Sparse Control of Robot Grasping from 2D Subspaces

Aggeliki Tsoli Odest Chadwicke Jenkins

Abstract— **Human control of high degree-of-freedom robotic systems is often difficult due to the overwhelming number of variables that need to be specified. Instead, we propose the use of sparse subspaces embedded within the pose space of a robotic system. Driven by human motion, we addressed this sparse control problem by uncovering 2D subspaces that allow cursor control, or eventually decoding of neural activity, to drive a robotic hand. Considering the problems in previous work related to noise in pose graph construction and motion capture, we introduced a method for denoising neighborhood graphs for embedding hand motion into 2D spaces. Such spaces allow for control of high-DOF systems using 2D interfaces such as cursor control via mouse or decoding of neural activity. We present results demonstrating our approach to interactive sparse control for successful power grasping and precision grasping using a 13 DOF robot hand.**

I. INTRODUCTION

Developing human interfaces for controlling complex robotic systems, such as humanoid robots and mechanical prosthetic arms, presents an underdetermined problem. Specifically, the amount of information a human can reasonably specify within a sufficiently small update interval is often far less a robot's degrees-of-freedom (DOFs). Consequently, basic control tasks for humans, such as reaching and grasping, are often onerous for human teleoperators of robot systems, requiring either a heavy cognitive burden or overly slow execution. Such teleoperation problems persist even for able-bodied human teleoperators given state-of-the-art sensing and actuation platforms.

The problem of teleoperation become magnified for applications to biorobotics, particularly in relation to prosthetic and assistive devices by users with lost physical functionality. In such applications, feasible sensing technologies, such as electroencephalogram (EEG) [1], electromyography (EMG) [2], [3], [4], and cortical neural implants [5], [6], provide a very limited channel for user input due to the sparsity and noise of the sensed signals. Specifically for **neural decoding**, efforts to decode these user neural activity into control signals have demonstrated success limited to 2-3 DOFs with bandwidth around 15 bits/sec. With such limited bandwidth, control applications have focused on low-DOF systems, such as 2D cursor control [7], planar mobile robots [1], and discrete control of 4 DOF robot arms [8], [9]. Additionally, Bitzer and van der Smagt [4] have performed high-DOF robot hand control by reducing the DOFs to a discrete set of pose that can be indexed by through kernel-based classification.

Robotic systems geared for general functionality or a human anthropomorphism will have significantly more than 2-3 DOFs, posing a **sparse control** problem. For instance, a prosthetic arm and hand could have around 30 DOF. While this mismatch in input and control dimensionality is problematic, it is clear that the space of valid human arm/hand poses does not fully span the space of DOFs. It is likely that plausible hand configurations exist in a significantly lower dimensional subspace arising from biomechanical redundancy and statistical studies human movement [10], [11], [12]. In general, uncovering intrinsic dimensionality of this subspace is crucial for bridging the divide between decoded user input and the production robot control commands.

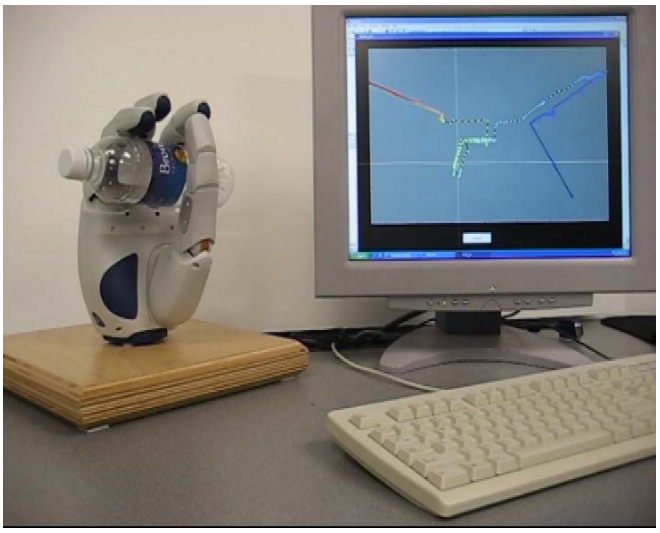


Fig. 1. Snapshot of our sparse control system driving a DLR/HIT robot hand to grasp an object from a user's 2D curson control.

