

Grasp Taxonomy for Robot Assistants Inferred from Finger Pressure and Flexion

Bahareh Abbasi, Mehdi Sharifzadeh, Ehsan Noohi, Sina Parastegari, and Miloš Žefran

Department of Electrical and Computer Engineering

University of Illinois at Chicago, Chicago, IL USA

babbas3@uic.edu

Abstract—Grasp is an integral part of manipulation actions in activities of daily living and programming by demonstration is a powerful paradigm for teaching the assistive robots how to perform a grasp. Since finger configuration and finger force are the fundamental features that need to be controlled during a grasp, using these variables is a natural choice for learning by demonstration. An important question then becomes whether the existing grasp taxonomies are appropriate when one considers these modalities. The goal of our paper is to answer this question by investigating grasp patterns that can be inferred from a static analysis of the grasp data, as the object is securely grasped. Human grasp data is measured using a newly developed data glove. The data includes pressure sensor measurements from eighteen areas of the hand, and measurements from bend sensors placed at finger joints. The pressure sensor measurements are calibrated and mapped into force by employing a novel data-driven approach. Unsupervised learning is used to identify patterns for different grasp types. Multiple clustering algorithms are used to partition the data. When the results are taken in aggregate, 25 human grasp types are reduced to 9 different clusters.

Index Terms—Grasp Taxonomy, Programming by Demonstration, Unsupervised Learning, Ensemble Clustering.

I. INTRODUCTION

Robots that work together with humans need to act in a way that is predictable and non-threatening to humans [1]. Programming by demonstration (PbD), where data collected while humans perform an activity is used to program the robot, has emerged as a powerful paradigm to achieve this [2]–[5]. Among different actions of interest, object manipulation actions are particularly challenging since they are difficult to measure in naturalistic settings. Grasp represents an important phase in object manipulation actions and is the focus of our paper.

Grasp has been studied broadly in robotics. Grasp recognition is studied as part of learning by demonstration framework in [6]; grasp planning has been considered in [7]; design and control of prosthetic hands are the focus in [8]. Various taxonomies have been proposed for human grasps used during the activities of daily living (ADLs) using different criteria including kinematics of the hand during the grasp, object shape and task context [9]–[12].

Attempts at classifying the grasp type are either dynamic, where a whole time sequence of data measured during the

grasp is considered [6], [13], [14], or static, where measurements at a single point in time once the object is securely grasped are used [15], [16]. In [14], the recordings of hand motion during the grasp are transformed to a lower-dimensional space by Gaussian Process Latent Variable Models. The transformed data is used to identify clusters for a subset of human grasp types that form the comprehensive grasp taxonomy presented in [9]. In contrast, [6] considers the grasp types proposed taxonomy in [10]. Both finger movements and arm trajectory are used to train Hidden Markov Models (HMMs) to recognize the grasp type. In [17], 12 grasp types from the taxonomy presented in [11] are studied and HMMs are trained to recognize the sequences of grasp gestures by using the contact information and sequences of joint angles.

As an example of the static approach, [16] classifies grasp types in the taxonomy presented by [10] using a neural network and angular data collected from a data glove. In [15], the force patterns are investigated for 25 different grasp types from the taxonomy in [9].

Most of commonly used grasp taxonomies by researchers rely on observational cues rather than measured numerical data. Following that, in most of the human grasp classification tasks it is assumed that the grasp types in a chosen taxonomy are distinguishable from the data. However, there is no prior work on determining which grasp classes are recognizable in force and posture domains.

In this paper, we extend the approach in [15] by considering the multi-modal data, both posture and force information related to each grasp type selected from the comprehensive grasp taxonomy in [9]. We employ unsupervised learning to recognize natural clusters in the data collected in our human study; each cluster represents grasp types that share similar patterns. An important insight of our work is that while different grasp taxonomies in the literature mostly consider visual appearance and the perceived (subjective) human intent in proposing different grasp types, grasp types that can be learned from the measured data are subsets of those. In other words, some grasp types proposed in existing taxonomies can simply not be distinguished just based on the measured data.

