Using Manifolds for Dextrous Hand Control

Jan Steffen, Robert Haschke and Helge Ritter
Neuroinformatics Group, Faculty of Technology, Bielefeld University, Germany
{jsteffen,rhaschke,helge}@techfak.uni-bielefeld.de

***Abstract*— In dextrous hand control, the implementation of manipulation movements still is a complex and intricate undertaking. Often, a lot of object physics and modelling effort has to be incorporated into a controller working only for a restricted task specification and performing quite artificially looking movements. In this paper, starting from our representation for dextrous grasping, the *Grasp Manifold*, we motivate and present adaptations which enable the new manifold – the *Manipulation Manifold* – to robustly represent manipulation movements which have been recorded from human demonstration. It facilitates later application and promotes natural motion. We use manifolds of hand postures embedded in the finger joint angle space which are constructed such that manipulation parameters including the advance in time are represented by distinct manifold dimensions. This allows for purposive navigation within such manifolds. We present the manifold construction using the Unsupervised Kernel Regression (UKR) and our first results of representing the turning movement of a bottle cap.**

I. Introduction

During the last decades, researchers and engineers have made huge advances in constructing and building anthropomorphic robot hands which have become more and more sophisticated as one can see for example in the Utah/MIT Hand [6], the DLR II Hand [2] and the Shadow Dextrous Hand [22]. Together with these developments, researchers are facing the question of how to dexterously control such complex robots with up to 20 degrees of freedom in up to five fingers and a wrist. It quickly became clear that implementing fixed grasp and manipulation programs does not lead to satisfying results as it is very time consuming on the one hand and not robust against or generalisable to differences in the grasping or manipulation situation. Thus, several sophisticated approaches have been presented to realise more robustness and generalisability.

In the domain of dextrous *grasping*, most of the published approaches focus on one of the two most obvious aspects of the grasping task - either the object geometry or the tactile impressions during the grasp. Concerning the geometry aspect, several approaches have been presented to incorporate explicit object geometry models to calculate (optimal) contact points and plan grasp postures to realise them [1, 11, 14]. On the other side, the grasping approaches involving tactile information do not plan beforehand, but close the fingers around the object solely based on the tactile feedback of the hand until stable object contact is detected [13, 21, 17].

In recent years, alternative ways of thinking about grasping have been presented that aim at finding less complex representations of the grasping problem. To this end, *eigengrasps* have been taken into account [3] or lower-dimensional manifolds embedded in the hand posture space are used as *experience* basis for a grasp strategy [20, 19] or in order to facilitate manual grasp control [23].

When it comes to dextrous *manipulation*, the task is more complex and the fundamental ideas of the approaches become more distinct and diverse. Michelman and Allen [10] implemented simple object translations and rotations with the Utah/MIT Hand and combined them to more complex tasks. In this manner, they achieved to remove a child-proof bottle top with two fingers exploiting a decomposition into subtasks and explicit force and position control schemes. Zhang et al. [26] define a graph of vertices representing *canonical grasps* consisting of topological hand/object feature pairs having contact when the associated grasp is achieved. Directed edges between two grasps represent possible transitions which have to be designed as lower-level control laws. Manipulation planning then is implemented as path planning in the graph between defined start and end vertices. Fuentes and Nelson [4] learn a mapping from *perceptual goals* – consisting of targeted object position/orientation and applied finger forces – onto robot commands realising these goals using an evolution strategy. Afterwards, manipulation can be performed by defining the task-specific perceptual goal and applying the learned mapping. Han et al. [5] propose a pure contact wrench analysis approach. They use a planner to generate a path in the space of *feasible configurations of the manipulation system* respecting hand/object constraints. A controller then incorporates sensor readings and system kinematics and statics to properly actuate the planned path. Platt et al. [15] address dextrous manipulation by sequencing concurrent combinations of hierarchical organised closed-loop controllers each derived from potential functions and realising force-related objectives. By dint of operating subordinated controllers in the nullspace of superiors, higher-level conditions like wrench closure can be prioritised and thus sustained.

To a certain extent, all these approaches require the manual design of (lower-level) controllers from scratch. In [18], Schaal argues that learning without incorporating prior knowledge is a mostly artificial approach rarely taken by humans and analyses the benefit of learning from demonstration. He applies reinforcement learning on balancing a pole with an anthropomorphic robot arm to find an optimal policy and solves the problem based on data from a 30 second demonstration. Nevertheless, he concludes that not every learning problem can profit from prior knowledge in the same the way.