In addressing the sparse control problem, our objective is to discover 2D subspaces of hand poses suitable for interactive control of a high-DOF robot hand, with the longer-term goal sparse control with 2D cursor-based neural decoding systems. We posit viable sparse control subspaces should be **scalable** (not specific to certain types of motion), **consistent** (two dissimilar poses are not proximal/close in the subspace), and **continuity-preserving** (poses near in sequence should remain proximal in the subspace). To uncover control subspaces, we follow a **data-driven** approach to this problem through the application of manifold learning (i.e., dimension reduction) to hand motion data, motion captured from real human subjects.

Our previous work [13] identified **noise in both in the motion capture and pose graph construction procedures** as major limiting factors in uncovering subspaces for sparse control. In this paper, we address both of these limitations through: 1) graph denoising using probabilistic belief propa-

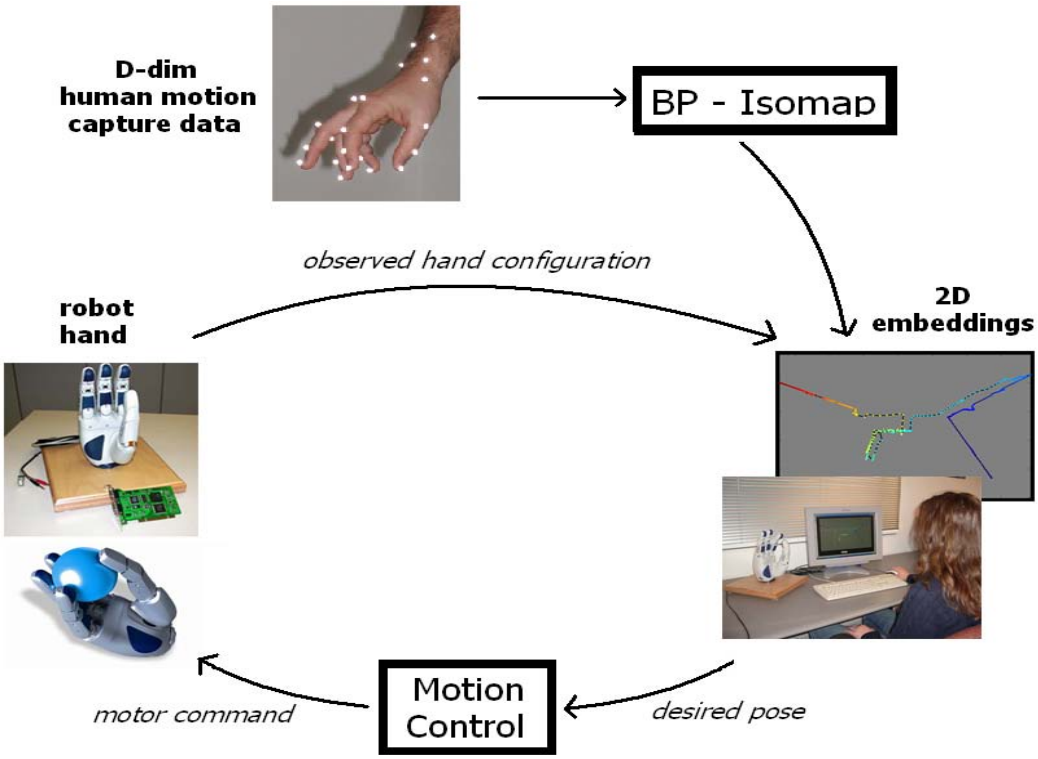


Fig. 2. Diagram for hand control by the user using human hand motion capture data for training

gation [14] and 2) more careful selection of motion capture data. We present results from embedding power grasps, precision grasps, and tapping motions into sparse control spaces and their use for interactive control of the DLR robot hand (17 DOFs).