The contribution of this paper is twofold. As noted above, we identify a set of grasp types that naturally describe grasps when considering grasp forces and finger joint angles. Since grasp forces are difficult to measure directly, pressure sensors

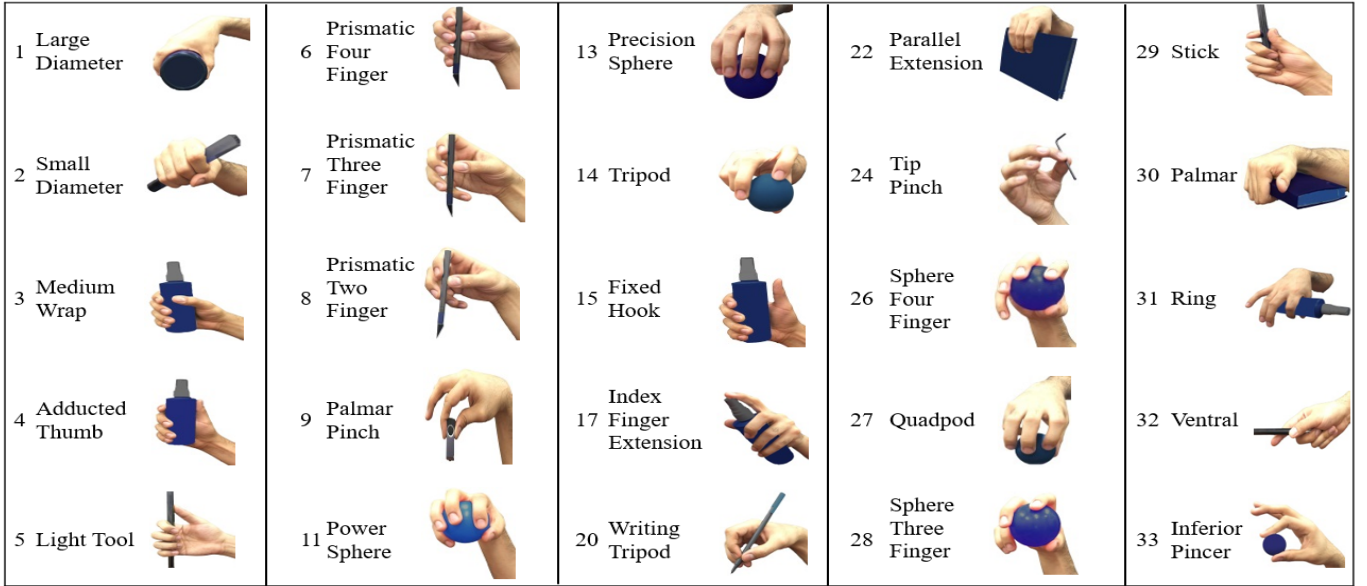


Fig. 1: The 25 grasp types from [9] that are investigated in our study.

are used instead. The second contribution of our work is thus a novel data-driven method that can be used to map pressure sensor data into grasp forces.

The rest of the paper is organized as follows. Previous work on grasp classification is reviewed in Sect. II. Sect. III describes the human study and the data collection process. In Sect. IV, our novel data-driven calibration method for mapping pressure data into grasp forces is introduced. We describe how clustering of the collected data was implemented in Sect. V and the results are discussed in Sect. VI. Finally, Sect. VII concludes the paper and outlines possible future work.

II. RELATED WORK

In this Section we briefly discuss the features that are used to characterize the human grasp types and the taxonomies proposed by previous researchers. In [12], [18], the human prehensile actions are divided into two main categories of *power*, and *precision* grasps. Kamakura et al. introduced an additional category as *intermediate* grasp in [11] and accordingly their taxonomy classified the human grasp types into 14 distinct classes. Cutkosky et al. in [10] took advantage of geometric properties of the manipulated object to classify the human grasps types.

In addition to the main grasp type, namely *power*, *precision*, and *intermediate*, the Virtual Finger (*VF*) and the opposition type are other features of the grasp. *VF* is defined as a subset of the fingers that act against the other part of the hand in terms of force/torque during the grasp [19]. *VF* formation patterns can be different among different grasp types. Subsequently, in [20], the opposition type is defined as the direction of applied forces by the two opposing parts of the hand while grasping an object. The opposition types are divided into three main categories: *palm*, *pad*, and *side* [20].

In [9], Feix et al. exploited the thumb position to further distinguish the human grasp types. They proposed a detailed

grasp classification which incorporates several grasp features including *VF*, opposition type, and the thumb position (*abducted* or *adducted*). In their taxonomy, totally 33 different grasp types are recognized, however these grasp types can fall into 17 distinct classes if the object properties are disregarded.

