated body parts. The pipeline is resilient to noisy data and produces clear clusters in response to gestures that are intuitively similar in terms of skeletal positioning. Our main contribution is the novel combination of existing strategies for clustering and dimensionality reduction into a pipeline that can be used in a variety of movement improvisation domains.

## REFERENCES

[1] 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.

[2] 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.

[3] 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

[4] Sarah Fdili Alaoui, Jules Françoise, Thecla Schiphorst, Karen Studd, and Frédéric Bevilacqua. 2017. Seeing, Sensing and Recognizing Laban Movement Qualities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 4009–4020.

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

[6] 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.

[7] Rudolf Laban and Lisa Ullmann. 1971. *Mastery of Movement* (3rd ed.). Macdonald & Evans Ltd, London, United Kingdom.

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

[9] Mikhail Jacob, Gaëtan Coisne, Akshay Gupta, Ivan Sysoev, Gaurav Verma, and Brian Magerko. 2013. Viewpoints AI. http://www.aaai.org/ocs/index.php/AIIDE/AIIDE13/paper/view/7398

[10] 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.

[11] R Keith Sawyer. 2006. Group creativity: Musical performance and collaboration. *Psychology of Music* 34, 2 (2006), 148–165.

[12] Daniel L Silver, Qiang Yang, and Lianghao Li. 2013. Lifelong Machine Learning Systems: Beyond Learning Algorithms.. In *AAAI Spring Symposium: Lifelong Machine Learning*, Vol. 13. 05.

[13] Carlos Oscar Sánchez Sorzano, Javier Vargas, and A Pascual Montano. 2014. A survey of dimensionality reduction techniques. *arXiv preprint arXiv:1403.2877* (2014).

[14] 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).

[15] Tian-Shu Wang, Heung-Yeung Shum, Ying-Qing Xu, and Nan-Ning Zheng. 2001. Unsupervised analysis of human gestures. In *Pacific-Rim Conference on Multimedia*. Springer, 174–181.

[16] Dongkuan Xu and Yingjie Tian. 2015. A comprehensive survey of clustering algorithms. *Annals of Data Science* 2, 2 (2015), 165–193.

[17] Yang Yang, Hubert PH 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*. ACM, 479–482.