II. THE SPARSE CONTROL PROBLEM

The essence of the sparse control problem is to estimate a control mapping $f : X \rightarrow Y$ that maps coordinates in a 2-dimensional control space $x \in \Re^2$ into the space of hand poses $y \in \Re^d$, where d is the number of DOFs expressing hand pose. The estimation of the mapping f is founded upon the assumption that the space of plausible hand poses for desired motion is intrinsically parameterized by a low-dimensional manifold subspace. We assume each hand pose achieved by a human is an example generated within this manifold subspace. It is given that the true manifold subspace of hand poses is likely to have dimensionality greater than two. With an appropriate dimension reduction technique, however, we can preserve as much of the intrinsic variance as possible. As improvements in user interfaces (namely for neural decoding) occur, the dimensionality of the input signal will increase but we will still leverage the same control mapping.

Tangenting briefly, the application of sparse control for interactive control of the DLR/HIT hand is illustrated in Figure 2. Human hand motion data in high-dimensional pose space is given as input. Using manifold learning, the hand pose data is embedded into a 2D space. The embedding space is presented to a human user through a Matlab graphical interface. Every time the user clicks on a point in this space, the 2D input coordinates s translated to a high-dimensional hand configuration, serving as the target joint angles for actuating the robot hand. The user can observe the results of their action and interactively guide the performance of the robot hand.

We create a control mapping by taking as input a set of training hand poses $y_i \in \Re^d$, embedding this data into control space coordinates $x_i \in \Re^2$, and generalizing to new data. The configuration of points in control space $x_i = f^{-1}(y_i)$ is latent and represents the inverse of the control mapping. Dimension reduction estimates the latent coordinates y such that distances between datapairs preserve some criteria of similarity. Each dimension reduction method has a different notion of pairwise similarity and, thus, a unique view of the intrinsic structure of the data. Once embedded, the pose-subspace pairs (y_i, x_i) are generalized into a mapping through interpolation [15] to allow for new (out-of-sample) points to be mapped between input and control spaces.

Discovery of the sparse control mapping is performed using Isomap [16]. We focus on Isomap, but have also explored the use of other dimension reduction techniques (PCA, Hessian LLE, Spatio-temporal Isomap) [13]. Isomap is basically a "geodesic" form of multidimensional scaling (MDS) [17], where shortest-path distances in pose space represent desired Euclidean distances the control subspace. Isomap constructs the approximation of geodesic distance (contained in the matrix D):

$$D_{y,y'} = \min_p \sum_i D'(p_i, p_{i+1}) \quad (1)$$

where D' is a sparse pose graph of local distances between nearest neighbors and p is a sequence of points through D' indicating the shortest path between poses y and y'. MDS is performed on the matrix D to generate subspace coordinates x based on the distance preserving error E (which can be optimized efficiently thorugh eigendecomposition):

$$E = \sqrt{\sum_x \sum_{x'} (\sqrt{(x-x')^2}) - D_{y,y'})^2} \quad (2)$$

A canonical Isomap example is the "Swiss roll" dataset (Figure 1), where input data generated by 2D manifold is contorted into a "roll" in 3D. Given a sufficient density of samples and proper selections of neighborhoods, Isomap able to flatten this Swiss roll data into its original 2D parameterization, within an affine transformation.

In practice, however, noise-free nearest neighborhood construction can be difficult and prohibit the application of Isomap to noisy datasets, such as motion capture data. In the Swiss Roll example, the inclusion of a noisy neighborhood edge between points at the start and end of the manifold creates a "short circuit" for shortest path computation. Consequently, the approximation geodesic distance is invalid and the resulting embedding lacks consistency with the manifolds' true parameterization.

III. NEIGHBORHOOD DENOISING WITH BELIEF PROPAGATION

To enable application of Isomap to noisy data, we propose BP-Isomap a method for denoising a neighborhood graph in pose space using probabilistic loopy belief propagation. BP-Isomap consists of three steps: 1) construction of a neighborhood graph between hand poses using k-nearest neighbors, 2) denoising of neighborhood edges and 3) embedding of the denoised neighborhood graph using Isomap. Step 2, neighborhood denoising, is the primary focus of this section. For denoising, BP-Isomap attempts to estimate the true latent distance of a neighborhood edge x_{ij} between two points y_i and y_j given the distance of their observed neighborhood edge $y_{ij} = D'(y_i, y_j) = ||y_i - y_j||$. Once x_{ij} has been estimated for all neighboring pairs $(i,j) \in D'$, edges with distances greater than an allowed threshold τ are removed for the denoised neighborhood graph $\hat{D}'$.