Although these approaches all realise robust dextrous grasping or manipulations to a certain degree, their implementations require considerable effort in problem modelling on the level of task definition and object characteristics. In addition, to our knowledge, none of the mentioned grasping approaches could be generalised to the manipulation task, nor vise versa.

In this paper, starting with a review of our previous approach of representing grasp postures as manifolds in the hand joint angle space [20, 19], we propose a modification of this approach to represent manipulation movements as well. The main idea is to construct manifolds – again embedded in the finger joint angle space – which represent the subspace of hand postures associated with a specific manipulation movement. Instead of learning these representations in a purely unsupervised manner yielding unpredictable, ”undirected” manifolds, we want to construct them such that specific movement parameters – and especially the advance in time – are explicitly represented by specific and distinct manifold dimensions. For our initial experiments, we focus on the manipulation movement of turning a bottle cap incorporating the advance in time and the cap radius as manipulation parameters.

The paper is organised as follows: In Section II, we review the basic principle of our manifold representation for dextrous *grasping*. Section III addresses the differences in the representational requirements for grasp postures and manipulation movements, Section IV concerns the computational means that we chose for the implementation, namely the *Unsupervised Kernel Regression*. In Section V we describe our first, constructional approach to fulfilling the stated requirements followed by the results in Section VI. Finally, we end up with a conclusion and an outlook on future work in Section VII.

II. The Grasp Manifold

In our previous work [20, 19], we presented a new approach to dextrous robot grasping that combines the advantages of geometry-based and tactile-driven grasping employing an implicit representation of grasping experience using a self-organising map (SOM, cf. [20, 16]). The SOM lattice is trained with previously recorded hand postures which led to successful grasps. In this manner, it forms a discrete approximation of a smooth *Grasp Manifold* representing the subspace of the hand joint angle space described by the training data and thus by the set of known grasp postures. Using tactile information to infer implicit knowledge about the object position and shape, the algorithm dynamically exploits the SOM to adapt the grasping motion to the actual situation. According to observed finger contacts, the most suitable hand posture is selected from the grasp manifold represented by the SOM nodes' reference vectors. Fig. 1 visualises a 10x10 SOM-based Grasp Manifold. Each hand picture represents one of the adapted reference vectors of the SOM and thus one point on the Grasp Manifold. Notice that this representation of *grasp postures* consists only of hand postures in the final grasp positions in which the hand is in contact with the object.
For further details, please refer to [20, 19].

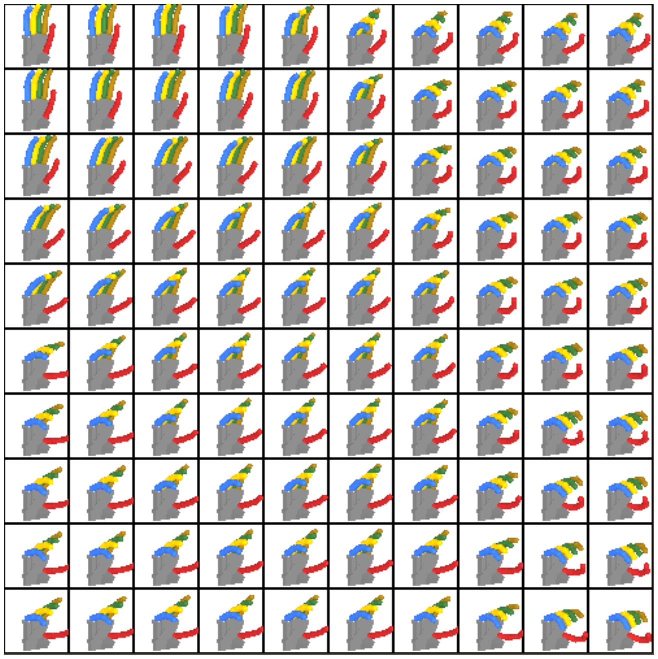


Fig. 1. A two-dimensional 10x10 SOM-based *discrete Grasp Manifold* trained with 4220 cylinder grasp postures. Each hand picture represents a reference vector of a single SOM-node and a point on the Grasp Manifold respectively. Represented are only hand postures in the final grasp position.

III. From grasping to manipulation

Grasping and manipulation are closely related. Nevertheless, the targeted aims usually are quite different: in the case of grasping, we want to fixate the object, whilst when thinking of manipulation, we rather aim at performing a certain movement with it. In terms of our manifold representation, this can be formulated as follows:

Grasping corresponds to ”pulling” the current hand posture onto the manifold such that the resulting posture is *one point* on it which causes a stable immobilisation of the object. Hence, the goal is one specific final hand posture.