We select a subset of 25 grasp types from this recent taxonomy [9]. These grasp types are more frequent in ADLs [9] and are measurable with our data glove. Fig. 1 presents the list of the studied grasp types in this paper.

III. HUMAN STUDY

To measure applied force by different areas of the hand and joint angles of fingers, we developed our own data glove. It is equipped with eighteen pressure sensors (FSR400 – 402 from Interlink Electronics) on the front side of the glove, eight bend sensors (unidirectional bend sensor from Flexpoint), and an inertial measurement unit (10 DOF IMU breakout from Adafruit) on the back side of the glove. The locations of the pressure sensors are selected based on the active areas of the hand during the grasp action [15]. Three force sensitive resistors (FSR sensors) are located on each segment of the fingers except for thumb which has one. Furthermore, Five FSR sensors are mounted on the palm. Two bend sensors are positioned on each finger, one on metacarpophalangeal joint and one on proximal interphalangeal joint, except for the pinkie and thumb which have one. Sampling frequency is 100Hz using Arduino board (*Atmega* 2560). Fig.2 shows the schematic of the data glove with sensor positions.

Seven participants were recruited to perform the grasps listed in Fig. 1. They were instructed on how to perform each grasp type. Each subject performed each grasp type four to five times. In each trial, he/she started from an arbitrary initial position and grasped the object securely for a few seconds. The average of sensor readings in the *stabilized* grasp mode is computed for each trial. In order to remove the effect of

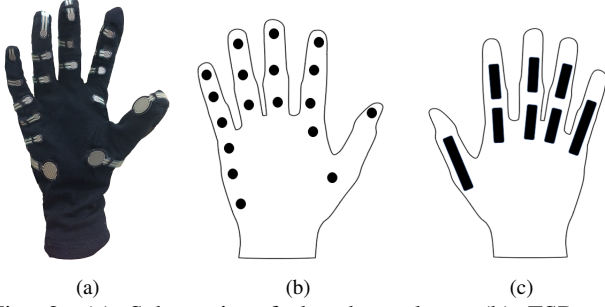


Fig. 2: (a) Schematic of the data glove. (b) FSR sensor positions. (c) Bend sensor positions.

the mass of the object on the grasp pattern, all of the objects that were used for this experiment were lightweight (less than 50g).

After removing the corrupted signals, in which some sensor wires got disconnected, a total of 604 trials were recorded. The recorded data consists of pressure and bend sensors measurements in the static phase of the grasp. To extract force pattern, pressure sensors readings are preprocessed and converted to force using a novel data-driven procedure which is discussed next.

IV. CALIBRATION PROCEDURE

Here, we introduce a data-driven approach that uses a Neural Network (NN) and a generalized sensor model to accurately calibrate all the pressure sensors at once. We compare the results with Decision Tree (DT) regression [21], and another commonly used calibration method, polynomial model [22], [23]. In particular, we show that NN-based model is able to compensate for the sensor hysteresis and estimate the force directly using the output voltage of FSR sensors.

A. Calibration Setup

In addition to FSR sensors, a Force/Torque sensor (Gamma, ATI Industrial Automation) is used as a ground truth. During the experiments, a FSR sensor is connected to a pull down resistor ($5K\Omega$) and V_{cc} (5V) and the output voltage is measured. The value of the pull down resistor determines the sensitivity range of the sensor before reaching the saturation. With our setup, the sensing range is roughly between 1 – 5N. The FSR sensor is mounted on top of the F/T sensor. Sampling process is done for both of the sensors simultaneously with approximate frequency of 110Hz. The F/T sensor is factory calibrated. In all the trials, using a method similar to [22], a button is glued to the center of the active area of the FSR sensor to assure consistency of the contact area during the trials.