The denoising procedure used by BP-Isomap follows the formulation of a Markov Random Field (MRF) as described by Yedidia et. al [14]. This formulation maintains a probability distribution (or belief) about each latent variable x_{ij} (neighborhood edge distance). The belief $b_{ij}(x_{ij})$ is formed as the product of incoming messages $m_{jm \to ij}$ from other latent variables x_{mi} and a local evidence function $\phi_{ij}(x_{ij}, y_{ij})$:

$$b_{ij}(x_{ij}) = k\phi_{ij}(x_{ij}, y_{ij}) \prod_{jm \in \text{adj}(ij)} m_{jm \to ij}(x_{ij}) \quad (3)$$

where k is a normalization constant and adj(ij) is the set of edges adjacent to edge ij. In terms of MRFs, each neighborhood edge is a vertex in the message passing structure and the connectivity between these vertices are defined by the adjacency of their neighborhood edges. For computational simplicity, we assume the belief $b_{ij}(x_{ij})$ is a discrete distribution representing probabilities over a given set of fixed distances.

The local evidence function $\phi_{ij}(x_{ij}, y_{ij})$ inclines the edge distance x_{ij} to preserve the observed Euclidean edge distance y_{ij}. This function weights possible values of x_{ij} with a Gaussian distribution centered at y_{ij} with variance σ^2:

$$\phi(d_{ij}|y_{ij}) \sim \mathcal{N}(y_{ij}, \sigma^2) \quad (4)$$

Messages $m_{jm \to ij}$ to ij incoming from adjacent edges jm are formed using the following message update rule:

$$m_{jm \to ij}(x_{ij}) \propto \sum_{x_{jm}} \phi(x_{jm}, y_{jm}) \psi(x_{ij}, x_{jm}) \prod_{x_{mk} \in \text{adj}(jm) - ij} m_{mk \to jm}(x_{jm}) \quad (5)$$

