

Using Structured UKR Manifolds for Motion Classification and Segmentation

Jan Steffen, Michael Pardowitz and Helge Ritter

Neuroinformatics Group, Faculty of Technology, University of Bielefeld, Germany

{jsteffen,mpardowi,helge}@techfak.uni-bielefeld.de

Abstract—Task learning from observations of non-expert human users will be a core feature of future cognitive robots. However, the problem of task segmentation has only received minor attention.

In this paper, we present a new approach to classifying and segmenting series of observations into a set of candidate motions. As basis for these candidates, we use Structured UKR manifolds, a modified version of Unsupervised Kernel Regression which has been introduced in order to easily reproduce and synthesise represented dextrous manipulation tasks. Together with the presented mechanism, it then realises a system that is able to reproduce and recognise the represented motions at the same time and can thus not only be used for the segmentation into the candidate motions, but also for their reproduction.

I. INTRODUCTION

Task learning from observations of non-expert human users will be a core feature of future cognitive robots. Although several systems for programming by demonstration or imitation learning have been proposed (see [18], [7] for overviews), the problem of task segmentation has only received minor attention. The decomposition of a task demonstration into its constituting subtasks was tackled only in problem specific ways and a general framework and methodology for task decomposition is still missing.

In this paper, we propose a novel approach to tackle the task decomposition problem that is based on the claim that the same representations of actions should be used for recognition and segmentation as for execution of the same action on a robot. This claim is well supported by neurophysiological findings, namely the mirror neuron theory [16]. This basically states that the same areas of the brain show activity during the own execution and the recognition of an action by another person. Other indications for holistic representations of task knowledge come from gestalt-based approaches [24], that propose unified representations for segmentation and action [15]. It seems natural to extend this line of thinking into the realm of task segmentation for technical systems, since robots using the same representations for recognition and execution could use their learning episodes and memory more efficiently, enabling them to avoid duplicates of knowledge.

This paper presents a method for the classification and segmentation of motion data exploiting structure features of the manifolds used to represent the candidate motions, namely Structured UKR manifolds [19], [20]. Whereas Structured UKR has been originally introduced as a manifold structure

for motion reproduction and synthesis, we here present features defined in the manifold domain which enable us to exploit its structure for the recognition of the same motion, yielding a system that realises reproduction and recognition on the basis of one and the same representation.

The paper is organised as follows: Section II will review some related work. In Section III and IV, we briefly review basic *Unsupervised Kernel Regression* and its modification to Structured UKR. Section V will address the training and test data and in Section VI, we present the new manifold features for classification and segmentation which then are evaluated in Section VII. We finish with a conclusion and an outlook on future work in Section VIII.

II. RELATED WORK

Robot task learning from human demonstration has drawn increasing attention during the past decade. Nonetheless, task segmentation has been tackled only implicitly by most of the presented systems. [1] applies hand-crafted rules to detect state transitions from video sequences. Segments are characterised through stable contact points between the objects recognized in the scene. More formalized models use Hidden-Markov-Models (HMMs) to segment walking or grasping actions from motion-capture data [4]. [5] performs unsupervised clustering using Vector Quantisation (VQ) to segment the basic actions (codes) for a discrete HMM. This method is refined in [6] to Gaussian Mixture Models where each Gaussian represents a single segment of a task demonstration. This GMM is then fed into a continuous HMM for sequence learning. A taxonomy of action primitives is presented in [8]. These primitives of action (mainly concerned with grasping) are learned in a supervised way in [26]. This allows to classify each frame of a task demonstration and to construct task segments from those classifications. These segments have been transformed into petri-nets for execution on a humanoid robot [25]. A similar way is proposed in [3] where a user demonstration is segmented based on the most likely primitives performed in each timestep. [14] applies a similar method using a winner-takes-all selection of the most probable behaviour to segment a sequence of navigation tasks. Several methods try to avoid the segmentation problem: [9] lets the user define the segmentation with explicit verbal commands that directly guide the robot through a demonstration. [2] and [21] do not decompose a task demonstration at all but search for direct mapping functions between input and output trajectories.

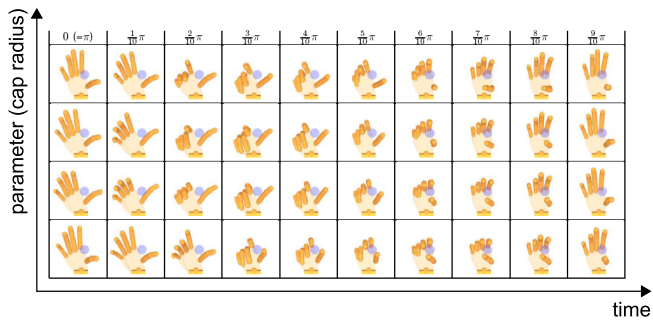


Fig. 1. Visualisation of a structured UKR latent space in the example of representing the turning movement of a bottle cap from [20]. The hand pictures correspond to the mappings of a regular grin in latent space back into the original input space of hand postures. The horizontal latent dimension corresponds to the time within the represented movement whereas the vertical latent dimension controls the motion parameter (the cap radius).

