

Low-Dimensional Embeddings for Interaction Design

Marius Mihai Rusu,* Svenja Yvonne Schött,* John H. Williamson, Albrecht Schmidt, and Roderick Murray-Smith

Physical interactions with the real world have many degrees of freedom, which has led to the development of novel input devices with a multitude of sensors to capture increasingly high-dimensional data. This high dimensionality makes the design of interactive systems more complex. Herein, the use of autoencoder-based dimensionality reduction is explored to simplify the design process. For this purpose, a data glove equipped with accelerometers is used to record high-dimensional hand movement data that are thereafter reduced to 2D embeddings using autoencoders. The exploration and evaluation of the resulting embeddings suggest that autoencoders can be used to create meaningful low-dimensional representations of complex human movement. The characteristics *generality*, *variability*, *connectivity*, and *distinguishability* are established and a guideline is provided for assessing low-dimensional embeddings. Referring to these characteristics, system engineers can evaluate different input modalities and gestures for their specific interaction task. Further, a framework is outlined for designing and evaluating gesture interaction in the low-dimensional space. By demonstrating the exemplary design of the interaction with a virtual lever, this research gives system engineers a template for interaction design in the low-dimensional space.

everyday life, they can also be quite complex.^[4] Designing for more interactive objects, with hand gestures acting as one of the main controls, is an area of increasing research and commercial interest. While the degrees of freedom (DOF) the human hand possesses allow for “flexibility to perform skilled finger movements”^[5] in real life, they pose technological challenges for digital interfaces, as most typical interaction design processes are not yet well adapted to high-dimensional input.^[6]

The goal when prototyping such controls is to find gestures that are easy for the user to recall and perform reliably, and easy for the computer to distinguish from one another. However, gestures possess a high degree of complexity and human movement possesses a variability^[7] that needs to be known and incorporated into the design process. Confronting designers with the raw high-dimensional data as input and asking them to design robust, usable interactive systems has proven to be challenging.

This has limited the impact of novel sensing systems that capturing more of the complexity of real-life interactions. Such systems’ datasets possess higher dimensionalities, which makes them more challenging for designers to understand and utilize. Lu et al., for example, are developing a system that can capture the full 25 dimensions hand movement possesses with the help of multiple inertial sensors.^[8] Further, large amounts of high-dimensional data can be generated from the deployment of multiple sensors across physical objects, as is the case for the WebBike Project, where the researchers equipped e-bikes with sensors.^[9]

The highly complex datasets created by those novel sensing systems require a layer of simplification to become comprehensible for researchers and designers and to proactively apply design processes appropriately. We explore the role of low-dimensional embeddings in conceptualizing and visualizing high-dimensional movement data by answering the following research question.

Can Autoencoder-Based Dimensionality Reduction Simplify the Data-Driven Design Process?

Our findings suggest that the low-dimensional representation of complex movement data is more convenient to work with than high-dimensional data, which builds a basis for analyzing and visualizing hand movements and interactions among team members during the design process. Further, we define

1. Introduction

Daily life is shaped by interactions where we use our hands.^[1–3] From the coffee mug we put to our mouths, to our fingers finding their way to the correct letters on our keyboard, hand movements are ever-present. Using data gloves, optical tracking, or accelerometers, a lot of information about these habitual interactions can be captured, showing that, as well as dominating

M. M. Rusu, S. Y. Schött, Prof. A. Schmidt
Lehrstuhl Medieninformatik
LMU Munich
Frauenlobstr. 7a, München 80337, Germany
E-mail: M.Rusu@campus.lmu.de; S.Schoett@campus.lmu.de

Dr. J. H. Williamson, Prof. R. Murray-Smith
School of Computing Science
University of Glasgow
18 Lilybank Gardens, Glasgow G12 8RZ, Scotland

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202100045>.

© 2021 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202100045

characteristics that low-dimensional embeddings should display to support interaction design. By means of exemplary studies, we demonstrate how the low-dimensional space can be used to analyze high-dimensional hand movements, predict prospective design performance, and analyze the resulting systems. This approach enables the design team and machine learning team to communicate constructively, thus leading to better collaboration and interaction designs.

2. Background

2.1. Interacting with Hands

Hands are a powerful tool for interacting with our surroundings both in real life and virtually. If we think of virtual interactions as finite state machines coupled into direct control tasks, we can use our hands to trigger transitions between states. Through continuous changes in the way we move our hands, we can guide changes in our surroundings. Think about how you operate a mouse. By changing the posture of your hand and moving your index finger down you click and trigger an interaction; changing the speed of your hand in combination with this posture enables you to drag something—a continuous interaction.

A posture is directly related to the DOF of the hand, as it is specified through the degree by which joints of fingers are bent.^[10] Postures do not contain information on movement as they are “static hand gestures.”^[11] Gestures are defined as movement^[12] of different body parts,^[3] most notably hands, to convey information.^[3,11] This movement suffices to define a gesture according to Xu et al.^[12] In contrast, the posture of the hand is considered a defining characteristic of gestures by others.^[3,11,13]

One approach to the evaluation of the quality of gestures for interactions is via gesture elicitation studies.^[14] These are user-centered studies with the goal of understanding the users’ mental models of gesture interaction.^[14–16] Participants typically pick^[17] or design^[15,18] gestures for interactions in a think-aloud experiment setting,^[17] sometimes enhanced through additional assessment of the quality of the gesture for the interaction in the form of Likert scales.^[17] Thereafter, a consensus agreement is calculated using the agreement rate formula by Wobbrock et al.^[18] Alternatively, we built the theoretical foundation for taking a data-driven approach of analyzing the quality of gestures based on their inherent features, which we record using a data glove and accelerometers and process using autoencoder-based dimensionality reduction. We do not aim at presenting our research in contrast to other state-of-the-art approaches but rather aim at presenting our data-driven approach as a general tool to simplify the design process.

2.2. Data Gloves and Accelerometers

Data gloves are input devices in the shape of a glove to sense hand and finger positioning and movement.^[1] They are typically used for data acquisition in areas ranging from information visualization to medicine.^[1,19] In recent developments, research has focused on increasing robustness,^[2] scalability,^[20] sensor

density,^[20] utilizing less intrusive materials,^[21] and becoming more lightweight and inexpensive.^[6,20] An example for this is the Dexmo glove that we are using for our research. The Dexmo is a lightweight force-feedback glove, capturing 11 DOF.^[22] The glove can capture data on changes in the bending state of the five different fingers, as well as finger split angles and thumb rotation. We enhance the data collected via the Dexmo data glove with additional sensors. The information is captured via lightweight inertial measuring units (IMUs) attached to the glove as displayed in **Figure 1**. Each sensor has three-axis accelerometers, gyroscopes, and magnetometers, which can be used to estimate body poses^[23] and hand gestures.^[12] Instead of utilizing existing peer-reviewed datasets such as Zhang and Harrison,^[24] for example, we decided to create our own dataset using the enhanced Dexmo glove to have control over the input modality, sensors, and dimensionalities. Further, our goal was not to produce generalizable results, but to analyze and demonstrate how to prototype high-dimensional interactions in the low-dimensional space.