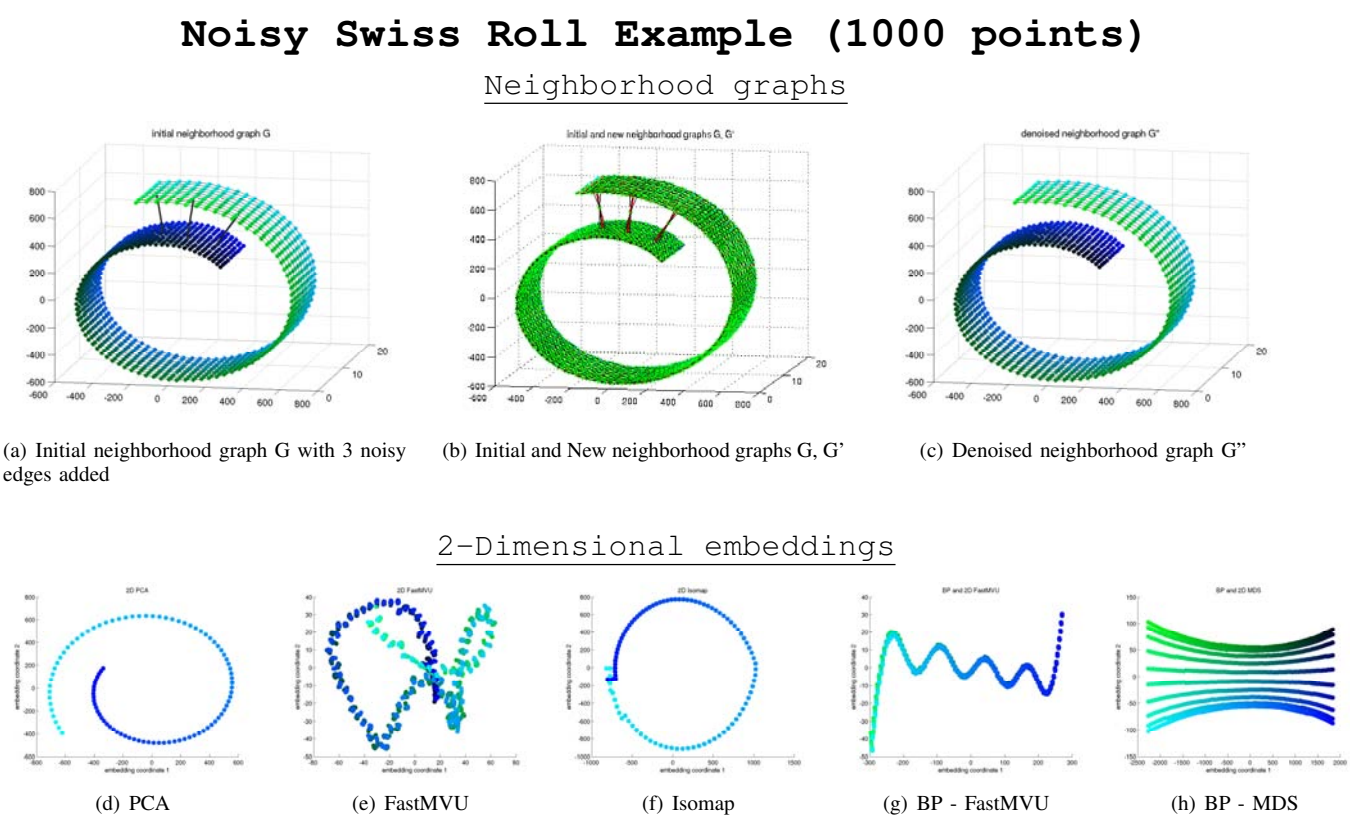
The compatibility function $\psi(x_{ij}, x_{jm})$ outputs scalar values proportional to the compatibility of a specific edge distance of x_{ij} with another edge distance x_{jm}. This function considers two cases for the relation between data points y_i and y_m (which are adjacent to a common point y_j in D'): 1) vertices y_i and y_m are adjacent or share a common neighbor $y_k \neq y_j$, or 2) vertices y_i and y_m are neither adjacent nor have common neighbors. In both cases, we are concerned with weighting the compatibility of x_{ij} and x_{jm} by the triangle they form, specifically via the distance of a third edge d_{mi}:

$$d_{mi} = ||v_{mi}|| \quad (6)$$

$$v_{mi} = -x_{ij} \frac{y_i - y_j}{||y_i - y_j||} + x_{jm} \frac{y_m - y_j}{||y_m - y_j||} \quad (7)$$

In the first case, we consider common neighbors to indicate external validation for variable x_{jm} to consider points y_i and y_j to be neighbors. Consequently, the compatibility prefers the triangle to be maintained and d_{mi} to be roughly equal to the observed Euclidean distance y_{mi}. In the second case, x_{jm} considers a neighborhood edge between y_i and y_j to be noise, preferring the distance d_{mi} to be as far as possible. We enforce these two cases in the compatibility function using a Gaussian distribution centered on y_{mi}, in the common neighbor case, and a logistic sigmoid function, in the distal case:

$$\psi(x_{ij}, x_{jm}) \sim \begin{cases} \mathcal{N}(y_{mi}, \sigma^2), & \text{if } y_m \in \text{adj}(y_i) \\ & \text{or } \exists\ k \text{ s.t. } y_k \in \text{adj}(y_i) \\ & \text{and } y_k \in \text{adj}(y_m) \\ \texttt{logsig}(0.2(d_{mi} - 1.8y_{mi}), & \text{otherwise} \end{cases} \quad (8)$$



(a) Initial neighborhood graph G with 3 noisy edges added

(b) Initial and New neighborhood graphs G, G'

(c) Denoised neighborhood graph G"

(d) PCA

(e) FastMVU

(f) Isomap

(g) BP - FastMVU

(h) BP - MDS

Fig. 3. **Noisy "Swiss Roll" example (1000 points): the data initial noisy neighborhood graph (a), with denoised edges highlighted (b) and then removed (c). A comparison between 2D embeddings of the original neighborhood graph with (d) PCA, (e) Isomap, (f) FastMVU, (g) BP combined with Isomap, (h) BP combined with FastMVU. PCA and Isomap are unable to preserve consistency due to graph "short circuiting". Adding a denoising step allows for proper embedding into two dimensions.**

The constants in these cases were found through informal experimentation and are considered user parameters.