III. UNSUPERVISED KERNEL REGRESSION

Unsupervised Kernel Regression (UKR) is a recent approach to learning non-linear continuous manifold representations, that is, to finding a lower dimensional (latent) representation $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) \in \mathbb{R}^{q \times N}$ of a set of observed data $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N) \in \mathbb{R}^{d \times N}$ and a corresponding functional relationship $\mathbf{y} = \mathbf{f}(\mathbf{x})$. It has been introduced as the unsupervised counterpart of the Nadaraya-Watson kernel regression estimator by Meinecke et al. in [12]. Further development has led to the inclusion of general loss functions, a landmark variant, and the generalisation to local polynomial regression [11]. In its basic form, UKR uses the Nadaraya-Watson estimator [13], [23] as smooth mapping $\mathbf{f}: \mathbf{x} \in \mathbb{R}^q \rightarrow \mathbf{y} \in \mathbb{R}^d$ from latent to observed data space:

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^N \mathbf{y}_i \frac{K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i)}{\sum_j K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_j)}. \quad (1)$$

The original estimator realises a smooth, continuous generalisation of the functional relationship between two random variables \mathbf{x} and \mathbf{y} described by given data samples $(\mathbf{x}_i; \mathbf{y}_i)$. Here, $K_{\mathbf{H}}(\cdot)$ is a density kernel (e.g., Gaussian) with associated bandwidth parameters \mathbf{H} .

UKR now treats (1) as a mapping from the latent space to the original data space in which the manifold is embedded and from which the observed data samples $\mathbf{Y} = \{\mathbf{y}_i, i = 1..N\}$ are taken. The associated set $\mathbf{X} = \{\mathbf{x}_i, i = 1..N\}$ now plays the role of the input data to the regression function (1). Here, they are treated as *latent parameters* corresponding to \mathbf{Y} . As the scaling and positioning of the \mathbf{x}_i 's are free, the formerly crucial bandwidth parameter \mathbf{H} becomes irrelevant and we can use unit bandwidths. Thus, the regression function can be denoted as

$$b_i(\mathbf{x}; \mathbf{X}) = \frac{K(\mathbf{x} - \mathbf{x}_i)}{\sum_j K(\mathbf{x} - \mathbf{x}_j)} \quad (2)$$

$$\mathbf{f}(\mathbf{x}; \mathbf{X}) = \sum_{i=1}^N \mathbf{y}_i b_i(\mathbf{x}; \mathbf{X}) = \mathbf{Y} \mathbf{b}(\mathbf{x}; \mathbf{X}). \quad (3)$$

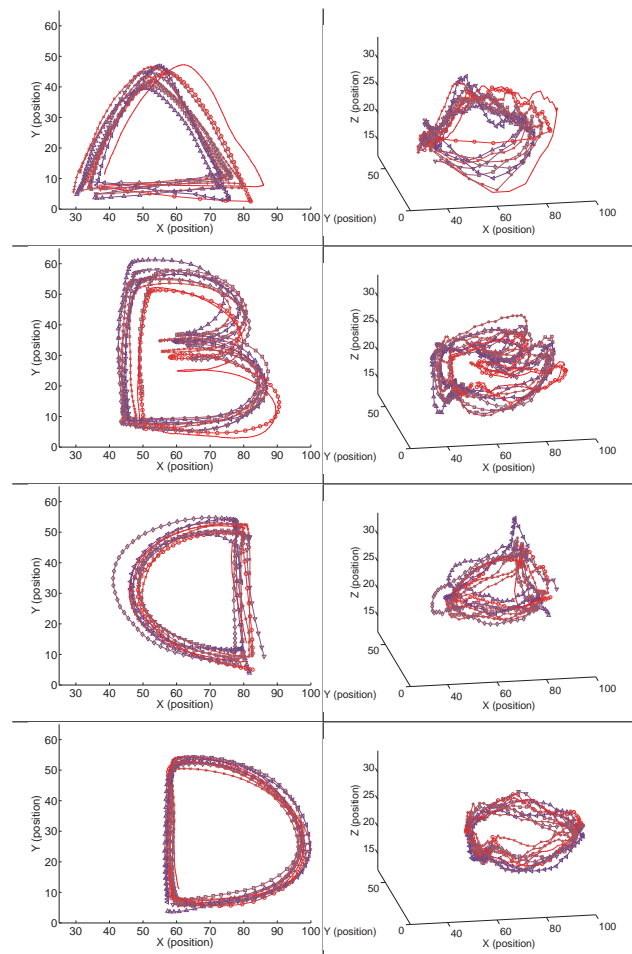


Fig. 2. The ABCD-data consist of recorded hand trajectories from drawing the letters 'A', 'B', 'C' and 'D' in the air. The 3D positions and 3D orientations are extracted from a monocular video camera pictures with the aid of the ARToolkit. The camera is placed on top of the scene (in Z direction) resulting in good position and orientation precision in X- and Y-direction and a less precise Z information (cp. X/Y projections in the left column and the X/Y/Z projections in the right column).

where $\mathbf{b}(\mathbf{x}; \mathbf{X}) = (b_1(\mathbf{x}; \mathbf{X}), b_2(\mathbf{x}; \mathbf{X}), \dots, b_N(\mathbf{x}; \mathbf{X}))^T \in \mathbb{R}^N$ is a vector of basis functions representing the effects of the kernels parametrised by the latent parameters.