2.3. Dimensionality Reduction with Autoencoders

Dimensionality reduction techniques are transformations that create a low-dimensional representation of high-dimensional data,^[25] which are used for further processing, visualizing, and analysis.^[26,27] The quality of the low-dimensional embedding is traditionally defined through how much information and structure within the data is preserved,^[25–29] optimizing on the loss between original and restored data.^[30] The suitability of neural networks for representation of the complex, high-dimensional mappings required to process the data from data gloves makes them an obvious choice to explore. As early as 1993, Fels demonstrated with his GloveTalk system that neural networks can be used for processing movements and enabling complex control of

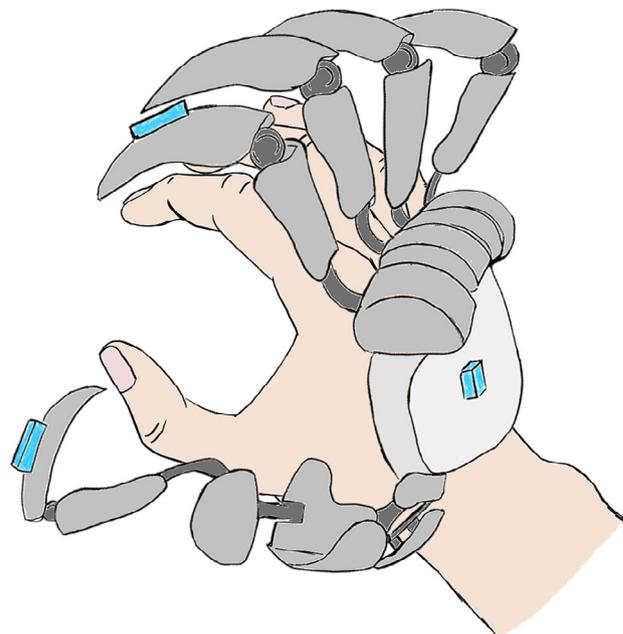


Figure 1. Outline of the Dexmo glove and the attached IMUs (blue).

behavior in interfaces.^[31,32] Machine learning approaches are applied not only to understand human movement,^[33] but also to personalize interaction and thus accommodate for differences in people.^[34]

There is a wide range of options for acquiring a low-dimensional embedding that is useful for interaction design, from traditional linear mappings such as principal component analysis (PCA) and factor analysis (FA)^[25] to modern nonlinear methods such as t-distributed stochastic neighbor embedding (t-SNE), uniform manifold approximation and projection (UMAP), and autoencoders. Whereas supervised learning methods learn from predefined input and output,^[31] autoencoders are unsupervised in that they do not require manually selected examples of the desired output. Autoencoder models are multilayered neural networks with a low-dimensional (compared to the input dimension) *bottleneck layer* in the middle. The activations of this bottleneck layer can be seen as a low-dimensional *encoding*, created from the input that can be reconstructed by the later *decoder* layers. Although the standard gradient-descent training process of autoencoders can be computationally intensive, in comparison with other dimensionality reduction techniques, they offer the opportunity of generalizing on the basis of new data, and importantly for analysis of human movement, autoencoders perform well on natural, real-world data.^[29]

3. Turning High-Dimensional into Low-Dimensional Data

To answer our research question, we first need to create exemplary low-dimensional embeddings of complex movement data. In this section we give an overview of the recorded data and their characteristics as well as the training and optimization process of the autoencoders. We will not go into the technical details, as the Supporting Information is intended to give interested readers a deeper understanding of the implementation and allow replicability.

3.1. Recording High-Dimensional Data

To gather the required data, we performed a “Rewarding the Original”-like^[35] study with four participants, three of whom are co-authors (3 male, hand length = 18.7 cm, 19 cm, 20.5 cm, aged 22, 38, and 52; 1 female, hand length = 17 cm, aged 22). The task was to move and gesticulate as freely and as diversely as possible for a duration of roughly 3 min. The goal of this was to attempt to cover the entire movement space that contains all possible hand postures by that user, as well as the transitions between those postures, and thus obtain a dataset with high generality. Using this procedure, we gathered 20 000 samples of high-dimensional data in an unsupervised fashion for training and optimizing the autoencoder. Large datasets as such are only required once to train the autoencoder on human movement and therefore are not part of the design process.

Further, to evaluate the low-dimensional embedding and analyze gestures within that space, we gathered data on everyday gestures. Recording, visualizing, and analyzing gestures is a fundamental part of the design process as it allows designers

to learn the low-dimensional representation of gestures and determine the best fit for their specific interaction task. We selected six gestures that are frequently used during the daily routine and instructed the participants to perform each of these gestures five times. Half of the gestures were general gestures, while the other half relied on the accelerometer to grasp the movement properly. Using this procedure, we gathered 20 time-series for each of the predefined gestures to visualize and evaluate within low-dimensional embedding. The following list and visualization in the Supporting Information give an overview of the recorded gestures: 1) general gestures: 1.1) pinching with index finger and thumb, 1.2) pinching with index finger, middle finger, and thumb, 1.3) squeezing with the entire hand; 2) accelerometer-specific gestures: 2.1) pointing with the index finger, 2.2) rotating a knob clockwise with the entire hand, 2.3) swiping from left to right with the entire hand.

The data in their original state comprise the following features: 1) 5 dimensions—bending state of each finger ranging from 0 (fully stretched) to 1 (fully bent); 2) 5 dimensions—bending velocity of each finger derived from the bending state; 3) 18 dimensions—acceleration and gyroscope values from three accelerometer devices.

3.2. Optimizing and Training the Autoencoder

Using the collected data, we optimized and trained three separate autoencoders that operate on different subsets of features using the PyTorch framework^[36,37] and the GPyOpt library.^[38] Instead of creating just one autoencoder, we want to investigate how different input modalities shape the low-dimensional embedding.

The autoencoders will hereinafter be referred to as *model A*, *B*, and *C*. Model A operates only on the bending state and thus reduces 5 to 2 dimensions. Model B operates on both the bending state and the bending velocity and therefore maps 10 dimensions to 2. Model C operates on all features, thus reduces dimensionality from 28 dimensions to 2 dimensions.

By using different sets of sensors for gathering high-dimensional data and reducing the data to a low-dimensional space, designers can assess these sensors and identify the most suitable sensors for their desired interaction. This provides a basis for a data-driven judgement on finding the balance between using enough sensors and adding too much noise through too many input modalities.

Although a higher dimensionality of the reduced data potentially preserves more information about the hand movement and can be appropriate for further classification work, in this article we focus on 2D embeddings as they can be visualized comprehensible. The resulting visualizations of complex movement data support interaction design and allow constructive communication between the design team and machine learning team.

4. Designing Embeddings for Interaction

In this section, we present and visually interpret the low-dimensional embeddings generated by the autoencoder-based dimensionality reduction. In addition, we extract the main characteristics visible in the embeddings and discuss how they affect the design of gesture interaction.

Applying the autoencoder-based dimensionality reduction to the high-dimensional data results in three 2D embeddings visualized as scatterplots in **Figure 2**. As in Section 3.2, we refer to the various embeddings as *Embedding A*, *B*, and *C*, on the basis of the underlying model. Depending on which model we examine, each data point in the scatter cloud has a different meaning because the models operate on different sets of features. While *Embedding A* visualizes simple hand postures, *Embeddings B* and *C* are more complex. Each point created by model *B* illustrates a certain position and velocity of the fingers. Model *C* extends model *B* by additionally comprising the orientation and acceleration of the entire hand and therefore holds the highest density of information. As mentioned, these data were collected from four participants. During the interpretation of the embeddings, we refer to them as users to underline the context of interaction design. To gain further understanding of the low-dimensional representations, we visualize the recorded gestures mentioned in Section 3.1 as trajectories in **Figure 2**. One trajectory is a 2D representation of the time-series during which the respective gesture was performed once. Multiple repetitions of the same gesture thus lead to overlying trajectories of the same color. Exploring those gestures in a low-dimensional context reduces the complexity and offers a better understanding of hand movement compared to the high-dimensional data.