Manipulation in contrast is not the search for only one specific *point on* the manifold, but rather for a whole *trajectory through* an adequate manifold. Thus, the goal is a sequence of several intermediate hand postures which result in the targeted manipulation when actuated sequentially. Notice, that in the initial phase, the hand posture needs to be pulled onto the manifold similar to the grasping process, but the resulting posture causes not necessarily an object immobilisation or not even object contacts.

The *Grasp Manifold* is a good starting point for the construction of the *Manipulation Manifolds*, but the conditions which have to be fulfilled need to be adapted to the new task. The basic ideas of the *Grasp Manifold* realised in [20] are:

a) every point on the manifold is a *grasp posture* being a hand posture that realises a grasp in combination with the corresponding object in the corresponding position.

b) the manifold spans over the whole targeted workspace or over all targeted grasping contexts, respectively. Obviously, the grasp algorithm would not be able to perform well in all targeted contexts otherwise.

c) the manifold representation needs to support the *associative completion* mechanism [20, 24] which enables the projection of partially specified data points onto the manifold resulting in the completed hand postures lying on the manifold. This allows for a dynamic incorporation of the tactile information.

d) the sample resolution in the represented manifold realised by prototypes or reference vectors need to be high enough. Nevertheless, a continuous manifold representation is not necessary because a generic closing algorithm is able to terminate the grasp starting from a hand posture near a final grasp posture.

In the case of *Manipulation Manifolds*, we modified these conditions in order to fit to the new task:

a) To represent the pure manipulation phases – thus only those moments in which the hand is in direct interaction through contacts with the object – it is again necessary that every point corresponds to a contact situation. However, in our approach, we aim at representing whole manipulation movements possibly containing non-contact phases. Hence, we can soften the previous requirement to: the resulting manifold consists only of points corresponding to hand postures generated by the manipulation movement to be represented. Notice that the *grasp* postures demanded by the *Grasp Manifold* conditions are comprised herein. Additionally, remark that only hand postures representing contact situations (not necessarily grasp postures) implicitly inhere object information and thus, only these postures can be used later on to infer object-specific parametrisation of the manipulation movement.

b) For manipulation, it is still necessary to represent the whole subspace of the aimed manipulation movement to enable an algorithm to reproduce it.

c) In the grasping case, the associative completion is used to ”pull” the current hand posture onto the manifold. Basically, this is not needed to perform the manipulation movement by navigating through the manifold. Nevertheless, to find a good starting point, it is a good approach to perform an initial ”pulling” onto the manifold similar to grasping. Hence, we aim at manifold representations again allowing for associative completion.

d) As we do not target only one specific point but need to navigate through the manifold following a trajectory where every intermediate hand posture lies on the manifold, a discrete approximation like the SOM lacks the necessary precision. Instead, we use a continuous manifold representation, namely the *Unsupervised Kernel Regression* (cf. next section).

Notice that – if we guarantee condition c) and specially mark the regions of the manifold described by hand postures in the training data that effect object contacts – the *Grasp Manifold* is a subset of the *Manipulation Manifold* and we can combine grasping and the reproduction of manipulation movements in one representation. In this combination, the initial grasping can be seen as object-specific parametrisation of the subsequent manipulation movement.

IV. Unsupervised Kernel Regression

Unsupervised Kernel Regression (UKR) is a recent approach for learning continuous manifold representations. It has been introduced as an unsupervised formulation of the Nadaraya-Watson kernel regression estimator by Meinecke, Klanke et al. in [9] and further developed by Klanke in [8, 7]. It uses the Nadaraya-Watson estimator [12, 25] to find a lower-dimensional (latent) representation of the original data and a smooth mapping from that latent space back to the original data space at the same time. The original Nadaraya-Watson estimator defines a mapping

$$\vec{y} = f(\vec{x}) = \sum_i \vec{y}_i \frac{K(\vec{x} - \vec{x}_i)}{\sum_j K(\vec{x} - \vec{x}_j)} \tag{1}$$

which realises a smooth, continuous generalisation of the functional relationship between $\vec{x}$ and $\vec{y}$ described by given data $(\vec{x}_i; \vec{y}_i)$. Here, $K(\cdot)$ is a density kernel. UKR now treats Eq. (1) as a mapping from a lower dimensional latent space $\mathcal{X}$ to the data space $\mathcal{Y}$ which is described by a set of observed data $Y = \{\vec{y}_i\}$. Here, the corresponding $X = \{\vec{x}_i\}$ play the role of *latent parameters* of the regression function:

$$\vec{y} = f(\vec{x}; X) = \sum_i \vec{y}_i \frac{K(\vec{x} - \vec{x}_i)}{\sum_j K(\vec{x} - \vec{x}_j)}. \tag{2}$$