As loss function for the UKR training, the reconstruction error is considered and can be denoted as

$$R(\mathbf{X}) = \frac{1}{N} \sum_i \|\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i; \mathbf{X})\|^2 = \frac{1}{N} \|\mathbf{Y} - \mathbf{Y} \mathbf{B}(\mathbf{X})\|_F^2. \quad (4)$$

Here, $\mathbf{B}(\mathbf{X}) = (\mathbf{b}(\mathbf{x}_1; \mathbf{X}), \mathbf{b}(\mathbf{x}_2; \mathbf{X}), \dots, \mathbf{b}(\mathbf{x}_N; \mathbf{X}))$ is an $N \times N$ *basis function matrix*. Note that moving the \mathbf{x}_i infinitely apart from each other results in $\mathbf{B}(\mathbf{X})$ being the identity matrix which corresponds to a trivial minimisation solution $R(\mathbf{X}) = 0$. In order to prevent this undesired case, several regularisation methods are possible [11]. Most notably, with UKR one can very efficiently perform leave-one-out cross-validation, that is, reconstruct each \mathbf{y}_i without using the \mathbf{y}_i term itself. To this end, the only additional step is to zero the diagonal of $\mathbf{B}(\mathbf{X})$ before normalising its column sums to 1. For a preselected density kernel, the highly non-linear reconstruction error (4) only depends on

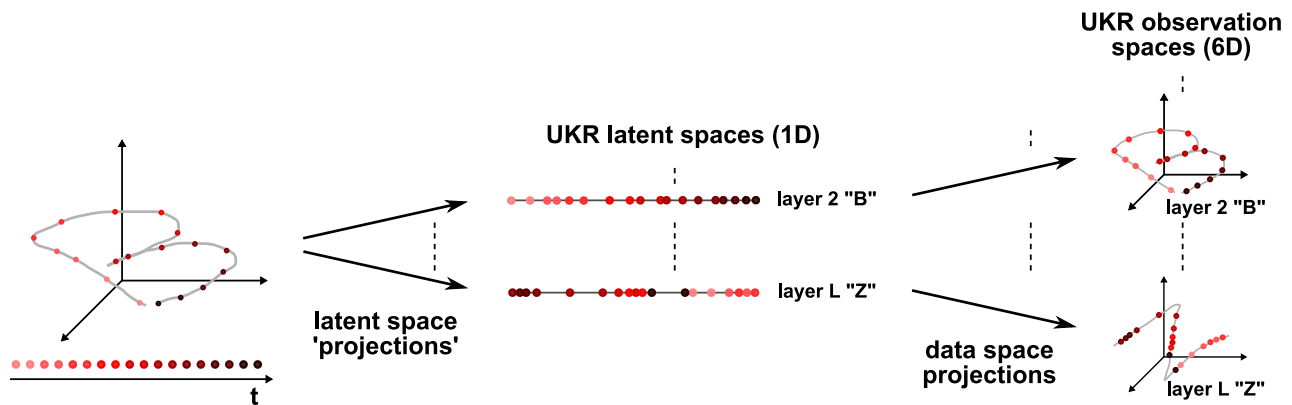


Fig. 3. Visualisation of the UKR feature basis for recognition. **[left]** The observed trajectory in 3D position space for the letter 'B' (gray) and the observed data samples (dots, coloured from red to black according to the temporal advance of the movement). **[centre]** The observed data samples are projected (by means of $\mathbf{g}(\cdot)$) into the latent spaces of different candidate UKR manifolds each representing a different letter. The order and arrangement of the projections of the chronologically ordered samples give information about how good the temporal aspect of the observed sequence is represented in the corresponding manifold. **[right]** The reprojections of the latent representations into data space (and the differences to their originally observed positions) can be used as measure for the quality of how good the single observation samples are represented by the manifold.

the set of latent parameters \mathbf{X} and thus can be optimised with respect to \mathbf{X} by gradient-based methods. As such often suffer from getting stuck in poor local minima, an appropriate initialisation is important. To this end, depending on the problem, PCA [10], Isomap [22] or LLE [17] are usually good choices. These eigenvector-based methods by themselves are quite powerful and efficient in uncovering low-dimensional structures in data sets. In contrast to UKR, however, PCA is restricted to linear structures and Isomap as well as LLE do not provide continuous mappings – a combination with UKR yields the best of both worlds.

An inverse mapping $\mathbf{x} = \mathbf{f}^{-1}(\mathbf{y}; \mathbf{X})$ from data space to latent space is not directly supported in UKR. Instead, one may use an orthogonal projection to define a mapping $\mathbf{x}^* = \mathbf{g}(\mathbf{y}; \mathbf{X}) = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{f}(\mathbf{x}; \mathbf{X})\|^2$ which approximates $\mathbf{f}^{-1}(\cdot)$. For further details, please refer to [12], [11].

IV. STRUCTURED UKR FOR MOTION DATA

In this original form, UKR is a purely unsupervised approach to continuous manifold learning.

However, in previous work, we applied the method on representing series of chronologically ordered data. To this end, we constructed [19] or trained [20] UKR latent spaces such that specific latent dimensions are associated with distinct task parameters. Here, especially the temporal order of the original data is captured by one latent dimension. Like this, we achieved to represent the movement of turning a bottle cap in a strongly structured latent space (cp. Fig. 1). Here, one (periodic) latent dimension corresponds to the temporal advance within the movement and the other (non-periodic) dimension is associated with the radius of the bottle cap.

Having enforced such strong structure in the latent representation, we are able to exploit the knowledge about it to easily reproduce and synthesise the represented movements. To this end, a simple controller has been presented in [20] which realises a grasping movement by following the radius

dimension in the direction of decreasing radii and then – after appropriate contacts have occurred – switches from grasping to the synthesis of the manipulation movement by following the temporal dimension.

V. THE 'ABCD-DATA'

For the following part of the paper, we leave the domain of the manipulation data that we used in previous work and concentrate on a simpler, but more descriptive and better to interpret data as it is better suited to explain the functionality of the following method and makes it easy for the reader to understand and comprehend the evaluation results later on.