4.1. Generality

We observe that the data points in *Embedding A* and *B* are well distributed across the entire movement space, although showing a higher density in frequently revisited portions of that space.

These evenly distributed embeddings without outliers suggest that a small number of users may suffice for creating general embeddings covering a wide range of hand postures and gestures. *Embedding C* in contrast shows a higher density in central areas, whereas peripheral areas contain a large number of outliers, especially in the lower right corner. This indicates that gathering larger amounts of data per user is necessary to cover the entire individual movement space. The reason for this is the high dimensionality of *Embedding C* given by the additional sensors comprising the orientation and the higher-order derivatives associated with the rotational velocities and acceleration of the fingers. This results in a much larger, more complex movement space, dependent on dynamic variation of hand poses.

From a designer's perspective, having an insight into the generality of the dataset allows for better judgement regarding the suitability of the sensors and the collected data for a specific interaction task. Designers can compare different input modalities and assess the amount of required data to cover the possible movement space for their desired application. Using dimensionality reduction to create low-dimensional representations of complex data supports this task.

4.2. Connectivity

To ease the understanding of the low-dimensional space, **Figure 3** shows a rasterized subset of *Embedding A* as glyphs, where the bending state of the fingers is encoded in the color. This color coding highlights a transition from a fist on the right to a stretched-out hand on the left. This transition indicates that neighboring points in the high-dimensional context are mapped

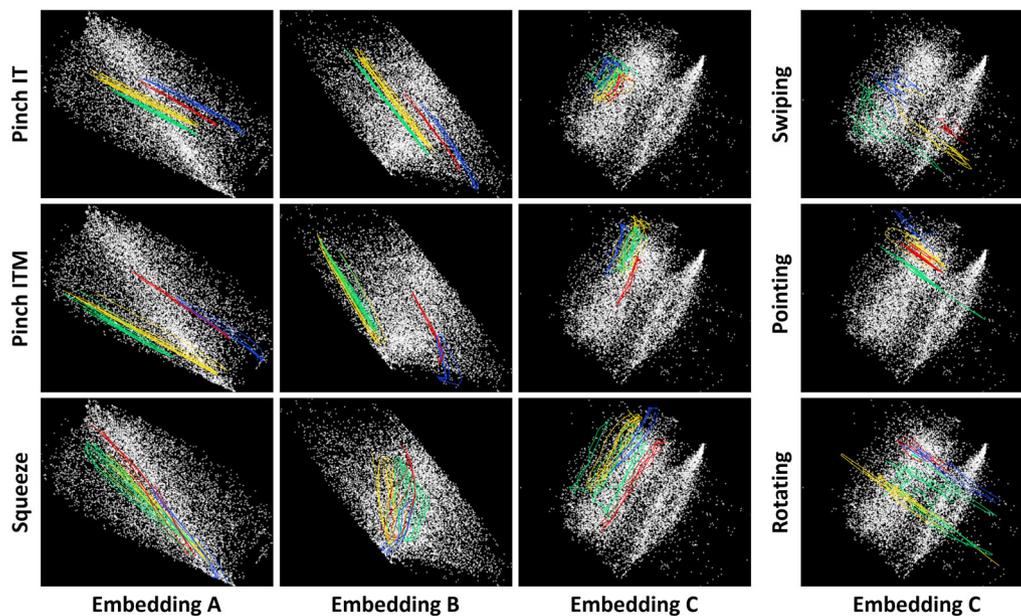


Figure 2. The trajectories of the recorded gestures in the scatterplots of the 2D embeddings. The *x*- and *y*-axes of the scatterplots represent the two dimensions of the embedding space. The range of the axes is not relevant as we are only interested in the overall topology of the embedding space. The four users are distinguished by color (green, blue, red, and yellow). *Embedding A*, *B*, and *C* (columns) show the general gestures (rows) “pinching with index finger and thumb,” “pinching with index finger, middle finger, and thumb,” and “squeezing” with the entire hand in the respective 2D space. In addition, *Embedding C* shows the accelerometer specific gestures “swiping” from left to right with the entire hand, “pointing” with the index finger, and “rotating” a knob clockwise with the entire hand.

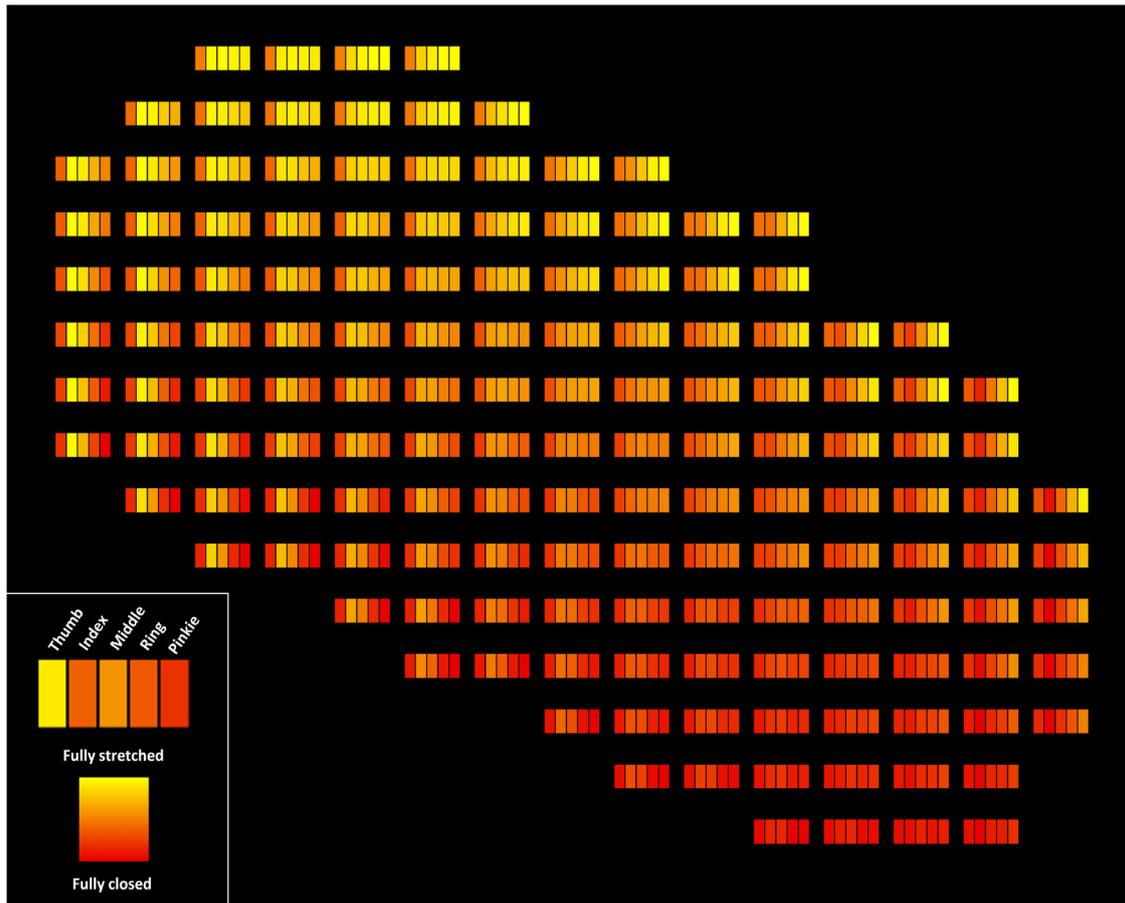


Figure 3. Rasterized glyph visualization of Embedding A. As shown in the legend, the fingers are displayed from thumb (left) to pinkie (right) as squares, while the bending state is encoded in color. The colors range from fully stretched (yellow) to fully closed (red).