The denoising procedure begins by considering all belief distributions to be uniform, with all distance values being equally probable. The procedure works continually updates the messages by selecting an edge pair at random and updating it using Equation 5. The procedure continues until convergence, with convergence properties described by Yedidia et al.[14].

IV. PRELIMINARY RESULTS

We present preliminary results from neighborhood denoising for manifold learning and interactive sparse 2D control of the DLR/HIT hand.

A. Swiss Roll Denoising

To evaluate our denoising procedure, we generated 3D Swiss Roll dataset by transforming data parameterized by a planar 2D bordered manifold. The ground truth geodesic distances were known based on the 2D coordinates used to seed the Swiss Roll generation. The neighborhood graph of this data was corrupted by adding three non-adjacent noisy edges between disparate datapairs. Illustrated in Figure II and quantified in Table IV-A, we compared the embeddings produced by PCA, FastMVU [18], Isomap [19] and our neighborhood denoising technique combined with MDS and FastMVU. Visually, it can be seen that PCA is simply an affine transform of the data, due to embedding with all edges both valid and noisy. As a result, unable to preserve the consistency of the data depth of the Swiss Roll is completely lost in 2D. The noisy edges also present a problem for Isomap in that consistency is lost in a similar manner as PCA. In addition, Isomap brings into closer proximity points at the edges of the manifold. This circumstance is worse than the PCA result because it gives the appearance that the edges of the manifold have continuity in the input space, when in fact they do not. Our denoising procedure was able to detect these three noisy edges and produce the proper embedding of the Swiss Roll.

From our informal experience, FastMVU is the best of the non-denoising embedding techniques. In this case, however, we were unable to produce quality results for both the original and denoised neighborhood graphs. Although these embeddings are themselves quite noisy, we anticipate in the long-run that denoising with FastMVU will yield the best embedding results.

B. Interactive Control of a Robot Hand

Our sparse control and subspace embedding systems were implemented in Matlab. Mex executables formed the bridge between our Matlab implementation and the C++ interface provided by DLR for the control of the robot hand. The robot hand used in the experiments was the DLR/HIT