The data – which is depicted in Fig. 2 – consists of several (position and orientation) trajectories of a hand that draws the letters 'A', 'B', 'C', and 'D' in the air. The position and orientation of each sample observation is extracted from the pictures of a monocular video camera which is placed on top of the scene, with orthogonal viewing direction onto the virtual 'drawing plane'. For the hand tracking, we utilised the ARToolkit¹ system.

VI. STRUCTURED UKR FOR RECOGNISING MOVEMENTS

In this Section, we present how the UKR manifold representations together with the enforced strong structure of the latent space described in the Sec. IV can be exploited – besides for the reproduction and synthesis of the represented movement – at the same time also for its recognition.

To this end, we introduce features in the manifold domain which express the similarity of the observed data and the data represented in the manifold. As we deal with motion data and thus time sequences of intermediate hand postures (or positions and orientations, as in the following example) of the observed motion, we utilise the *series* of orthogonal projections $\mathbf{g}(\cdot)$ and their reconstructions $\mathbf{f}(\mathbf{g}(\cdot))$ (cp. Sec. III) as the feature basis.

¹see <http://artoolkit.sourceforge.net>

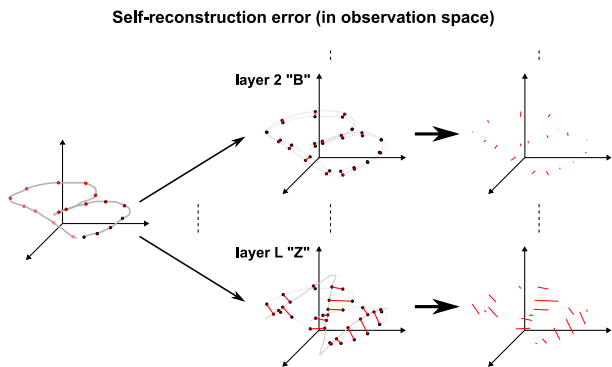


Fig. 4. Self-reconstruction scheme as visualisation of the self-reconstruction error in observation space (data and colours as in Fig. 3). The samples of the observed trajectory (left) are first projected into the manifold’s latent space and then reprojected into original observation space: $\mathbf{f}(\mathbf{g}(\cdot))$ (cp. Sec. III). According to the similarity or dissimilarity of observed and represented letter in the manifold, the self-reconstruction yields smaller or bigger errors, respectively. E.g., in the top row, the observed trajectory for the letter ‘B’ is projected onto a manifold representing the letter ‘B’ as well (visualised by the second gray line in the centre column) resulting in small reconstruction errors. In contrast, in the bottom row, the same observed ‘B’ is projected onto a manifold representing a ‘Z’ whose shape is basically different from the shape of the ‘B’. The reconstruction thus yields significantly higher errors (please compare the right most sub-figures, which visualise the position errors of the reconstructions for ‘B’ and ‘Z’, respectively).

Figure 3 visualises these two feature bases in the example of the letter trajectories presented in the last section. From this, we define two different features for the evaluation of the similarity and thus as measure for later recognition:

a) The compatibility of the single observations with the manifold which is basically expressed by the difference between the observation and its reconstruction. The samples of the observed trajectory are first projected into the manifold’s latent space and then reprojected into original observation space (cp. Sec. III). According to the similarity or dissimilarity of observed and represented letter in the manifold, the self-reconstruction yields smaller or bigger errors, respectively, as visualised in Fig. 4. As a measure for the reconstruction compatibility, we define

$$C_{rec}(\mathbf{y}^*; \mathbf{Y}) = -1 + 2 \cdot \exp(-\Delta^T \mathbf{V}^{-1} \Delta) \quad (5)$$

where $\Delta(\mathbf{y}^*) = \mathbf{y}^* - \mathbf{f}(\mathbf{g}(\mathbf{y}^*))$ is the self-reconstruction error of observation \mathbf{y}^* and $\mathbf{V}(\mathbf{Y})^{-1} = \text{diag}(\frac{1}{\text{var}(y_{\cdot,1})}, \dots, \frac{1}{\text{var}(y_{\cdot,d})})$ is used to weigh the different dimensions according to the corresponding variance in the manifolds training data set $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N) \in \mathbb{R}^{d \times N}$ (cp. Sec. III). In this formulation, $C_{rec}(\mathbf{y}^*; \mathbf{Y})$ can take values in the interval $[-1; +1]$ whereas -1 corresponds to \mathbf{y}^* being maximally incompatible and $+1$ to \mathbf{y}^* being maximally compatible with the underlying UKR manifold.

b) The compatibility of a single observation with its preceding observations (*history*). As Structured UKR in the described form (cp. Sec. IV or [19], [20]) enforces the propagation of the training sequence order to the latent representations, this compatibility measure can be expressed by the relative positions (or distances) of the latent representation of the current and preceding observations as shown

Temporal distances of latent representations (1D, periodic in $[0; \pi]$) of successive data points