to neighboring areas in the low-dimensional context with neighboring points referring to points in spatial and temporal proximity to each other. This visualization is helpful to understand the effects of the dimensionality reduction on the data but reaches its limits for more complex embeddings that go beyond simple postures such as Embedding B and C. Assessing the connectivity of the low-dimensional representation early in the design process helps designers to better understand the nature of their dataset.

Designers can gain insight into which features of the hand movement are decisive and shape the movement space, and thus adjust their gesture interaction accordingly. Without low-dimensional representations of the high-dimensional movement data, assessing such characteristics is difficult and often infeasible for humans.

4.3. Within-User Variability

Looking at the trajectories of multiple repetitions of one gesture and from one user at a time, we notice that Embedding A and B show a lower variability than Embedding C as the repetitions manifest in close proximity. As Embedding C is based on both the glove data and the accelerometer data, it can grasp more characteristics of the movement. This higher number of degrees of

freedom causes multiple repetitions of the same gesture to differ more distinctly from one another compared to lower degrees of freedom in Embedding A and B. We observe for example that the trajectories visualizing the gesture “pinching with index finger and thumb” are irregular in Embedding C, even though they are dense and regular in Embedding A and B. Such a high level of within-user variability can especially be observed in the gesture “rotating” in Embedding C, where the trajectories of the user represented by the color green stretch across the entire center of the embedding space.

4.4. Between-User Variability

Further, comparing the trajectories between users gives us an insight into individual behavior. Due to the generally high variability in Embedding C, the trajectories derived from different users become hard to evaluate. Nevertheless, we can observe that the trajectories of the gesture “swiping” differ significantly between participants, especially the trajectories of the users represented by the colors blue and red. By looking at Embedding A and B, we also note that some gestures possess a higher variability than others. The trajectories showing the gesture “pinching with the index finger and thumb” are similar across users, unlike the

trajectories of the gesture “pinching with the index and middle finger and thumb”. Even though the general interpretation of a pinching motion is clear, the exact execution may differ due to small variations in the bending state of the fingers and the dynamics of the movement execution. The users represented by the colors red and blue might have a different understanding of the gesture “pinching with the index and middle finger and thumb” compared to the users represented by green and yellow.

A central stage in the design process of gesture interaction is the acquisition and analysis of different gestures. One interesting question is how a certain gesture is performed repeatedly by a user and how different users perform this gesture. Visualizing and evaluating the gesture in a low-dimensional context provides useful information by pointing out differences and similarities regarding both between-user and within-user performance.

4.5. Distinguishability

We observe that Embedding A and B separate the gestures in a more distinct fashion than Embedding C. This means that the time-series of the different gestures have a low resemblance and do not contain a large number of similar data points. This difference in distinguishability can especially be observed with the trajectories of the gesture “pinching with index finger and thumb” and “pinching with index, middle finger and thumb”. In Embedding C, those two distinct gestures share a large number of similar data points in the upper portion of the embedding space, whereas they are clearly separated in Embedding A.

Being able to distinguish gestures is vital for designing interactions, where different gestures control different interactive devices. Having knowledge of how distinct or similar certain gestures are helps assign gestures to control tasks and reduces errors and confusion. Similar to variability, low-dimensional embeddings reveal such characteristics of gestures and present them in a more comprehensible way compared to the high-dimensional context.

Visualizing and interpreting the low-dimensional representation of complex hand movement reveals various characteristics of the data. With our exemplary dataset collected from four users and reduced to two dimensions using autoencoders, we establish the characteristics *generality*, *connectivity*, *within-user* and *between-user variability*, and *distinguishability*. These characteristics can be visually evaluated in a low-dimensional context and provide guidance during the design process of gesture interaction.

5. Designing Simple Interaction

In the previous section, we show how autoencoder-based dimensionality reduction can provide theoretical guidance during the design process. In this section, we want to demonstrate how designers can use the low-dimensional representation to prototype interaction tasks directly within the embedding space. The main advantage of designing in a low-dimensional context is the reduced complexity of the data and providing designers with a more easily visualized representation of the gestures and control tasks.

To do so, we put ourselves in the position of a system engineer trying to design an exemplary gesture interaction task. From the wide range of possible everyday tasks we chose that of “interaction with a lever.” The goal of this exemplary design process is to model the interaction with a virtual lever using only low-dimensional embeddings and simple state classification methods. We want to demonstrate how designers can leverage quick results when prototyping gesture interaction while also visualizing the design progress and keeping it transparent and comprehensible. We divide the design process into the following four steps.

5.1. Find a Suitable Embedding

Since manipulating a lever is a straightforward motion, a small number of sensors are sufficient to capture this motion. As stated in Section 4, reducing the number of sensors also reduces the variability of gestures in the low-dimensional embedding and therefore increases the robustness of gesture and posture recognition. Therefore, we decide to use Embedding B, which takes the bending state and velocity of the fingers into account.

5.2. Record a Distinct Posture for Selecting and Deselecting the Lever

Ideally, we do not want to be limited to one single control task but to be able to choose from a variety of different objectives. We therefore need to be able to manually select and deselect the interaction with the lever through one distinct posture. For this, we choose the posture “pinching with thumb and ring finger” as it is easy to put into practice. We instructed one user (demographics mentioned in Section 3.1) to perform the posture “pinching with thumb and ring finger” as diversely as possible to encode a high variability into the training data. By doing so, we collected five samples of the posture, which are visualized as scattered blue points in Figure 4.

5.3. Record a Continuous Gesture to Manipulate the Lever

For manipulating the lever, we need a continuous gesture that translates well between the different states of the object. As there is a multitude of potentially suitable gestures for that task, we choose not one but three different gestures and compare them in Section 6 with regard to performance. We recorded five repetitions from one of the same users as earlier, where they performed the following gesture candidates: 1) pinching with index finger and thumb; 2) pinching with index finger, middle finger, and thumb; and 3) squeezing with the entire hand.

Figure 4 visualizes the repetitions of these gestures as single trajectories in the low-dimensional embedding. We observe that both variations of pinching are straight lines, aligned in a precise manner, while “squeezing” shows a higher variability between individual repetitions.