In total, eleven FSR sensors were used for data collection. Each test took five minutes. In each trial, one of the FSR sensors is placed on the F/T sensor, and a random varying amplitude and frequency force signal generated by a human hand is applied to it; the data is captured from both sensors. Our choice of the force signals is motivated by the fact that

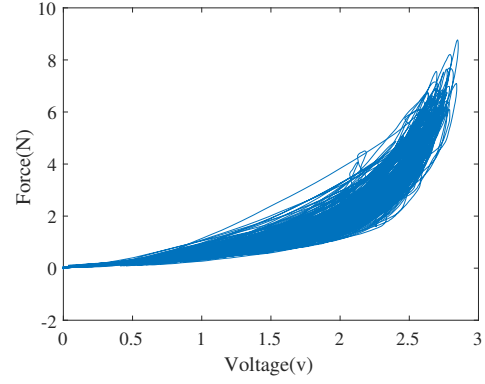


Fig. 3: The applied force measured by F/T sensor vs. output voltage measured by FSR sensor.

the target application for the FSR sensors are typically human interfaces such as data gloves in which the human applies the force. The applied forces ranged over the whole sensing range of the FSR sensors, up to the saturation region.

B. Model Description

By analyzing the data it was observed that the sensor response is not repeatable during a trial due to hysteresis and nonlinearities. This implies that it is likely insufficient to use a polynomial function to predict the applied force. Even though there exist calibration curves with specific features such as an addition of a moving integral [22], [23], time varying behavior of sensor would require continuous recalibration. Fig. 3 shows a random trial which demonstrates time dependent behavior of the sensor.

In order to accurately predict the force in applications where the contact forces vary dynamically, the nonlinearities and time-dependent features need to be modeled and compensated. To achieve that, we propose using a Neural Network (NN). The advantage of a NN is that it can model nonlinear dynamical systems with no prior assumptions on the physical model of the sensor, unlike the traditional resistor or resistor-capacitor models [24].

We introduce two NNs which we designed and fine-tuned to model nonlinearities in these sensors: a Convolutional Long Short-Term-Memory (LSTM) Neural Network (CLNN), and a deep Distributed Delay Neural Network (DDNN). In CLNN, we use two convolutional layers in network architecture to process time-series inputs and extract sample dependencies in time domain, followed by an LSTM layer, a well-known Recurrent Neural Network, to process and forecast the time series by considering the sequence of recent past events within the network (see [25]), followed by a fully connected (FC) layer.

DDNN model is one of the variations of the time series NNs and is basically a feed-forward NN model which considers the history of the system in terms of delay within its different layers. The equation describing the behavior of network can be written as:

$$y(t) = f(x(t), x(t-1), \dots, x(t-d_1)) \quad (1)$$

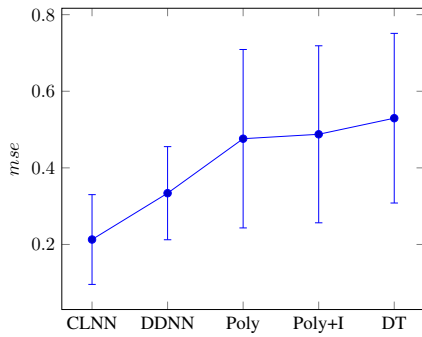


Fig. 4: Overall mean and standard deviation of methods.

where f is a nonlinear function which can predict the output y at time t from the last d_1 values of input x .

In addition to the above models, we have used decision tree (DT) regression method that also consider the sequence of past inputs, i.e. using the moving integral as an additional input for the system. For further comparison, the traditional polynomial model [22], [23] with and without the moving integral was also tested. For these methods, the current output voltage of the sensor together with a sequence of a predetermined length of previous inputs are fed to the model.

C. Model Comparison

For evaluation, all the models are trained by a trial collected from one of the sensors, and the trained models are tested on the remaining sensors. Before training the models, the measured signals by both sensors are denoised using a moving average filter with the window size of 20. In all the models, the output voltage measured by the FSR sensor and the force signal measured by the F/T sensor are the input and the desired output of the models, respectively. For performance evaluation, mean squared error (mse) is computed for each model.

The DDNN model includes four hidden layers with ten hidden neurons in each, with delays of 21, 8, 4, 2 respectively, and one output neuron in output layer. The training algorithm is Scale Conjugate Gradient Descent and the performance measure is the mean absolute error (mae). The activation functions are hyperbolic tangent sigmoid and linear for the hidden and output layers respectively.

The CLNN model includes two convolutional layers with rectifier activation, 20 and 10 filters, and 20, 3 window sizes, respectively, followed by one LSTM layer with 100 hidden neurons and a FC layer with one neuron with linear activation. The performance measure is the mae and the optimizer is Adam [26].