The training of the UKR manifold thus can be realised by gradient-based minimisation of the reconstruction error

$$R(X) = \frac{1}{N} \sum_m \parallel \vec{y}_m - f(\vec{x}_m; X) \parallel^2 . \tag{3}$$

As special benefit, the UKR efficiently performs Leave-K-Out Cross-Validation by using a modified $f(\vec{x}_m; X)$ in Eq. (3):

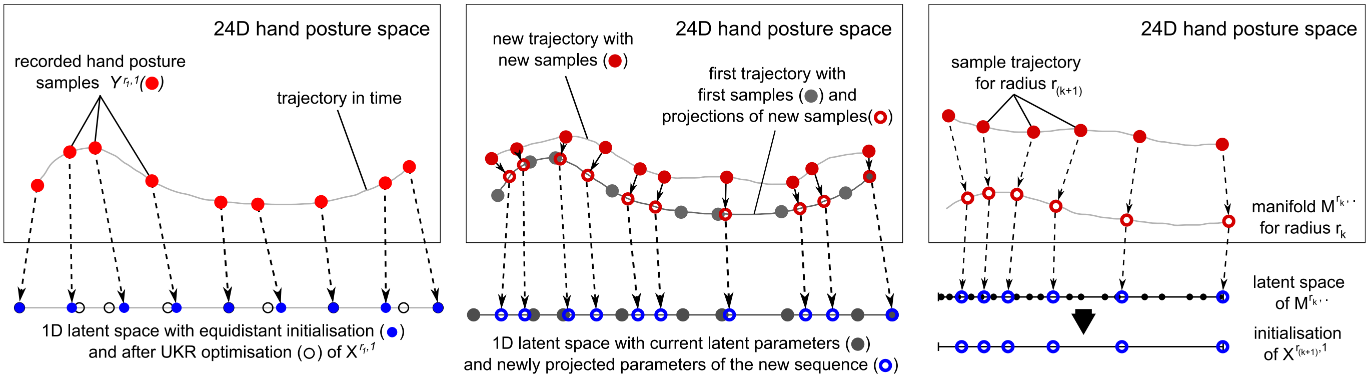
$$f_m(\vec{x}; X) = \sum_{i \notin \mathcal{N}_m} \vec{y}_i \frac{K(\vec{x} - \vec{x}_i)}{\sum_{j \notin \mathcal{N}_m} K(\vec{x} - \vec{x}_j)} \tag{4}$$

where $\mathcal{N}_m$ are the neighbours excluded for reconstructing $\vec{y}_m$.

For further details, please refer to [9, 8, 7].

V. Manifold Construction

The problem using unsupervised manifold learning methods often is that there are usually only limited means of incorporating partial task knowledge or of controlling the way of how ”the manifold is laid into the underlying data” respectively. On the other side, we can not provide completely specified training data that enable us to perform a purely supervised approach. Thus, our goal was to develop a mechanism that renders us possible to learn a manifold representation of the data in a partly unsupervised manner and additionally imprint specific meanings into the directions within the manifold. In terms of our scenario, the representation shall provide that every point on the manifold corresponds to one moment in time of a motion trajectory and additionally that the directions within the manifold, and especially its single dimensions, inhere the meaning of one specific motion parameter. Thus, in the example of turning a bottle cap, the goal is to realise a manifold in which one dimension controls the progress in


24D hand posture space
recorded hand posture samples $Y^{r_1,1}$(●)
trajectory in time
1D latent space with equidistant initialisation (●) and after UKR optimisation (○) of $X^{r_1,1}$
24D hand posture space
new trajectory with new samples (●)
first trajectory with first samples (●) and projections of new samples(○)
1D latent space with current latent parameters (●) and newly projected parameters of the new sequence (○)
24D hand posture space
sample trajectory for radius $r_{(k+1)}$
manifold $M^{r_k,\cdot}$ for radius r_k
latent space of $M^{r_k,\cdot}$
initialisation of $X^{r_{(k+1)},1}$

(a) Equidistant initialisation of the latent parameters (blue) respecting the order of the hand posture samples in the movement trajectory (red). The black circles in latent space depict possible displacements of the latent parameters due to the UKR optimisation.

(b) Procedure of incrementally adding new sequences by projecting new samples into the latent space of the previously trained manifold. The new manifold consisting of the combination of old and new data samples and latent parameters gets retrained afterwards.