5.4. Map the Posture and Gesture to a Virtual Lever

Once we record and visualize the posture for selection and deselection and a variation of multiple gestures for manipulating the

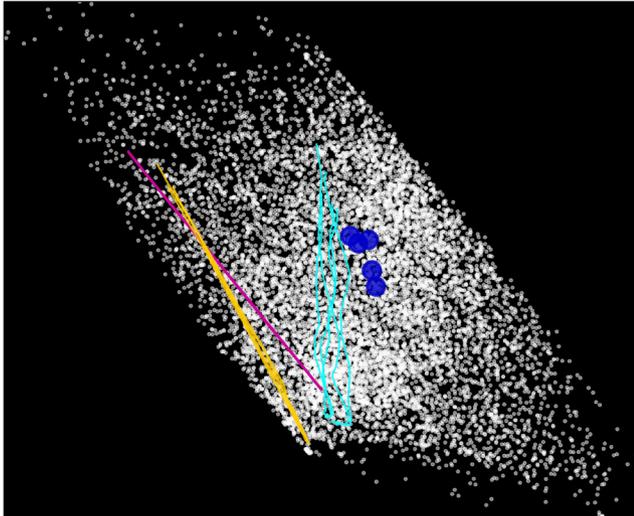


Figure 4. Potentially suitable gestures and postures for the interaction with a virtual lever in Embedding B. “Pinching with thumb and ring finger” (blue) is used for selecting and deselecting the lever. The gestures “pinching with index finger and thumb” (magenta), “pinching with index finger, middle finger, and thumb,” (yellow) and “squeezing” with the entire hand (cyan) are used for manipulating the lever.

lever, we need to perform posture and gesture recognition to map the user’s movement to the virtual lever.

5.4.1. Posture Recognition for Selecting and Deselecting the Lever

For this subtask, we compute the Euclidean distance between the currently performed posture of the user and the five repetitions of the recorded posture. Once the Euclidean distance is below a predefined threshold, the posture is recognized and the interaction task is either selected or deselected. This threshold can be seen as a level of tolerance and can be modified depending on the embedding and requirements of the task.

5.4.2. Gesture Recognition for Manipulating the Lever

For this more complex subtask, we compute the mean trajectory for each of the three gestures. The mean trajectory allows us to span a vector from the starting point to the end point of the motion and we map this transition to the maximum and minimum state of the virtual lever. By partitioning this vector into an arbitrary number of small vectors, we calculate equally distributed keyframes on the trajectory. These keyframes represent the different states of the gesture and therefore the different positions the lever can reach. In our setup, we partition each gesture into 31 different states but higher or lower values are possible. Similar to subtask 4.1, we compute the Euclidean distance between the currently performed posture and the 31 different states of the gesture. If the smallest distance to a certain state falls below the predefined threshold, the state is recognized and the lever moves to that position. The selection and deselection posture is responsible for selecting and deselecting the

interaction object. Once selected, the lever can be manipulated by one of the different gestures in a continuous fashion. To avoid interference between the three gestures for manipulating the lever, only one of those gestures can be used at a time and we manually switch between gestures. The lever can be positioned at 31 different states, defined by keyframes in the trajectory of the currently selected gesture. This level of tolerance can again be modified depending on the embedding and requirements of the task.

For a better understanding of the interaction, **Figure 5** shows the user interacting with the virtual lever displayed as a state machine. At the beginning, the lever is not selected and the user moves freely in the space. Upon reaching the selection and deselection posture, the lever is activated. The user can now manipulate the lever by performing the gesture “squeezing” within the tolerated threshold.

By following these four steps, we create an exemplary interaction task using only a small amount of data and simple mathematical operations such as the Euclidean distance between points in the low-dimensional embedding. We demonstrated how low-dimensional representations can be used to simplify the design process and help designers leverage quick prototypes.

Note that if different sensors are used, or extra sensors are added to an existing interface, we would collect new calibration data and repeat the aforementioned design process. The core interaction mechanism can be kept, but we would need to retrain the embedding and then redefine the regions associated with state transitions.

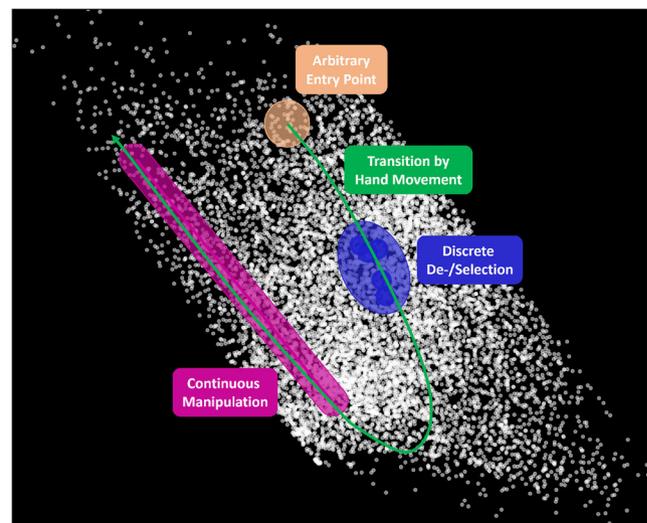


Figure 5. Simplified interaction with the virtual lever using the gesture “pinching with index finger and thumb” shown as a simple state machine. The user transitions between states according to the trajectory (green). The user begins the interaction at an arbitrary position in space, marked as *arbitrary entry point* (orange), making their way toward the *discrete selection and deselection state* (blue). Once there, the lever is selected and a discrete state transition occurs. Now, the user approaches the *continuous manipulation state* (magenta), where they can use the gesture “pinching with index finger and thumb” to manipulate the lever in a continuous fashion.

6. Evaluating the Interaction

In the last two sections, we position the potential contribution of the use of low-dimensional embeddings to the design process of gestural interaction systems and demonstrate the practical application of embeddings to prototype interaction tasks. In this section, we want to evaluate the exemplary interaction task of Section 5 and investigate the user performance when interacting within the low-dimensional space.

To do this, we perform an exemplary user study applying Fitts' law,^[39] where we compare the three different gesture candidates "pinching with index finger and thumb," "pinching with index finger, middle finger, and thumb," and "squeezing." The aim of this study is not to generalize results but to show an example of possible evaluation routines for gestures and demonstrate how designers can determine the best fit for their interaction prototype.

Two users (demographics mentioned in Section 3.1), who were not part of the data collection process described in Section 5, took part in this study. During the study, a randomly sized red cube appeared at a random position on the lever visualized in **Figure 6** for the users to hit. Once hit, the users were instructed to press a button, which completed the trial and caused the target to reappear at a new location. The users were instructed to hit the cube $n = 200$ times for each of the three gestures. For each trial, we recorded the distance between the current position of the lever and the spawn position of

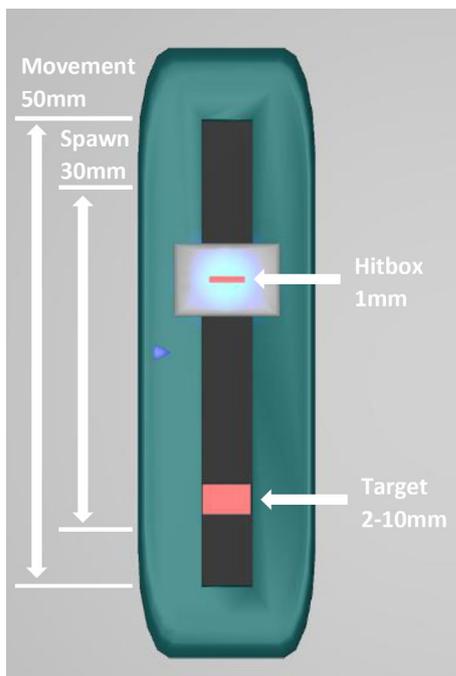


Figure 6. The setup during the Fitts' law user study. The red cube (target) appears at random locations within its boundaries (spawn). The user has to hit the cube with the lever (hitbox) by moving within the boundaries (movement). The user can move the lever by performing one of the following gestures: "pinching with index finger and thumb," "pinching with index finger, middle finger, and thumb," or "squeezing".