In DT regressor, the current input and the moving integral (the summation of the data points within 0.5 seconds delay with linearly increasing coefficients) are used as the inputs of the model. The mae is selected as the splitting criterion. To further evaluate the results, the traditional approach, a fourth-order polynomial curve with and without the moving integral input [22], [23] was tested using the least-squares fit to predict the output signal. The mean and the standard deviation for each model over all the trials are plotted in Fig. 4.

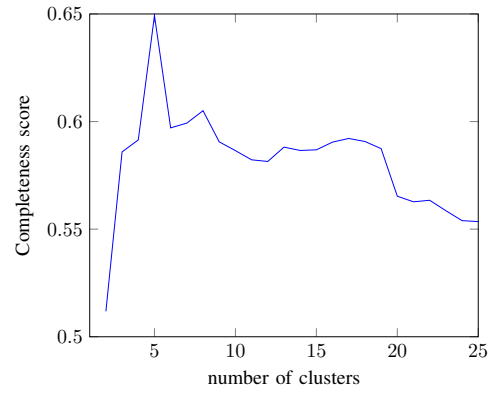


Fig. 5: The completeness score as a function of the number of the clusters.

The results show that NN-based models outperform the other methods. The CLNN model with its two convolutional layers and one LSTM layer can handle calibration and the hysteresis compensation problem due to its nonlinear memory-based modeling ability. The DDNN tries to solve the same issues by applying delay to the network. On the other hand the polynomial fitting method just fits the best fourth degree curve to the training data set. If the training data is biased or very noisy, the curve could be far from the optimal one and the accuracy drops dramatically.

Our results also show that adding a moving integral to the polynomial model does not necessarily improve the accuracy. Further, because of the polynomial form, the error can actually be multiplied. As an example, consider that the window size for the moving integral is 0.5 seconds as suggested in [22], and the frequency of the applied sinusoidal force is 100 Hz. By adding a moving integral term, no useful information is added to the model except for the noise. This shows that the window size needs to be matched to the frequency content in the applied force. The figure also suggests that although the physical features of the sensors could be slightly different from each other, a single trained network seems to be adequate as a force estimator for all the sensors. The CLNN model is best, with DDNN being the second. We thus used the CLNN model to process pressure sensor measurements.

V. DETERMINING TAXONOMY THROUGH CLUSTERING

The dataset collected during the human study has 26 features, with 18 corresponding to pressure sensors and 8 to finger bending angles. The data, the average of sensors reading in *stabilized* grasp mode in each trial, is normalized to have zero mean and unit variance for each feature. Furthermore, the grasp types are used as the ground truth label of each data point, e.g. the data points of the *Large Diameter* grasp have label 1 (see Fig. 1).

We employ unsupervised learning to partition collected multi-modal data into meaningful classes. Our goal is to find common patterns among different grasp types in terms of force and posture, i.e. to recognize grasp types which share similar patterns. Generally, majority of data points of a grasp type should appear in the same cluster since they have similar

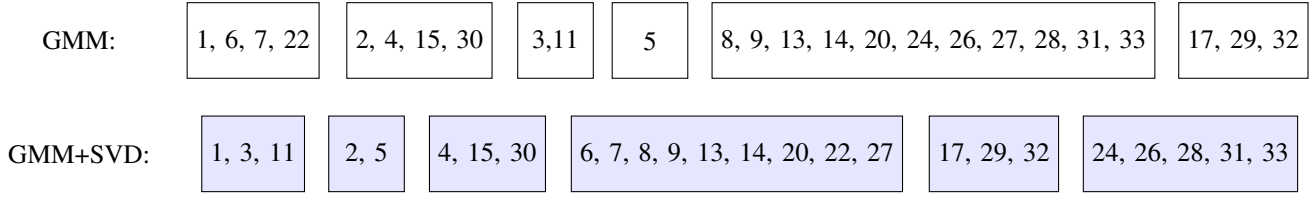


Fig. 6: Two examples of the clustering output resulting from applying GMM with and without SVD transformation.

pattern. Moreover, data points of different grasp types may appear in the same cluster indicating those grasp types have similar patterns.