(c) Synchronising different radii r_k and $r_{(k+1)}$ by projecting new samples corresponding to $r_{(k+1)}$ into the latent space $X^{r_k,\cdot}$ corresponding to r_k. The projections then are used as initialisation of $X^{r_{(k+1)},1}$ for the subsequent training of $\mathcal{M}^{r_{(k+1)},1}$.

Fig. 2. Schematic description of different steps in the manifold construction process.

time of the turning movement and another dimension specifies the radius of the cap. Then, performing the movement reduces to modifying the time component of the latent parameter.

A simple but - as presented in the following - effective approach to achieving this is to construct the final manifold out of several sub-manifolds each realising a manipulation movement of one motion parameter set.

In this first approach, we incorporate two parameters, the time and the cap radius. As training data, we recorded sequences of hand postures during cap turning movements for five different cap radii ($r = 1.5cm,\ 2.0cm,\ 2.5cm,\ 3.0cm$ and $3.5cm$). For each radius, we produced five sequences each of about 30 to 45 hand postures. Each hand posture consists of a vector of the 24 joint angles corresponding to our anthropomorphic robotic *Shadow Dextrous Hand* [22]. The construction of the final manifold is performed iteratively starting with training sequences corresponding to the minimal cap radius successively increasing the radius of the subsequent sequences. For the first sequence of hand postures $Y^{r_1,1} = \{\vec{y}_i^{\,r_1,1}\}$ corresponding to the minimal cap radius r_1 , we manually distribute the latent parameters of a 1D-UKR manifold equidistantly in a predefined interval of the latent space according to the intra-sequence order of the hand postures and perform the UKR training to optimise the latent parameters $X^{r_1,1} = \{\vec{x}_i^{\,r_1,1}\}$ afterwards (cf. Fig.2a). We denote the resulting UKR manifold as $\mathcal{M}^{r_1,1}$. The incorporation of the second sequence (corresponding to the same radius) is performed in an iterative manner: the hand posture vectors $Y^{r_1,2}$ of this sequence are projected pointwise into the latent space of the previously trained 1D-manifold $\mathcal{M}^{r_1,1}$ resulting in $X^{r_1,2}$ (cf. Fig.2b). By dint of this projection, we approximate a synchronisation of the temporal advance of the two movements. In the next step, we combine these data to a new manifold $\mathcal{M}^{r_1,\{1,2\}}$ with observed data $Y^{r_1,\{1,2\}} = Y^{r_1,1} \cup Y^{r_1,2}$ and latent parameters $X^{r_1,\{1,2\}} = X^{r_1,1} \cup X^{r_1,2}$. A subsequent UKR training optimises the latent parameters subject to the whole combined data set.

By performing this procedure for all sequences of hand postures corresponding to similar cap turning movements for one specific cap radius, a 1D–UKR is trained representing a generalised radius-specific movement. Thus, by applying this method to all sets of radius-specific sequences, we generate one 1D–UKR per radius. To promote the synchronisation of the temporal advances also between the different radius-specific manifolds, we only initialise the first manifold $\mathcal{M}^{r_1,1}$ with equidistant latent parameters as describes above. When proceeding with sequences $Y^{r_{(k+1)},1}$ of a new radius $r_{(k+1)}$, we project the sequence onto the previously trained manifold $\mathcal{M}^{r_k,\{1,..,n\}}$ (as with sequences for the same radius) and utilise the resulting latent parameters as initialisation $X^{r_{(k+1)},1}$ of the new manifold $\mathcal{M}^{r_{(k+1)},1}$ instead of combining them to a manifold $\mathcal{M}^{r_k,\{1,..,n+1\}}$ (cf. Fig.2c). The training then continues as described above. Notice that each first sequence used to initially train a new 1D-manifold plays a special role and determines the relevant subspace in the hand posture space. Therefore, it is important that these first sequences represent complete movements rather than only specific phases.

The subsequent combination of all 1D-manifolds $\mathcal{M}^{r_i,\cdot}$ to one 2D-manifold $\mathcal{M}$ representing the complete movements for all radii r_i covered by the training data then is performed manually and without UKR training. $\mathcal{M}$ then consists of all incorporated training data $\{Y^{r_i,j}\}_{i,j}$ together with the corresponding latent parameters $\{X^{r_i,j}\}_{i,j}$ and represents the whole manipulation movement described by the training data. Therefore, we denote it as *Manipulation Manifold*. The extension to two dimensions is realised by expanding each latent parameter $\vec{x}_i$ by a second dimension denoting the appropriate radius corresponding to the associated training sequence.