the cube (D) as well as its width (W) and the time passed until the cube was hit by the lever (MT). We evaluated the recorded data with the help of the Shannon form^[40] of Fitts' law,^[39] by calculating the index of difficulty (ID) and index of performance (IP) as

$$ID = \log_2 \left(\frac{D}{W} + 1 \right) \quad (1)$$

$$IP = \left(\frac{ID}{MT} \right) \quad (2)$$

$$MT = a + bID \quad (3)$$

where D refers to the distance to the target, W to the width of the target, and MT to the movement time required to reach the target. a and b refer to the offset and slope of the linear model, which we fit to the gathered data.

Additionally, we perform a post hoc statistical analysis to assess the Pearson correlation coefficient (PCC),^[41] which is used to reveal statistical correlations between two variables. We visualize the results in **Figure 7**. As the PCC ranges between 0.46 and 0.55, we observe a linear correlation between ID and MT , which becomes evident when fitting a linear function to the data. This finding suggests that a higher movement time is attributed to a higher difficulty of the trial.

When comparing the different gesture candidates, we observe a similar performance across the all gestures. The gesture "pinching with index finger and thumb," however, stands out by having the best IP . This indicates that the user had the least trouble using the gesture "squeezing" and thus performed better than using the other gestures.

When looking at the overall distribution of samples between the two users, individual patterns arise. We note that the user represented by the color green had a higher number of outliers, especially for the gesture "squeezing." This can be caused by low familiarity with the task, variable executions of speed, or a lower agility when performing the gestures. The linear trend in movement time with ID is however clear in both cases.

With our exemplary user study, we show how designers can quickly compare different gestures when prototyping interaction tasks. In addition, we show that it suffices to gather reference movement data from one user, which are then compatible with different users. This insight further lowers the effort for prototyping simple interaction tasks. In this context, Fitts' law offers a simple method for assessing the user performance with limited resources. However, this method does not give insight into usability or user satisfaction, which requires more extensive future study.

7. Discussion

In this work, we investigate the following research question.

Can Autoencoder-Based Dimensionality Reduction Simplify the Data-Driven Design Process?

By conducting exploratory research with exemplary data, we present a framework for the use of autoencoder-based dimensionality reduction to simplify data-driven design,

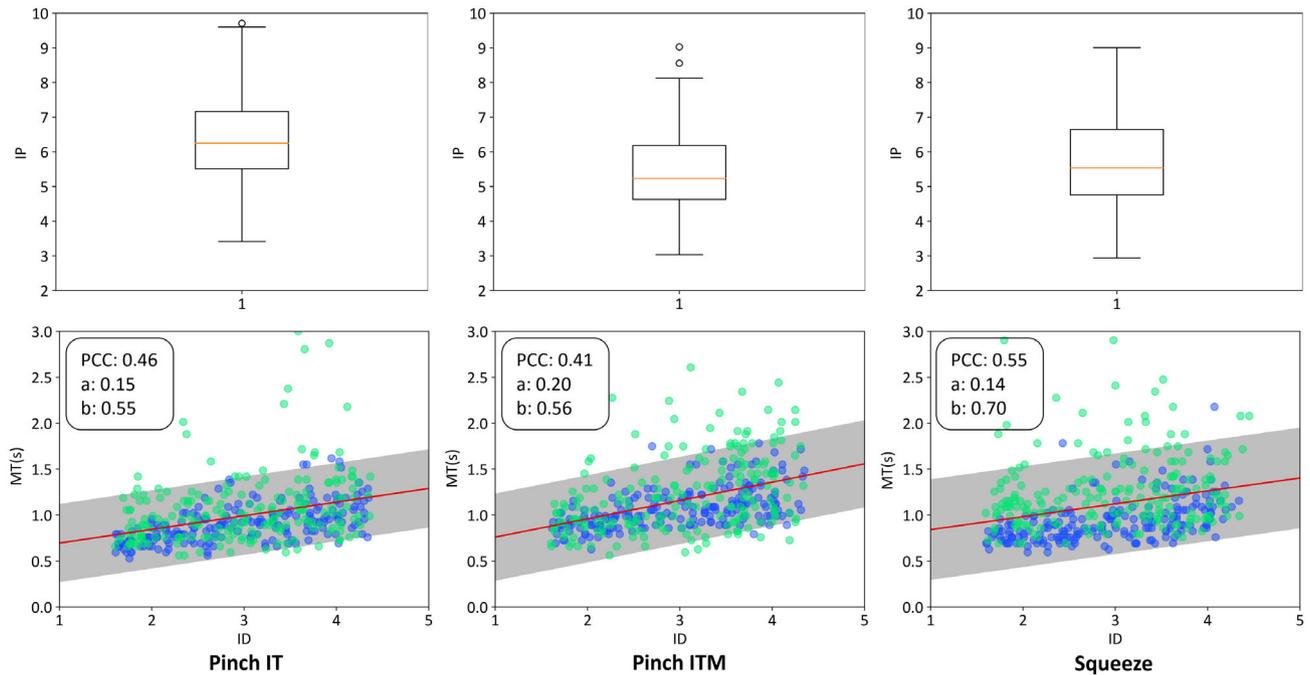


Figure 7. Boxplots showing the index of performance (IP) for realizations of the gestures “pinching with index finger and thumb” (left), “pinching with index finger, middle finger, and thumb,” (middle) and “squeezing” (right). Line plots showing the index of difficulty (ID) in relation to the movement time (MT) and displaying the Pearson correlation coefficient (PCC) and the linear coefficients a and b of a linear fit. The two users are distinguished by color (green and blue).

supporting the process of linking machine learning and design teams via low-dimensional embeddings of sensor data.

We record general hand movements, as well as everyday gestures (Section 3.1), and apply autoencoder-based dimensionality reduction (Section 3.2). We visualize the data in a low-dimensional context and propose various characteristics of embeddings, which impact their use in interaction design (Section 4). Via illustrative examples, we emulate the design process of gesture interaction by creating the interaction with a virtual level (Section 5) and evaluate the user performance with Fitts’ Law (Section 6).

Our findings suggest that autoencoders can be used to create useful low-dimensional representations of complex human movement, which are suitable for designing interaction tasks in a more comprehensible environment compared to high-dimensional data. We now provide an overview of the main research topics discussed in this article.

7.1. Create Comprehensible Data

Designing interactions in a high-dimensional context can be difficult for system engineers. Low-dimensional embeddings simplify the design process by reducing the complexity of the movement space, thus making the search for a suitable low-dimensional embedding vital for useful interaction design. 2D embeddings are a good fit because they can be visualized comprehensively in scatterplots, allowing designers to better understand the data.