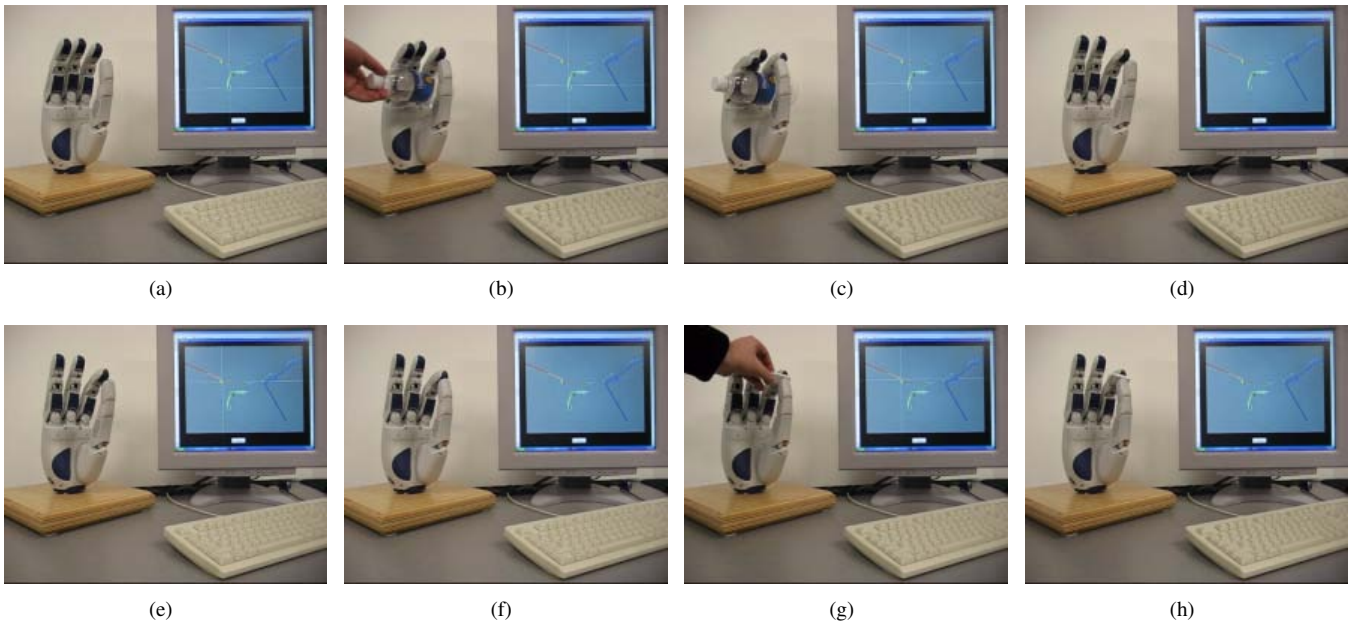


Fig. 4. Performances of interactive sparse control for power grasping on a water bottle (top, a-d) and precision grasping of a small eraser (bottom, e-h).

Embedding Error	2D
PCA	1.2580×10^{12}
FastMVU	2.7928×10^{12}
Isomap	8.7271×10^{11}
BP - FastMVU	2.1027×10^{12}
BP - MDS	$\mathbf{1.2099 \times 10^{8}}$

TABLE I

ERROR BETWEEN EUCLIDEAN DISTANCES IN THE NOISY SWISS ROLL EMBEDDINGS AND GROUND TRUTH DISTANCES.

anthropomorphic robot hand, constructed with 4 fingers, 17 DOFs (with 4 redundant DOFs). This hand has a form factor of roughly 1.5 times the size of a human hand. The human hand motion sequence that was used for training was a concatenation of finger tapping motions (once with each finger), 2 power grasps, and 3 precision grasps (one with each finger) captured by a Vicon optical motion capture system. The performer's hand was instrumented with 25 reflective markers, approximately 0.5cm in width, as shown in Figure 2. These markers were placed at critical locations on the top of the hand: 4 markers for each digit, 2 for the base of the hand, and 3 on the forearm. The resulting dataset consisted of approximately 500 frames and intentionally selected to have at most one missing (occluded) marker at any instant of time. Each frame of hand motion is considered a point in a high-dimensional pose space. The pose space is defined as the 3D endpoints of the fingers in the hand's local coordinate system, resulting in a 12-dimensional vector. Because the DLR hand has only 3 fingers and a thumb, data for the fifth human finger (pinky) is omitted. Joint angles used for motion control of the hand were computed using an inverse kinematics procedure that minimized the distance between each finger's endpoint position with respect to the knuckle of the finger.

The 2D space of embeddings is produced by running our neighborhood denoising technique in combination with Isomap on a neighborhood defined from the high-dimensional input poses. Neighborhood graphs are constructed by finding the eight nearest neighbors to each data point (k = 8). While the user is moving on the 2D space depicted on the screen, the high-dimensional robot configuration that each 2D point corresponds to is applied to the robot hand. The desired configuration of the hand is determined by the nearest neighboring point in the 2D embedding with respect to the current mouse position. Within these embeddings, continuity and consistency of the original poses are preserved (from manual inspection), placing the different types of grasps in different areas of space and facilitating user control.

We performed two interactive tasks using our sparse control system: power grasp of a water bottle and precision grasp of an eraser between the thumb and index fingers. Figures 4(a-d) show the different phases of the power grasp before, during and after performing the actual grasp. The phases of the precision grasp are illustrated in figures 2(e-h). As shown these figures and accompanying video, control of the robot hand is performed in a reasonably consistent and "accessible" manner. The objects are successfully grasped and postures of the robot hand resemble the postures of a human hand during grasping. In terms of the quality of the embeddings space, 2D trajectories that correspond to different motions as well as 2D points that correspond to frames of the same motion have got only a small overlap. In this way, the performance of a single desired motion becomes much easier and more effective.