VI. Results

We applied the method described in Section V to all recorded training sequences. After having trained one 1D-manifold for each of the training radii $r = 1.5cm,\ 2.0cm,\ 2.5cm,\ 3.0cm$ and $3.5cm$ in the described synchronised manner, we added the corresponding radius values as second dimension to the latent parameters. The distribution of the latent parameters in the new latent space is depicted in Fig. 3. As constructed, the horizontal dimension represents the temporal advance within the cap turning movement and the vertical dimension denotes the associated cap radius. As no further UKR training is performed, the latent parameters only lie on the previously set discrete radius values.

To get a more distinct impression of the movements represented by the manifold and its generalisation abilities, Fig. 4 depicts a matrix of hand postures corresponding to the positions in a regular grid covering the latent space of the manifold. Again, the temporal advance is depicted in the horizontal and the radii in the vertical direction. To facilitate the comparison, a bottle cap $(r = 1.5cm)$ is depicted in each sub-figure. As shown in Fig. 3, only the radii $r = 1.5cm,\ 2.0cm,\ 2.5cm,\ 3.0cm$ and $3.5cm$ are directly supported by training data. Thus, the depicted intermediate radii $r = 1.75cm,\ 2.25cm,\ 2.75cm$ and $3.25cm$ in Fig. 4 visualise the generalisation ability of the constructed manifold to new cap radii. The corresponding movements for the intermediate radii are clearly similar to the training sequences. Secondly, it illustrates the effect of the temporal synchronisation between the different 1D-manifolds by projecting new sequences into the latent space of the previously trained manifolds before newly training as described in Section V. The most distinct picture of this synchronisation can be seen in columns 3-5 where the fingers are shown in the moments (virtually) contacting the cap (or in the video referenced in Fig.4). Additionally, those columns give an impression of the smoothness in the $2nd$ manifold dimension. Remark the smooth finger opening with increasing cap radius in the column direction. In the row direction, all rows depict smooth transitions from left to right indicating a smooth manifold also in the row direction.

One effect of the presented training method is not directly obvious in Fig. 4. When reaching the manifold border in the temporal dimension while performing the turning movement, the temporal position has to be reset to the beginning (go back from 100% to 0%) to restart the turning movement. As by now, there is no regulation for border synchronisation incorporated in the training, the 100%- and 0%-postures usually significantly differ from each other yielding an abrupt non-smooth hand movement when jumping back to the 0%-posture. As the beginning and the end of the movement are the phases where the fingers are the farthest away from the cap, the motion artifact resulting from the missing border synchronisation does not effect the cap turning and thus is not of particular relevance for the success of the cap turning manipulation. Nevertheless, we will address this issue in future work to optimise the natural impression of the manipulation.

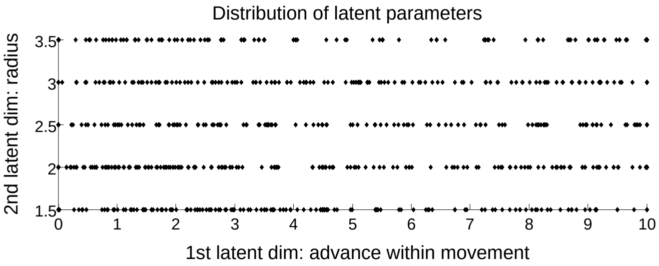


Fig. 3. Distribution of the latent parameters of $\mathcal{M}$.

VII. Conclusion

We presented the first steps towards a new approach in dextrous manipulation that uses sequences of hand postures recorded from human demonstration to learn and construct *Manipulation Manifolds* in which distinct dimensions represent distinct parameters of the associated manipulation. With the example of turning a bottle cap, we provided a proof of concept by incorporating two parameters in a manifold. From our experiments, we conclude several subjects and aims for our future work. The most important for us will be to change the learning and construction mechanism such that it works in a more unsupervised fashion with the goal of completely replacing the manual construction part by an unsupervised learning. For this, we have several ideas in mind of how to modify the UKR learning to better fit to the problem of chronologically ordered data sequences. Other important objectives are to explicitly incorporate contact conditions and to remove artifacts due to the missing border synchronisation. While following our goals, we want to sustain our main principle of this work of constructing manifolds in which distinct dimensions have imprinted distinct and specific meanings like the radius of a bottle cap and the temporal advance within the manipulation movement.