Different clustering methods partition data based on different criteria. In addition, the clustering result may depend on the initialization parameters. With different parameterizations, a particular clustering algorithm can result in different outcomes. Thus, one common approach to partitioning data is to combine the results of multiple clustering instances and generate an ensemble clustering [27]. Since each clustering algorithm analyzes data using different criteria, the resulting partitions can complement each other. Furthermore, there is no need to determine number of the clusters in consensus clustering [28].

In this work, we employ different clustering algorithms and use their results to generate the ensemble clustering. The first algorithm that is exploited is *k-means* algorithm, which partitions the data points by minimizing their distances from the centroids. The initialization parameters includes the number of the clusters, i.e. the number of the centroids. Here, the *Euclidean distance* is chosen as the distance measure. The initialization of the centroids is done through *k-means++* algorithm [29]. Since our goal is to find the correlation between different grasp types, majority of the data points that belongs to a grasp type should be in the same cluster. Hence, we utilize the completeness score as a heuristic indicator to choose the optimal parameters of the algorithms, e.g. number of the clusters in *k-means*. The completeness score checks whether data points of one grasp type fall into the same cluster by comparing the clustering labels with the ground truth labels. To measure this score, we apply *k-means* algorithm to data for different number of clusters from 2 to 25 (the number of total grasp types). We record two clustering results with the highest completeness score.

Another clustering algorithm that is used to partition the data is Gaussian Mixture Model (GMM) algorithm that uses Expectation Maximization (EM) to find the representative GMMs to explain data accurately. For this clustering, the same as for *k-means*, the number of the clusters need to be determined a-priori. Similarly as before, the number of the mixture models is selected based on maximizing completeness score. The two clusterings which have the highest completeness score are selected.

Next, we utilize Singular-Value Decomposition (SVD) [30] to reduce the dimension of the data and map it to a lower-dimensional space. We choose the minimum number of the components which can explain more than 85% of the data. Accordingly, the data is transformed via SVD to the new space

and partitioned with *k-means* and GMM. As before, for each algorithm, the two clusterings with the highest completeness scores are selected. Furthermore, the data is transformed via *Random Tree Embedding* method [31], [32] to a binary high dimensional sparse feature space. Subsequently, the transformed feature space is used to cluster the data. The *hierarchical* clustering is used to cluster the data in the new space. In this clustering method, each data points is considered as an individual cluster, and at each iteration similar clusters are merged. The merging method is *complete* which links two clusters based on the maximum distance between the sample points in those clusters. Again, the top two clusterings are selected based on the completeness score. Fig. 5 shows the completeness score with respect to number of clusters for this method.

The ten resulting clusterings partition the data according to different criteria, and in different spaces. Fig. 6 depicts two of them. Due to physical limitations, we represent the grasp types with their class numbers from Fig. 1 instead of actual grasp figure. The consensus clustering, which results from the ensemble of them, can find the commonality among these different partitions and represent the results with higher confidence. To find the commonality among different clusterings, we utilize graph/hyper-graph method which forms the hyper-graph based on the clustering results and applies a cut on the graph to get the ensemble clustering. The final cut is selected based on the mutual information between the clustering results and the consensus clustering [27]. To find the final cut, we employ three different algorithms: Cluster based Similarity Partitioning Algorithm (CSPA), Hyper-Graphs Partitioning Algorithm (HGPA), and Meta Clustering Algorithm (MCLA) (refer to [33] for more details). Among the resulting consensus clusterings from these three algorithms, the one which maximizes the normalized mutual information is selected. We have utilized MATLAB implementation of the cluster-ensemble algorithms [34]. The computed mutual information for CSPA, HGPA, and MCLA are 0.68, 0.67, 0.60 respectively, therefore we select the CSPA result as our final consensus clustering.

VI. CLUSTERING RESULTS

The consensus clustering partitions data points into 9 different clusters while the original number of grasp types is 25. This implies that several grasp types are grouped together in the same cluster. In turn, we conclude that the data of those grasp types exhibits similar patterns. Fig. 7 shows the results of the consensus clustering. It is apparent that three main