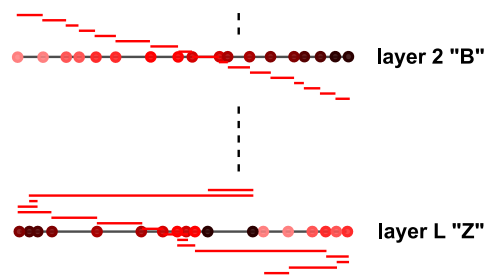


Fig. 5. Schematic visualisation of the UKR latent space feature for recognition (data and colours as in Fig. 3). Schematically depicted are latent space projections of an observed trajectory for the letter ‘B’ into the latent space of a manifold also representing a letter ‘B’ (top) or a letter ‘Z’ instead (bottom) (cp. Fig. 4). If observed and represented letter are similar and the order of latent representations equals the order of the observations – as in the top sub-figure – the distances between projections of successive data points are small and the sum over all distances is equal to the range between smallest and highest value of the latent *time* dimension. In contrast, if observed and represented trajectories are very dissimilar, the order does not correspond the order of the observed sequence and distances between successive projections can be very big. The sum over all distances then is usually significantly higher than the range between smallest and highest value in the latent time dimension.

in Fig. 5: if observed and represented sequence are similar and the order of latent representations equals the order of the observations, the distances between projections of successive data points are small (Fig. 5, top). In contrast, if observed and represented trajectories are very dissimilar, the order does not correspond to the order of the observed sequence and distances between successive projections can be very large (Fig. 5, bottom). As a measure for the compatibility of the observation \mathbf{y}_{t-h} in the history of \mathbf{y}_t , we define:

$$c_{hist}(h, t) = \frac{1}{2} \cos(\delta_{h,t}) + \frac{1}{2} C_{rec}(\mathbf{y}_{t-h}; \mathbf{Y}) \quad (6)$$

where $\delta_{h,t} = \text{mod}_{\pi}(\mathbf{g}(\mathbf{y}_{t-h-1}) - \mathbf{g}(\mathbf{y}_{t-h}))$ is the directed temporal difference of the latent space projections $\mathbf{g}(\cdot)$ of the historic observation \mathbf{y}_{t-h} and its predecessor \mathbf{y}_{t-h-1} (taking the period π of the dimension into account). By applying the modulo operation on the difference in $\delta_{h,t}$, the $\cos(\delta_{h,t})$ term of c_{hist} realises a *directed* distance in the periodic dimension and thus enables to distinguish between represented and reversed temporal direction of trajectories. C_{rec} again is the self-reconstruction error described in a).

For the compatibility of the whole history of \mathbf{y}_t of length H , we define:

$$C_{hist}(H, t) = \frac{\sum_{h=1}^H \gamma^h c_{hist}(h, t)}{\sum_{h=1}^H \gamma^h} \quad (7)$$

where $\gamma \in [0; 1]$ is the discount factor for historic observations. As C_{rec} , C_{hist} can take values in $[-1; +1]$ whereas -1 corresponds to maximally incompatible and $+1$ to maximally compatible with the underlying UKR manifold.