Acknowledgement This work has been conducted within the German Collaborative Research Centre "SFB 673 - Alignment in Communication" granted by the "Deutsche Forschungsgemeinschaft".

References

[1] C. Borst, M. Fischer, and G. Hirzinger. Calculating hand configurations for precision and pinch grasps. In *Proc. IROS*, 2002.

[2] J. Butterfass, M. Grebenstein, H. Liu, and G. Hirzinger. DLR-Hand II: next generation of a dextrous robot hand. In *Proc. ICRA*, 1986.

[3] M. Ciocarlie, C. Goldfeder, and P. Allen. Dimensionality reduction for hand-independent dexterous robotc grasping. In *Proc. IROS*, 2007.

[4] O. Fuentes and R. Nelson. Learning Dextrous Manipulation Strategies for Multifingered Robot Hands Using the Evolution Strategy. *Machine Learning*, 31, 1998.

[5] L. Han, Z. Li, J. Trinkle, Z. Qin, and S. Jiang. The planning and control of robot dextrous manipulation. In *Proc. ICRA*, 2000.

[6] S. Jacobsen, E. Iversen, D. Knutti, R. Johnson, and K. Biggers. Design of the Utah/M.I.T. Dextrous Hand. In *Proc. ICRA*, 1986.

[7] S. Klanke. *Learning Manifolds with the Parametrized Self-Organizing Map and Unsupervised Kernel Regression.* PhD thesis, Bielefeld University, 2007.

[8] S. Klanke and H. Ritter. A Leave-K-Out Cross-Validation Scheme for Unsupervised Kernel Regression. In *ICANN 2006*, volume 4132 of *LCNS.* Springer, 2006.