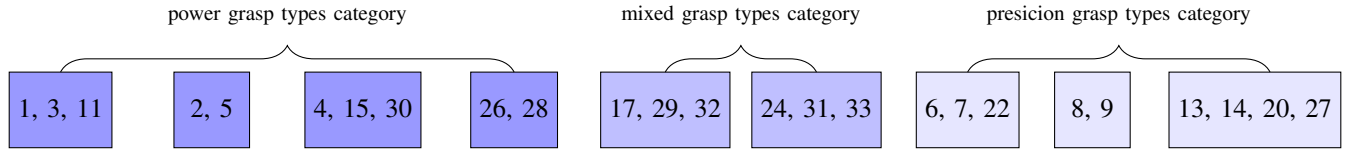


Fig. 7: The consensus clustering result.

categories of grasps can be identified within this taxonomy. The first category only includes the *power* grasp types, the second category only includes the *precision* grasp types, and the last, *mixed*, category contains the *power*, *precision*, and *intermediate* grasp types. Although the identified partitions are completely data-driven, some common features such as the opposition type, *VF*, and thumb position can be recognized among the grasp types that belong to the same cluster.

The *power* grasps category itself consists of four different clusters. The first cluster includes grasp types $\{1, 3, 11\}$ (see Fig. 1) which all have palm opposition and similar *VF* patterns. Moreover, in all of them, thumb is abducted. In the next cluster, grasp types $\{4, 15, 30\}$ are classified together. They share palm opposition, thumb adduction and similar *VF*. Grasp types $\{26, 28\}$ form another cluster and share pad opposition. The last cluster contains grasp types $\{2, 5\}$ which have similar opposition and *VF*.

The *mixed* category has the grasp types $\{17, 29, 32\}$ as one cluster and grasp types $\{24, 31, 33\}$ as the second one. In the former, although the grasp types belong to *power* and *intermediate* grasps, visually they are very similar. The only difference between them can be the grasped object, or task context. In the latter, all three grasp types, *power*, *precision*, and *intermediate*, can be seen. However, these grasp types share pad opposition and a similar *VF* pattern. The last category, *precision* grasps, includes three clusters: $\{6, 7, 22\}$, $\{8, 9\}$, and $\{13, 14, 20, 27\}$. These are all *precision* grasps and are similar to each other.

Comparing our results with [14], which only considers the dynamical hand motion for each grasp type, and [15], which only uses force distribution for different grasp types, some common patterns can be recognized even though they use different features for clustering. For example, in all three cases, [14], [15], and the present paper, grasp types $\{26, 28\}$, $\{29, 32\}$, $\{24, 33\}$, $\{6, 7\}$, and $\{13, 14, 27\}$ are clustered together.

VII. CONCLUSION AND FUTURE WORK

In this work, we studied human grasp types that occur during activities of daily living. We collected multi-modal data for 25 different human grasp types that have been identified in existing grasp taxonomies. The data includes pressure measurements from 18 different active areas on the hand, and finger joint angles. Our analysis is static and only considers measurements at a single point in time once object has been firmly grasped. Our work makes two important contributions. First, we developed a new data-driven procedure that uses Neural Networks to map pressure sensor data into

finger grasp forces. The method successfully models sensor hysteresis and nonlinearities and significantly outperforms the existing pressure sensor calibration methods. Second, unsupervised learning was employed to cluster the collected data and suggests a taxonomy in the force-flexion space. An ensemble of clustering algorithms was used and the results were subsequently combined to produce the final consensus clustering, resulting in 9 identifiable classes. It was observed that for each grasp type, the majority of the data points belong to the same cluster. This implies that the grasp types proposed in the literature share similar force and finger flexion patterns. However, several clusters combine more than one grasp type which suggests that they are perhaps not identifiable from the data. We hypothesize that these grasp types are based on our perception of the grasp intent which is informed by the context and not the data alone. Our results are important for Programming by Demonstration (PbD) of assistive robots that help with ADLs. Grasping is a major part of these activities, with finger force and finger flexion representing the natural setting for PbD. Our clustering results suggest that 9 composite grasp types may better represent what can be actually learned from human data. Our future work includes adding visual information to posture and force and investigate how this additional modality changes the grasp taxonomy.