The combination of a) and b) to one compatibility measures yields

$$C = \lambda C_{rec} + (1 - \lambda) C_{hist} \quad (8)$$

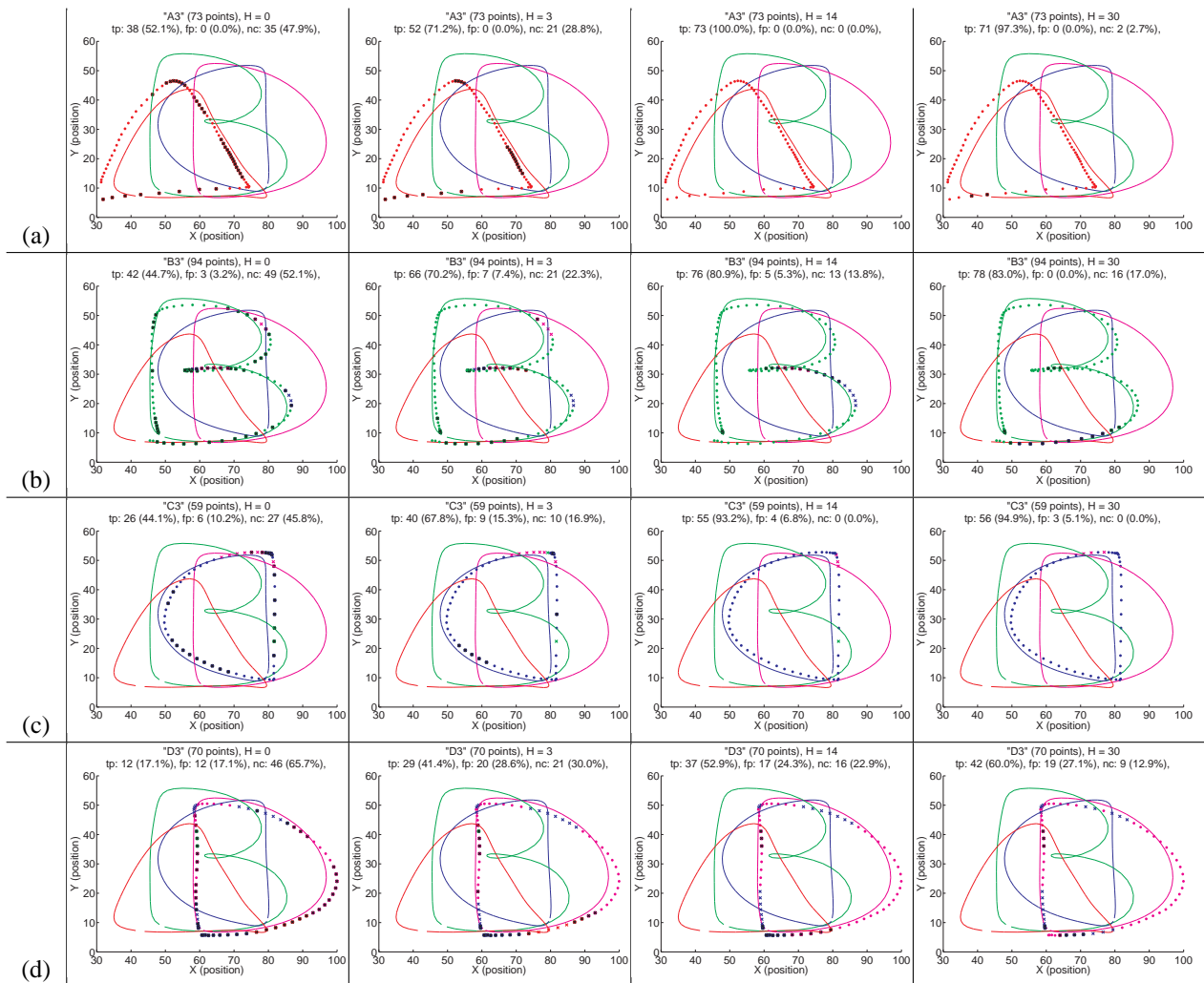


Fig. 6. Classification results for trajectories of single letters. Solid lines represent the trajectories reproduced by the four UKR manifolds: 'A'(red), 'B'(green), 'C'(blue), and 'D'(magenta). Points and crosses denote the observations of the test trajectory whereas coloured points depict true positive ('tp') and coloured crosses false positive ('fp') classifications. Here, the colours encode the true class corresponding to the UKR trajectories (see above). Black crosses on coloured points denote rejected points and thus points which are not compatible with neither of the tested manifolds (i.e. for which C is below the threshold). In this case, the colour of the underlying points denote the class with the maximal compatibility value. The headlines of the sub-figures denote the name of the dataset (e.g. "A3"), the size of the dataset, the history length H , the amount of true positive (tp) and false positive (fp) classifications and the amount of rejected points (nc: not classified). (a) Results for an observation of 'A' for different history lengths $H = 0, 3, 14$, and 30. (b) Results for 'B' for $H = 0, 3, 14$, and 30. (c) Results for 'C' for $H = 0, 3, 14$, and 30. (d) Results for 'D' for $H = 0, 3, 14$, and 30.

where $\lambda \in [0; 1]$ and thus $C \in [-1; +1]$.

Like this, C gives a measure for how compatible is the observation together with its history to the underlying manifold. In other words, C realises a measure to quantify the appropriateness of a candidate manifold to reproduce the observation and the history. The classification of the observation to one of several candidate classes then is performed by a winner-takes-all mechanism that works on the results of all UKR manifolds corresponding to the available candidate classes and thus chooses the class with the maximal compatibility.

To allow for rejecting observations, that is classifying that neither of the candidate manifolds is appropriate, a threshold for the compatibility measure C can be used as a lower boundary below which observations get rejected. For the initial experiments presented in the next section, we use the static value of zero as lower compatibility boundary. For later

experiments, it is possible to adapt this value according to the characteristics of the training data.

VII. EVALUATION

For the evaluation of the presented method, we trained four Structured UKR manifolds – one for each of the letters 'A', 'B', 'C', and 'D' – as described in Sec. IV. The represented trajectories are shown in Figs. 6 and 7 as solid lines ('A': red, 'B': green, 'C': blue, and 'D': magenta).

The evaluation of the compatibility works on the basis of single observations (together with their histories) and thus, the classification of trajectories is independent of the number of observations to be classified. In this sense, a classification of a trajectory as a whole is not performed directly but emerges from the classification of succeeding observations to the same class.

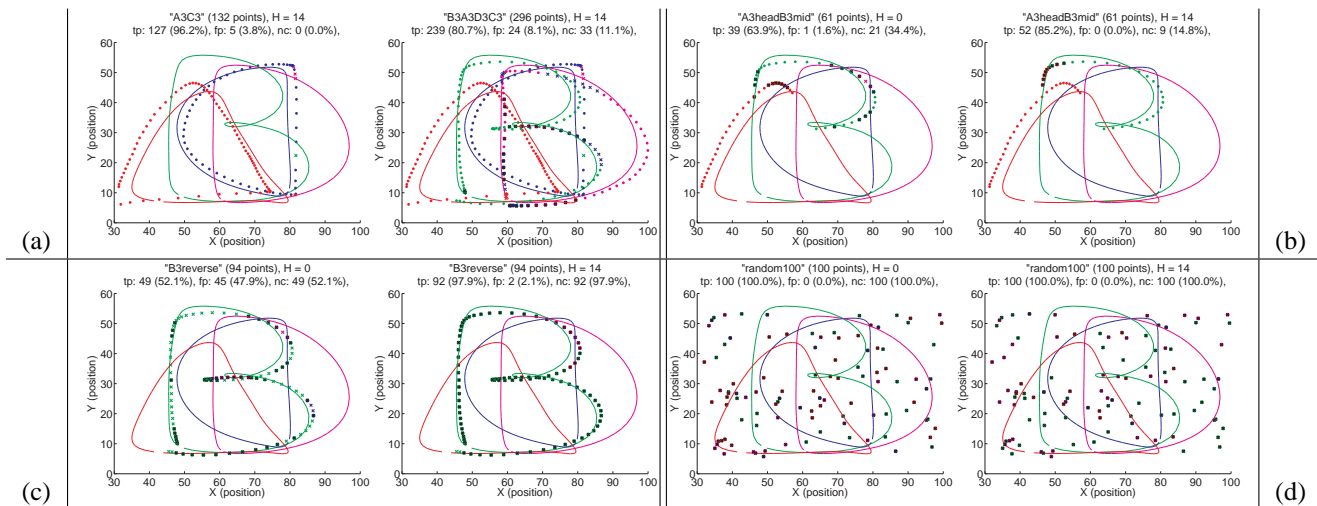


Fig. 7. Segmentation results for mixed and reversed datasets for history lengths $H = 0, 14$ (legend: see Fig. 6). (a) Segmentation results for concatenated whole-letter trajectories: (left) for 'A' and 'C'; (right) for 'B', 'A', 'D', and 'C'. (b) Segmentation results for concatenated first part of the 'A' trajectory and middle part of the 'B' trajectory. (c) Classification results for the reversed 'B' trajectory (data from Fig.6(b) in reverse order). (d) Classification results for a trajectory of 100 random observations.