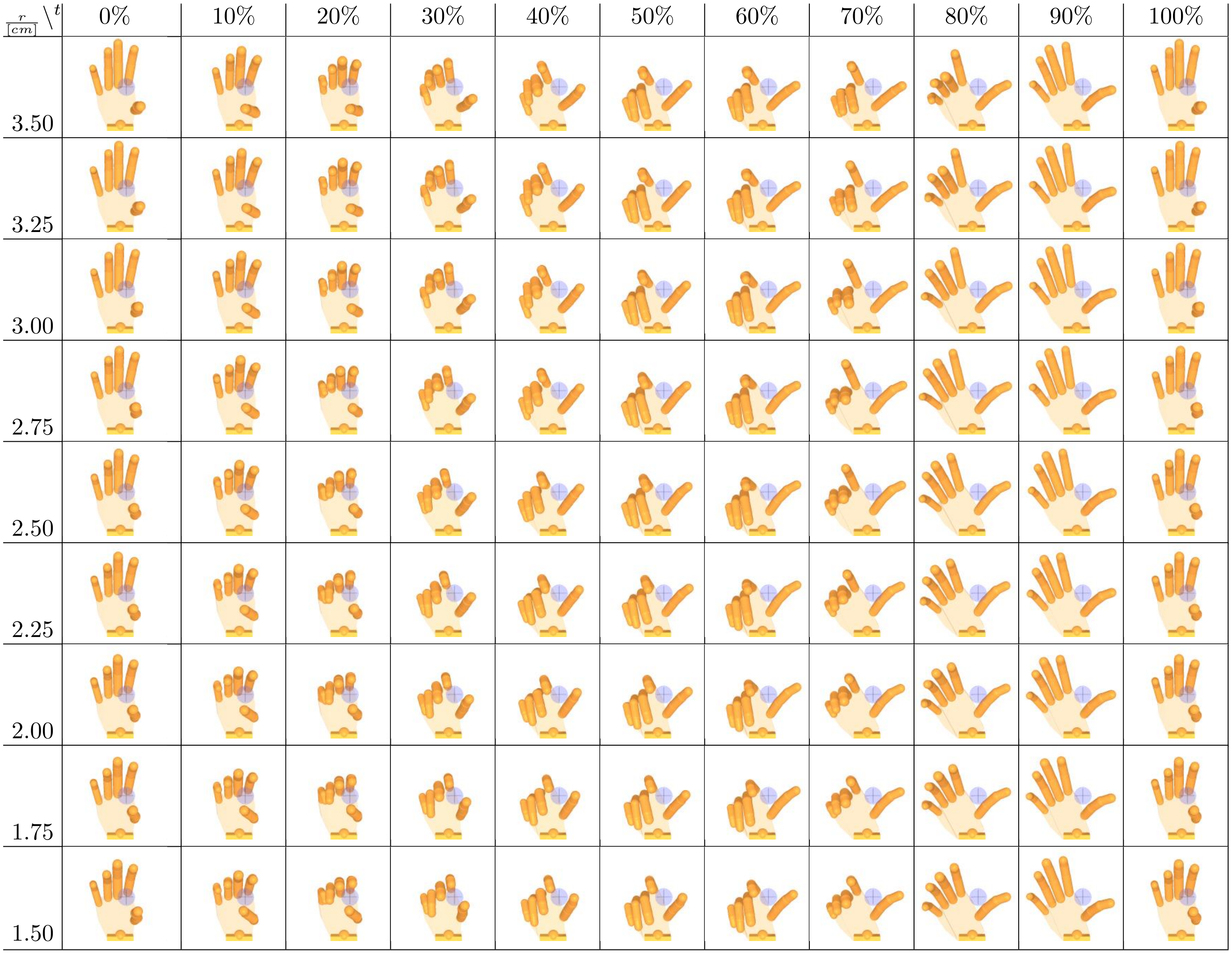


Fig. 4. Generalisation results of the UKR training/construction. The depicted hand postures correspond to the positions of a regular grid covering the latent space of the manifold. The horizontal direction is associated with the $1st$ (temporal) latent dimension and the vertical direction with the $2nd$ (radius) latent dimension. The size of the depicted bottle cap in all pictures is the same ($r = 1.5$) as a comparison aid. The radii $r = 1.5,\ 2.0,\ 2.5,\ 3.0$ and 3.5 correspond to radii covered by the training data, the radii $r = 1.75,\ 2.25,\ 2.75$ and 3.25 demonstrate the generalisation capability of the approach. A more sophisticated impression of the manifold can be received by means of a movie available under `http://www.techfak.uni-bielefeld.de/˜jsteffen/mov/`.

[9] P. Meinicke, S. Klanke, R. Memisevic, and H. Ritter. Principal Surfaces from Unsupervised Kernel Regression. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(9), 2005.

[10] P. Michelman and P. Allen. Forming Complex Dextrous Manipulations from Task Primitives. In *Proc. ICRA*, 1994.

[11] A. Miller, S. Knoop, P. Allen, and H. Christensen. Automatic grasp planning using shape primitives. In *Proc. ICRA*, 2003.

[12] E. A. Nadaraya. On Estimating Regression. *Theory of Probability and Its Application, Vol.9*, 1964.

[13] L. Natale and E. Torres-Jara. A sensitive approach to grasping. In *6th Intl. Conf. on Epigenetic Robotics, Paris, France*, 2006.

[14] R. Pelossof, A. Miller, P. Allen, and T. Jebara. An SVM Learning Approach to Robotic Grasping. In *Proc. ICRA*, 2004.

[15] R. Platt Jr., A. Fagg, and R. Grupen. Manipulation gaits: sequences of grasp control tasks. In *Proc. ICRA*, 2004.

[16] H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-Organizing Maps*. Addison-Wesley, 1992.

[17] F. Röthling, R. Haschke, J. J. Steil, and H. Ritter. Platform Portable Anthropomorphic Grasping with the Bielefeld 20-DOF Shadow and 9-DOF TUM Hand, 2007.

[18] S. Schaal. Learning from Demonstration. In *Advances in Neural Information Processing Systems*, volume 9. MIT Press, 1997.

[19] J. Steffen, R. Haschke, and H. Ritter. Experience-based and Tactile-driven Dynamic Grasp Control. In *Proc. IROS*, 2007.

[20] J. Steffen, R. Haschke, and H. Ritter. SOM-based experience representation for Dextrous Grasping. In *Proc. WSOM*, 2007.

[21] J. Steil, F. Röthling, R. Haschke, and H. Ritter. Situated robot learning for multi-modal instruction and imitation of grasping. *Robotics and Autonomous Systems*, Special Issue on ”Robot Learning by Demonstration”(47), 2004.

[22] The Shadow Robot Company. Shadow Dexterous Hand. www.shadowrobot.com/hand, 2002.

[23] A. Tsoli and O. Jenkins. 2d subspaces for user-driven robot grasping. In *RSS Workshop on Robot Manipulation: Sensing and Adapting to the Real World*, Atlanta, GA, June 2007.

[24] J. Walter. *Rapid Learning in Robotics*. PhD thesis, Faculty of Technology, Bielefeld University, 1996.

[25] G. S. Watson. Smooth Regression Analysis. *Sankhya, Series A*, 26, 1964.

[26] H. Zhang, K. Tanie, and H. Maekawa. Dextrous manipulation planning by grasp transformation. In *Proc. ICRA*, 1996.