7.2. Find the Right Sensors

Low-dimensional representations of high-dimensional movement provide valuable information on potentially useful subsets of sensors for specific interactions. As shown in Section 4, a broad set of sensors gathers information on a more complex movement space while also causing increased variability in the low-dimensional embedding, or requiring more training data. The importance of finding a balance between the number of sensors and the data required becomes evident when looking at Embedding C in Figure 2. Designers can use low-dimensional embeddings to guide the choice of the sensor subset best suited for their application and thus create more robust interactions, while also reducing the amount of required hardware.

7.3. Generality

This characteristic describes the extensiveness of an embedding and its ability to encompass all possible motions detectable with the given sensors. Assessing generality reveals to what extent the training data cover the possible movement space and allows designers to compare different input modalities and adjust the amount of required data for their desired application. Embedding C in Figure 2 for example contains a larger number of outliers compared to Embedding A, which indicates that larger amounts of data per user are necessary to cover the entire individual movement space. As shown by this example, low-dimensional representations of the data reduce the complexity and allow for a quick visual interpretation of this characteristic.

7.4. Connectivity

We define this characteristic as the preservation of smooth transitions within the low-dimensional space. It encompasses an embedding's ability to map neighboring high-dimensional points to neighboring low-dimensional points, with neighboring points referring to points in spatial and temporal proximity to each other. Connectivity gives insight into what features of the hand movement are decisive and shape the movement space and allow designers to adjust their gesture interaction accordingly. As an example, we can see in Figure 3 that the decisive gesture in Embedding A is the transition from a fist to a stretched-out hand. This 2D glyph visualization demonstrates how embeddings reveal otherwise complex features of the high-dimensional data.

7.5. Variability

This characteristic describes the similarity of multiple executions of a single gesture. We differentiate between within-user variability and between-user variability. Within-user variability shows how repetitions of a gesture differ when performed by the same user, while between-user variability reveals how repetitions of a gesture differ across multiple users. Assessment of the variability of a gesture provides knowledge on how reliable that gesture is and whether users perform it consistently. This insight allows designers to choose well-defined gestures with low variability for their design prototype. Embedding A in Figure 2, for example, shows that the gesture "squeezing" is similar both within and between users. The gesture "pinching with index finger and thumb," in contrast, differs between users, which demonstrates the loose constraints of that gesture. Low-dimensional embeddings prove helpful because they are suitable for visualizing gestures as trajectories, which reduces the complexity of the data and offers more comprehensible visualizations, helping team members communicate effectively.

As proposed by Bernstein, there is typically more than one way to achieve a particular behavior.^[42] This suggests the potential for variability between the trajectories of repeated gestures. The same gesture executed in different ways may be represented as disconnected trajectories in the low-dimensional embedding space. This potential issue, however, did not appear in the exemplary movements we analyzed. Although there was an expected amount of variability, such as for the gesture "pinching with index finger and thumb," the low-dimensional representation was still contiguous. However, we might expect noncontiguous mappings in other cases, and these can still be accommodated within the design process, as multiple equivalent paths. For this reason, we anticipate that embeddings can support the design process of more complex hand movements by grasping the variability of the movement space and creating families of easily visualized and useful mappings.

7.6. Distinguishability

Distinguishability describes how much gestures differ from one another and whether a system will be able to reliably tell them apart. Knowledge of the distinguishability of gestures is especially useful to avoid classification errors, where different gestures control different interactive devices. In the 2D space, possible conflicts between gestures are visually recognizable,

allowing designers to avoid confusions of controls early during the prototyping process. As an example, the gestures "pinching with index finger and thumb" and "squeezing" in Embedding C in Figure 2 overlap significantly, indicating possible classification errors, if these were both used for gestures in the same interface.

7.7. Design Gesture Interaction

Low-dimensional representations further benefit interaction design by allowing interaction mechanisms to take place within the low-dimensional context. As shown in Section 5, we designed an interaction with a virtual lever using unsophisticated mathematical operations such as the Euclidean distance. We utilized dimensionality reduction to find a suitable set of sensors, choose a pool of potential gesture candidates, and perform calculations necessary to interact with the lever, all within the low-dimensional space. Our framework shows an exemplary design process and demonstrates how designers can benefit from dimensionality reduction to quickly implement interaction prototypes. In addition, we show that a small amount of reference data suffices to prototype simple interactions, which reduces time and effort for designers.

7.8. Assess User Performance

We conducted an exemplary user study applying Fitts' Law to evaluate the user performance when interacting with the virtual lever using different gesture candidates in Section 6. Our post hoc analysis indicates a linear correlation between the ID and the MT when attempting to hit targets of random size and distance by operating the virtual lever. Further, we found that the gesture "pinching with index finger and thumb" had the highest IP, which suggests that the users performed best using this gesture candidate. With this exemplary study, we show how Fitts' law functions as a simple method for assessing the user performance and compare different gesture candidates for an interaction task.

All in all, our work explores various aspects of dimensionality reduction for a data-driven design process. Our contribution mainly consists of the theoretical characteristics *generality*, *connectivity*, *variability*, and *distinguishability*, which provide valuable knowledge when designing interaction prototypes and can be evaluated within the low-dimensional context. Furthermore, we provide exemplary frameworks for designing and evaluating simple gesture interactions within the low-dimensional embeddings to help designers leverage quick results with low effort.

8. Conclusion

Analyzing high-dimensional interactions in the low-dimensional space facilitates data-driven interaction design. Working with low-dimensional embeddings decreases the complexity of movement data to a manageable level, allowing designers to better comprehend, prototype, and evaluate interaction designs. In this research, we demonstrate a scalable approach where autoencoders are used to construct low-dimensional embeddings of high-dimensional movement data. This article introduces key characteristics for assessing embeddings and demonstrates the

design process of gestural interaction via a low-dimensional embedding.

These contributions build the foundation for future user studies for systematic evaluation of low-dimensional embeddings. Further studies are needed to verify the applicability of the explored concepts to real-life design scenarios and assess the impact of the established characteristics. Key points to investigate will include the impact on variability of real-time feedback during and after gesture execution, the scalability of the approach to more high-dimensional sensor data, and the ability of the method to dampen noisy sensor data.

As engineering teams develop rich new sensing technologies for user input, these may be much higher-dimensional and more variable than traditional input approaches. Machine learning techniques such as the use of autoencoders for low-dimensional mappings have the potential to better support the use of such new sensors in interaction design. This article outlines the broad issues of the approach for enabling future applications.

Acknowledgements

J.H.W. and R.M.-S. acknowledge funding support from EPSRC grant EP/R018634/1, Closed-loop Data Science. R.M.-S. also acknowledges funding support from EPSRC grants EP/M01326X/1, EP/T00097X/1 (QuantIC, the UK Quantum Technology Hub in Quantum Enhanced Imaging). M.M.R. and S.Y.S. were partly funded by a Nokia donation to the University of Glasgow. This work was also supported by the AMPLIFY project, which received funding from the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (grant agreement no. 683008). The authors thank Andrew Ramsay for support with acquisition of further experimental data. The experiments involving human subjects have been performed with the full, informed consent of the volunteers. *The Interactive Supporting Information of this article can be found at:* <https://authorea.com/doi/full/10.22541/au.162206321.17668649/v1>.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are openly available in "DexmoResearch" at <https://git.dcs.gla.ac.uk/svenja/dexmoresearch>, reference number 42.