In the first experiment, we utilise our method for the classification of trajectories corresponding to single and complete letters 'A', 'B', 'C', or 'D', respectively. As parameters, we use $\gamma = 0.9$ and $\lambda = 0.3$. Figure 6 depicts the results for the four different letters (rows), each for four different history lengths (columns; $H = 0, 3, 14, 30$). For the letter 'A' (Fig. 6(a)), already 52% of the observations are true positive (TP) classifications and no false positives (FP) as the rest gets rejected. However, by increasing the history length up to $H = 14$, a result of 100% TP can be achieved.

Letter 'B' (Fig. 6(b)) is more difficult as its trajectory is similar to parts of letter 'C' and 'D'. Without the usage of history information, only 44% of the observations can be safely classified correctly (TP) whereas only 3% are misclassified (FP) and the rest gets rejected. However, by incorporating the history, the TP rate increases to 80% (FP: 5%) for $H = 14$ up to 83% TP (FP: 0%) for $H = 30$.

For letter 'C' (Fig. 6(c)), similar results can be achieved whereas the TP rate is better (TP > 90%) with a worse FP rate at the same time.

Letter 'D' (Fig. 6(d)) yields the worst results with only 17% TP (FP: 17%, rejected: 66%) without history. Still, with increasing history length H , the TP rate reaches 60% with 27% misclassifications. These poor results are mainly caused by confusions with the letter 'C'. Nonetheless, the main parts of the trajectory can be classified correctly.

In general, one can state that for this kind of application on trajectories purely consisting of observations belonging to one class, longer histories of observations improve the classification results.

The second experiment concentrates on the segmentation of series of observations into the underlying classes. As before, the segmentation of trajectories is independent of the number of observations as it is based on the single observation compatibilities.

Figure 7(a) visualises the segmentation results for con-

catenated whole-letter trajectories. In principle, due to the classification of single points instead of the whole series, the results for a concatenation of two or more trajectories is similar to serially processing the pure whole-letter trajectories in the first experiment. However, for the transitions between two trajectories, the history distorts the classification results. Nevertheless, the results for the single letters are of similar quality as the results for single-letter trajectories (cp. Fig. 6): for the concatenation of 'A' and 'C', 96% of the observations are correctly classified (Fig.7(a,left)); the concatenation of 'B', 'A', 'D', and 'C' (containing the difficult 'B' and 'D' from the first experiment) result in 80% correct classifications. Note that this loss of performance is caused exclusively by the more difficult 'B' and 'D' letters and is independent from the higher number of observations.

Figure 7(b) depicts the results of another interesting application possibility: instead of whole-letter trajectories, only trajectory parts of different letters (here: first part of 'A' and middle part of 'B') are concatenated and processed by the segmentation. Whereas Fig7(b,left) does not incorporate history ($H = 0$) and thus yields suboptimal classification results, Fig7(b,right) uses a history of 14 observations ($H = 14$) and performs generally good. However, due to the consideration of the history, a hysteresis-like effect occurs when transitioning from the 'A' part to the 'B' part of the trajectory. Nevertheless, the segmentation into 'A' part and 'B' part is clearly visible.

In Figures 7(c-d), two other effects are visualised. Figure 7(c) depicts the influence of history consideration on the sensitivity against the temporal direction in which the trajectory is presented: evaluated are the same observations as in Fig. 6(b), but in reverse order. Without history (Fig.7(c,left), $H = 0$), the reverse order does not change the results compared to Fig.6(b, $H = 0$) (different TP/FP values result from NC being considered as the true classification in Fig.7(c,left)). On the other side, when considering the temporal history of

the observations, 97% get correctly rejected, as the reversed 'B' is not represented by the candidate UKR manifolds. Figure 7(d) demonstrates that random points are correctly rejected, either with or without using the history (notice that the figures are 2D projections of the 6D random data).

VIII. CONCLUSION

We presented a new approach for the classification and segmentation of motion data exploiting the manifold features of the Structured UKR manifolds which represent the candidate motions. Whilst this approach is limited to the classification and segmentation into known/represented candidate classes (with the possibility of rejecting observations effectively yielding one additional class for unknown/unrepresented motions which enables to semi-automatically recognise and train new classes), within these borders, it is very flexible and robust at the same time. The main strengths of this approach are: (1) it is – due to the definition of the compatibility features in the manifold domain instead of directly in the observation space – independent to a certain extend of the concrete task or observation space characteristics; (2) it is independent of a fixed window size or history length of the observations and can thus be applied on the 'raw' data without specific preprocessing steps and (3) as the computation of the compatibility measure only requires historic observations, it is basically also applicable to real time classification and segmentation of sensor data streams.