Despite developing our neighborhood denoising procedure, neighborhood graphs for our grasping trials ended up to not have noisy edges, and, thus, denoising was not necessary.

Subspace embeddings produced by BP-Isomap and Isomap were identical. Careful selection of motion capture data ended up being the largest facilitator of generating suitable control subspaces. However, the noise-free character of this data came at the cost of diversity in the set of hand motion. The motion trials used by Jenkins [13] consisted of 3000 frames and contained much more variety in the style of grasps, although with upto eight missing markers at any point in time. In our future work, we plan to further experiment and extend graph denoising to handle such diverse, but unruly, datasets.

V. Conclusion

In this paper, we have attempted addressed the problem of sparse control of a high-DOF robot hand. Considering the problems of noise in pose graph construction and motion capture, we introduced a method for denoising neighborhood graphs for embedding hand motion into 2D spaces. Such spaces allow for control of high-DOF systems using 2D interfaces such as cursor control via mouse or decoding of neural activity. Preliminary results were presented demonstrating our approach to interactive sparse control for successful power grasping and precision grasping using a 13 DOF robot hand.

References

[1] J. del R. Millan, F. Renkens, J. Mourino, and W. Gerstner, "Brain-actuated interaction," *Artif. Intell.*, vol. 159, no. 1-2, pp. 241–259, 2004.

[2] M. Zecca, S. Micera, M. C. Carrozza, and P. Dario, "Control of multifunctional prosthetic hands by processing the electromyographic signal," *Critical Reviews in Biomedical Engineering*, vol. 30, no. 4-6, pp. 459–485, 2002.

[3] R. R. e. a. B. Crawford, "Real-time classification of electromyographic signals for robotic control."

[4] P. v. d. S. S. Bitzer, "Learning emg control of a robotic hand: towards active prostheses," in *IEEE International Conference on Robotics and Automation*, 2006.

[5] J. Donoghue, A. Nurmikko, G. Friehs, and M. Black, "Development of neural motor prostheses for humans," *Advances in Clinical Neurophysiology (Supplements to Clinical Neurophysiology)*, vol. 57, 2004.

[6] D. Taylor, S. H. Tillery, and A. Schwartz, "Information conveyed through brain control: Cursor versus robot," *IEEE Trans. Neural Systems Rehab Eng.*, vol. 11, no. 2, pp. 195–199, 2003.

[7] M. Serruya, A. Caplan, M. Saleh, D. Morris, and J. Donoghue, "The braingate pilot trial: Building and testing a novel direct neural output for patients with severe motor impairment," in *Soc. for Neurosci. Abstr.*, 2004.

[8] L. R. et al., "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, pp. 164–171, 2006.

[9] M. V. E. Crawford, "Learning to select negotiation strategies in multi-agent meeting scheduling," in *Working Notes of the AAAI Workshop on Multiagent Learning*, 2005.

[10] J. Lin, Y. Wu, and T. Huang, "Modeling the constraints of human hand motion," in *IEEE Workshop on Human Motion*, 2000.

[11] E. Todorov and Z. Ghahramani, "Analysis of the synergies underlying complex hand manipulation," in *Intl. Conference of the IEEE Engineering in Medicine and Biology Society*, 2004.

[12] C. R. Mason, J. E. Gomez, and T. J. Ebner, "Hand synergies during reach-to-grasp," *The Journal of Neurophysiology*, vol. 86, no. 6, pp. 2896–2910, Dec 2001.

[13] O. C. Jenkins, "2D subspaces for sparse control of high-dof robots," in *Intl. Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2006)*, New York, NY, USA, Aug-Sep 2006. [Online]. Available:

[14] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," pp. 239–269, 2003.

[15] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet, "Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.

[16] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[17] T. Cox and M. Cox, *Multidimensional Scaling*. London: Chapman and Hall, 1994.

[18] K. Weinberger, F. Sha, Q. Zhu, and L. Saul, "Graph laplacian methods for large-scale semidefinite programming, with an application to sensor localization," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hofmann, Eds. Cambridge, MA: MIT Press, 2007.

[19] V. d. S. Joshua B. Tenenbaum and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," pp. 2319–2323, 2000.