Keywords

autoencoders, data gloves, dimensionality reduction, embedding, interaction

Received: March 16, 2021

Revised: May 23, 2021

Published online:

- [1] L. Dipietro, A. M. Sabatini, P. Dario, *IEEE Trans. Syst. Man Cyber. Part C: Appl. Rev.* **2008**, *38*, 461.
- [2] E. Remelli, A. Tkach, A. Tagliasacchi, M. Pauly, in *017 IEEE Int. Conf. on Computer Vision (ICCV)*, IEEE, New York **2017**, pp. 2554–2562.

- [3] J. Shukla, A. Dwivedi, in *2014 Fourth Int. Conf. on Communication Systems and Network Technologies*, IEEE, New York **2014**, pp. 919–923.
- [4] Y. Yan, J. M. Goodman, D. D. Moore, S. A. Solla, S. J. Bensmaia, *Nat. Commun.* **2020**, *11*, 1.
- [5] R. Vinjamuri, M. Sun, C.-C. Chang, H.-N. Lee, R. J. Sciabassi, Z.-H. Mao, *IEEE Trans. Biomed. Eng.* **2010**, *57*, 284.
- [6] M. Achibet, G. Casiez, M. Marchal, in *2016 IEEE Symp. on 3D User Interfaces (3DUI)*, IEEE, New York **2016**, pp. 3–12.
- [7] R. J. Jagacinski, J. M. Flach, *Control Theory for Humans: Quantitative Approaches to Model Performance*, Lawrence Erlbaum Associates, Mahwah, NJ **2003**.
- [8] C. Lu, J. Wang, L. Jing, in *Proc. of the 18th Conf. on Embedded Networked Sensor Systems*, SenSys '20, Association for Computing Machinery, New York **2020**, pp. 619–620.
- [9] C. Gorenflo, L. Golab, S. Keshav, *SSDBM '17*, Association for Computing Machinery, New York **2017**.
- [10] M. Ciocarlie, C. Goldfeder, P. Allen, in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, IEEE, New York **2007**, pp. 3270–3275.
- [11] V. I. Pavlovic, R. Sharma, T. S. Huang, *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 677.
- [12] R. Xu, S. Zhou, W. J. Li, *IEEE Sens. J.* **2012**, *12*, 1166.
- [13] D. Wu, L. Pigou, P.-J. Kindermans, N. D.-H. Le, L. Shao, J. Dambre, J.-M. Odobez, *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 1583.
- [14] R.-D. Vatavu, in *Proc. of the 2019 CHI Conf. on Human Factors in Computing Systems*, CHI'19, Association for Computing Machinery, New York **2019**, pp. 1–13.
- [15] N. Soni, S. Gleaves, H. Neff, S. Morrison-Smith, S. Esmaeili, I. Mayne, S. Bapat, C. Schuman, K. A. Stofer, L. Anthony, in *Proc. of the 2020 CHI Conf. on Human Factors in Computing Systems*, CHI '20, Association for Computing Machinery, New York **2020**, pp. 1–12.
- [16] R.-D. Vatavu, J. O. Wobbrock, in *Proc. of the 33rd Annual ACM Conf. on Human Factors in Computing Systems*, CHI '15, Association for Computing Machinery, New York **2015**, pp. 1325–1334.
- [17] A. D. W. Jacob, O. Wobbrock, M. R. Morris, in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, CHI '09, Association for Computing Machinery, New York **2009**, pp. 1083–1092.
- [18] J. O. Wobbrock, H. H. Aung, B. Rothrock, B. A. Myers, in *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, Association for Computing Machinery, New York **2005**, pp. 1869–1872.
- [19] T. Alldridge, E. Barker, M. Barlow, S. Sutherland-Dee, X. X. Teh, A. Roudaut, in *Extended Abstracts of the 2020 CHI Conf. on Human Factors in Computing Systems*, CHI EA '20, Association for Computing Machinery, New York **2020**, pp. 1–7.
- [20] S. Sundaram, P. Kellnhofer, Y. Li, J.-Y. Zhu, A. Torralba, W. Matusik, *Nature* **2019**, *569*, 698.
- [21] R. Takada, J. Kadomoto, B. Shizuki, in *Extended Abstracts of the 2019 CHI Conf. on Human Factors in Computing Systems*, CHI EA '19, Association for Computing Machinery, New York **2019**, pp. 1–4.
- [22] X. Gu, Y. Zhang, W. Sun, Y. Bian, D. Zhou, P. O. Kristensson, in *Proc. of the 2016 CHI Conf. on Human Factors in Computing Systems*, CHI '16, Association for Computing Machinery, New York **2016**, pp. 1991–1995.
- [23] T. von Marcard, B. Rosenhahn, M. Black, G. Pons-Moll, *Comput. Graphics Forum* **2017**, *36*, 349.
- [24] Y. Zhang, C. Harrison, in *Proc. of the 28th Annual ACM Symp. on User Interface Software & Technology*, UIST '15, Association for Computing Machinery, New York **2015**, pp. 167–173.
- [25] A. Gisbrecht, A. Schulz, B. Hammer, *Neurocomputing* **2015**, *147*, 71.
- [26] B. Abolhasanzadeh, in *2015 7th Conf. on Information and Knowledge Technology (IKT)*, IEEE, New York **2015**, pp. 1–5.

- [27] M. Cavallo, A. Demiralp, in *Proc. of the 2018 CHI Conf. on Human Factors in Computing Systems*, CHI' 18, Association for Computing Machinery, New York **2018**, pp. 1–13.
- [28] S. Ingram, T. Munzner, *Neurocomputing* **2015**, *150*, 557.
- [29] L. Van Der Maaten, E. Postma, J. Van den Herik, Technical Report, TiCC, Tilburg University, **2009**.
- [30] G. E. Hinton, R. R. Salakhutdinov, *Science* **2006**, *313*, 504.
- [31] S. S. Fels, G. E. Hinton, *IEEE Trans. Neural Networks* **1993**, *4*, 2.
- [32] S. S. Fels, G. E. Hinton, *IEEE Trans. Neural Networks* **1998**, *9*, 205.
- [33] Y. Visell, J. Cooperstock, in *2007 IEEE Int. Conf. on Systems, Man and Cybernetics*, **2007**, IEEE, New York 3373–3378.
- [34] R. E. O. S. Ascari, L. Silva, R. Pereira, in *Proc. of the 25th Int. Conf. on Intelligent User Interfaces*, IUI '20, Association for Computing Machinery, New York **2020**, pp. 100–110.
- [35] J. Williamson, R. Murray-Smith, in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, CHI' 12, Association for Computing Machinery, New York **2012**, pp. 1717–1726.
- [36] PyTorch, <https://pytorch.org/> (accessed: July 2020).
- [37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimeshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, in *33rd Conf. on Neural Information Processing Systems*, Curran Associates, Inc, Vancouver, CA **2019**.
- [38] GPyOpt, <https://pypi.org/project/GPyOpt/> (accessed: June 2021).
- [39] P. M. Fitts, *J. Exp. Psychol.* **1954**, *47*, 381.
- [40] I. S. MacKenzie, Ph.D. thesis, University of Toronto, **1991**.
- [41] *Encyclopedia of Public Health* (Ed: W. Kirch), Springer, Dodrecht **2008**.
- [42] N. A. Bernstein, *The Co-Ordination and Regulation of Movements*, Pergamon Press, Oxford, UK **1967**.