REFERENCES

- [1] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *Human-Robot Interaction (HRI), 2013 8th ACM/IEEE International Conference on*, pp. 301–308, IEEE, 2013.
- [2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, pp. 469–483, May 2009.
- [3] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot Programming by Demonstration," in *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.), pp. 1371–1394, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
- [4] R. Zollner, T. Asfour, and R. Dillmann, "Programming by demonstration: Dual-arm manipulation tasks for humanoid robots," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 1, pp. 479–484 vol.1, Sept. 2004.
- [5] D. Lee and Y. Nakamura, "Mimesis Scheme using a Monocular Vision System on a Humanoid Robot," in *2007 IEEE International Conference on Robotics and Automation*, pp. 2162–2168, Apr. 2007.
- [6] S. Ekvall and D. Kragic, "Grasp recognition for programming by demonstration," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 748–753, IEEE, 2005.
- [7] J. Aleotti and S. Caselli, "Grasp recognition in virtual reality for robot pregrasp planning by demonstration," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 2801–2806, IEEE, 2006.
- [8] B. Massa, S. Roccella, M. C. Carrozza, and P. Dario, "Design and development of an underactuated prosthetic hand," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 4, pp. 3374–3379, IEEE, 2002.

- [9] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic, "The grasp taxonomy of human grasp types," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 66–77, 2016.
- [10] M. R. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 269–279, 1989.
- [11] N. Kamakura, M. Matsuo, H. Ishii, F. Mitsuboshi, and Y. Miura, "Patterns of static prehension in normal hands," *American Journal of Occupational Therapy*, vol. 34, no. 7, pp. 437–445, 1980.
- [12] J. R. Napier, "The prehensile movements of the human hand," *Bone & Joint Journal*, vol. 38, no. 4, pp. 902–913, 1956.
- [13] S. Ekvall and D. Kragic, "Interactive grasp learning based on human demonstration," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4, pp. 3519–3524, IEEE, 2004.
- [14] J. Romero, T. Feix, H. Kjellström, and D. Kragic, "Spatio-temporal modeling of grasping actions," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 2103–2108, IEEE, 2010.
- [15] B. Abbasi, E. Noohi, S. Parastegari, and M. Žefran, "Grasp taxonomy based on force distribution," in *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*, pp. 1098–1103, IEEE, 2016.
- [16] H. Friedrich, V. Grossmann, M. Ehrenmann, O. Rogalla, R. Zöllner, and R. Dillmann, "Towards cognitive elementary operators: grasp classification using neural network classifiers," in *Proceedings of the IASTED International Conference on Intelligent Systems and Control (ISC)*, vol. 1, pp. 88–93, 1999.
- [17] K. Bernardin, K. Ogawara, K. Ikeuchi, and R. Dillmann, "A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models," *IEEE Transactions on Robotics*, vol. 21, no. 1, pp. 47–57, 2005.
- [18] J. Landsmeer, "Power grip and precision handling," *Annals of the rheumatic diseases*, vol. 21, no. 2, p. 164, 1962.
- [19] T. Iberall, "Grasp planning from human prehension.," in *IJCAI*, vol. 87, pp. 1153–1157, 1987.
- [20] T. Iberall, "Human prehension and dexterous robot hands," *The International Journal of Robotics Research*, vol. 16, no. 3, pp. 285–299, 1997.
- [21] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.
- [22] J. Flórez and A. Velasquez, "Calibration of force sensing resistors (fsr) for static and dynamic applications," in *ANDESCON, 2010 IEEE*, pp. 1–6, IEEE, 2010.
- [23] R. S. Hall, G. T. Desmoulin, and T. E. Milner, "A technique for conditioning and calibrating force-sensing resistors for repeatable and reliable measurement of compressive force," *Journal of biomechanics*, vol. 41, no. 16, pp. 3492–3495, 2008.
- [24] L. Paredes-Madrid, P. Torruella, P. Solaeche, I. Galiana, and P. G. de Santos, "Accurate modeling of low-cost piezoresistive force sensors for haptic interfaces," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 1828–1833, IEEE, 2010.
- [25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [27] S. Vega-Pons and J. Ruiz-Shulcloper, "A survey of clustering ensemble algorithms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 03, pp. 337–372, 2011.
- [28] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 4, 2007.
- [29] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, Society for Industrial and Applied Mathematics, 2007.
- [30] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische mathematik*, vol. 14, no. 5, pp. 403–420, 1970.
- [31] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.
- [32] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," in *Advances in neural information processing systems*, pp. 985–992, 2007.
- [33] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of machine learning research*, vol. 3, no. Dec, pp. 583–617, 2002.
- [34] A. Strehl, "Clusterensemble package." <http://www.strehl.com/soft.html>.