Another important benefit of the system is that, in principle, the recognition – or classification or segmentation, respectively – of a specific motion comes for free after having represented the same motion for reproduction and synthesis by means of Structured UKR manifolds, as the recognition mechanism directly works on the basis of these reproduction manifolds.

For our future work – after these very promising initial experiments with the ABCD-data – we plan to return to our original domain of dextrous manipulation where we want to apply the presented approach on different kinds of hand motions and manipulation movements. However, as the method does not rely on domain-specific data characteristics, the application on a variety of other (motion capture) data will be addressed as well.

ACKNOWLEDGEMENT This work has been carried out with support from the German Collaborative Research Centre "SFB 673 - Alignment in Communication" granted by the DFG and from the German Cluster of Excellence 277 "Cognitive Interaction Technology (CITEC)".

REFERENCES

- [1] A. M. Arsenio. Learning task sequences from scratch: applications to the control of tools and toys by a humanoid robot. In *Proceedings of the 2004 IEEE International Conference on Control Applications*, volume 1, pages 400–405, 2004.
- [2] C. Atkeson and S. Schaal. Robot learning from demonstration. In D. H. F. Jr., editor, *Proc. 14th Intl. Conf. on Machine Learning (ICML)*, pages 12–20. Morgan Kaufmann, 1997.
- [3] D. C. Bentivegna. *Learning from Observation Using Primitives*. PhD thesis, College of Computing, Georgia Institute of Technology, July 2004.
- [4] T. Beth, I. Boesnach, M. Haimerl, J. Moldenhauer, K. Bös, and V. Wank. Characteristics in human motion – from acquisition to analysis. In *IEEE Intl. Conference on Humanoid Robots HUMANOIDS*, page 56ff, 2003.
- [5] S. Calinon and A. Billard. Stochastic gesture production and recognition model for a humanoid robot. In *IEEE/RSJ Intl Conference on Intelligent Robots and Systems (IROS)*, 2004.
- [6] S. Calinon, F. Guenter, and A. Billard. On learning the statistical representation of a task and generalizing it to various contexts. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006.
- [7] R. Dillmann, O. Rogalla, M. Ehrenmann, R. Zöllner, and M. Bordegoni. Learning robot behaviour and skills based on human demonstration and advice: the machine learning paradigm. In *9th International Symposium of Robotics Research (ISSR '99)*, Snowbird, UT, USA, pages 229–238, October 1999.
- [8] M. Ehrenmann, R. Zöllner, O. Rogalla, S. Vacek, and R. Dillmann. Observation in programming by demonstration: Training and execution environment. In *Humanoids 2003, Karlsruhe/Munich, Germany, October 2003*, 2003.
- [9] S. Iba, C. Paredis, and P. Khosla. Interactive multi-modal robot programming. In *9th International Symposium on Experimental Robotics*, June 2004.
- [10] I.T. Jolliffe. *Principal Component Analysis*. Springer, New York, 2nd edition, 2002.
- [11] S. Klanke. *Learning Manifolds with the Parametrized Self-Organizing Map and Unsupervised Kernel Regression*. PhD thesis, Bielefeld University, 2007.
- [12] P. Meinicke, S. Klanke, R. Memisevic, and H. Ritter. Principal Surfaces from Unsupervised Kernel Regression. *IEEE Trans. on Pattern Analysis & Machine Intel.*, 27(9), 2005.
- [13] E. A. Nadaraya. On Estimating Regression. *Theory of Probability and Its Appl.* (9), 1964.
- [14] M. N. Niculescu and M. J. Mataric. Learning and interacting in human-robot domains. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 31(5):419–430, 2001.
- [15] M. Pardowitz, R. Haschke, J. Steil, and H. Ritter. Gestalt-Based Action Segmentation for Robot Task Learning. In *Proc. IEEE-RAS Intl. Conf. Humanoid Robots (HUMANOIDS)*, 2008.
- [16] G. Rizzolatti, M. Fabbri-Destro, and L. Cattaneo. Mirror neurons and their clinical relevance. *Nat Clin Pract Neuro*, 5(1):24–34, 2009.
- [17] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, Dec. 2000.
- [18] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3:233–242, 1999.
- [19] J. Steffen, R. Haschke, and H. Ritter. Towards Dextrous Manipulation Using Manifolds. In *Proc. Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008.
- [20] J. Steffen, S. Klanke, S. Vijayakumar, and H. Ritter. Realising Dextrous Manipulation with Structured Manifolds using Unsupervised Kernel Regression with Structural Hints. In *ICRA 2009 workshop: Approaches to Sensorimotor Learning on Humanoid Robots*, 2009. accepted.
- [21] W. Suleiman, E. Yoshida, F. Kanehiro, J.-P. Laumond, and A. Monin. On human motion imitation by humanoid robot. In *Proc. IEEE International Conference on Robotics and Automation*, 2008.
- [22] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.
- [23] G. S. Watson. Smooth Regression Analysis. *Sankhya, Ser.A*, 26, 1964.
- [24] M. Wertheimer. *A Source Book of Gestalt Psychology*, chapter Laws of Organization in Perceptual Forms, pages 71–88. Harcourt Brace, 1938.
- [25] R. Zöllner, T. Asfour, and R. Dillmann. Programming by demonstration: Dual-arm manipulation tasks for humanoid robots. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2004.
- [26] R. Zöllner, O. Rogalla, R. Dillmann, and J. Zöllner. Dynamic grasp recognition within the framework of programming by demonstration. In *10th IEEE International Workshop on Robot and Human Interactive Communication (Roman)*, 2001.