The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

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 both to reproduce and recognise the represented motions.**

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 [16], [6] 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 [14]. 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 [23], that propose unified representations for segmentation and action [13]. 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 [17], [19]. 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, we briefly review basic *Unsupervised Kernel Regression* and its modification to Structured UKR. Section IV will address the training and test data and in Section V, we present the new manifold features for classification and segmentation which then are evaluated in Section VI. We finish with a conclusion and an outlook on future work in Section VII.

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 [3]. [4] performs unsupervised clustering using Vector Quantisation (VQ) to segment the basic actions (codes) for a discrete HMM. This method is refined in [5] 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 [7]. These primitives of action (mainly concerned with grasping) are learned in a supervised way in [25]. 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 [24]. Several methods try to avoid the segmentation problem: [8] lets the user define the segmentation with explicit verbal commands that directly guide the robot through a demonstration. [2] and [20] do not decompose a task demonstration at all but search for direct mapping functions between input and output trajectories.

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 $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \in \mathbb{R}^{q \times N}$ of a set of observed data $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N) \in \mathbb{R}^{d \times N}$ and a corresponding functional relationship $\mathbf{y} = \mathbf{f}(\mathbf{x})$. It has been

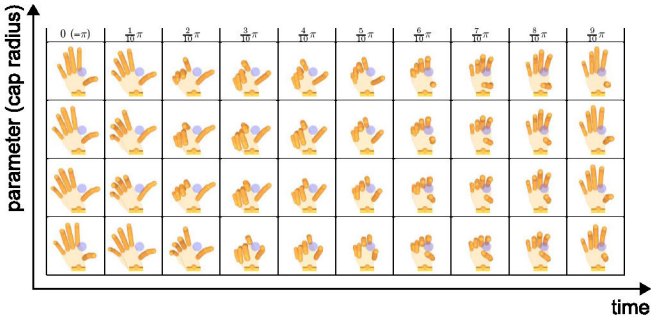

Fig. 1. Visualisation of a structured UKR latent space in the example of representing the turning movement of a bottle cap from [18]. The hand pictures correspond to the mappings of a regular grin in latent space back into the original input space of hand postures. The horizontal latent dimension corresponds to the time within the represented movement whereas the vertical latent dimension controls the motion parameter (the cap radius).

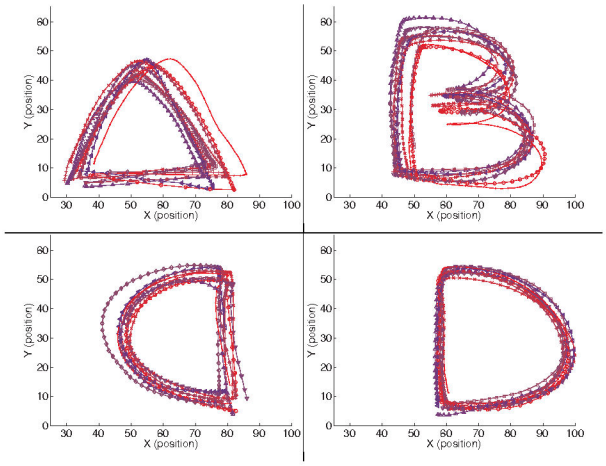

Fig. 2. The ABCD-data consist of recorded hand trajectories from drawing the letters 'A', 'B', 'C' and 'D' in the air. 6D positions/orientations are extracted from a monocular video camera pictures using ARToolkit. The camera is placed on top of the scene (in Z direction) resulting in good position/orientation precision in X- and Y-direction, but rather noisy height information in Z direction (not shown in the figure).

introduced as the unsupervised counterpart of the Nadaraya-Watson kernel regression estimator by Meinecke et al. in [11]. Further development has lead to the inclusion of general loss functions, a landmark variant, and the generalisation to local polynomial regression [10]. In its basic form, UKR uses the Nadaraya-Watson estimator [12], [22] as smooth mapping $\mathbf{f}\colon \mathbf{x} \in \mathbb{R}^q \rightarrow \mathbf{y} \in \mathbb{R}^d$ from latent to observed data space:

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

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

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

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

where $\mathbf{b}(\mathbf{x}; \mathbf{X}) = (b_1(\mathbf{x}; \mathbf{X}), b_2(\mathbf{x}; \mathbf{X}), \ldots, b_N(\mathbf{x}; \mathbf{X}))^T \in \mathbb{R}^N$ 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

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

Here, $\mathbf{B}(\mathbf{X}) = (\mathbf{b}(\mathbf{x}_1; \mathbf{X}), \mathbf{b}(\mathbf{x}_2; \mathbf{X}), \ldots, \mathbf{b}(\mathbf{x}_N; \mathbf{X}))$ is an $N \times N$ *basis function matrix*. Note that moving the $\mathbf{x}_i$ infinitively 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 [10]. 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 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 [9], Isomap [21] or LLE [15] 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 $\mathbf{x}^\star = \mathbf{g}(\mathbf{y}; \mathbf{X}) = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{f}(\mathbf{x}; \mathbf{X})\|^2$ which approximates $\mathbf{f}^{-1}(\cdot)$. For further details on original UKR, please refer to [11], [10].

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 [17] or trained [19] 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

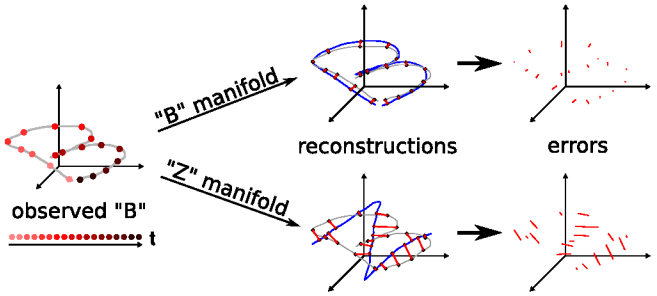

Fig. 3. Schematic visualisation of the reconstruction compatibility in observation space. The observed samples (red points, darker with development of time) are first projected into the manifold's latent space and then reprojected into original observation space: $\mathbf{f}(\mathbf{g}(\cdot))$ (cp. Sec. III). Depending on the similarity of observed (grey line) and represented (blue line) movement, the resulting errors significantly differ: as shown, errors are very small when self-reconstructing observed "B" samples with a matching "B" manifold (upper pathway), and contrarily very high, when reconstructing the same samples with a non-matching manifold (e.g. "Z": lower pathway).

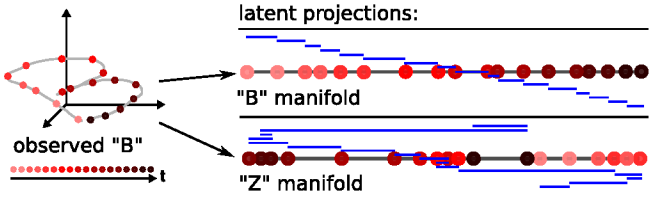

Fig. 4. Schematic visualisation of the history compatibility in latent space. Corresponding to Fig. 3, "B" samples are projected into the latent space of candidate manifolds (red points, darker with development of time); if observed and represented movements correspond to each other (top row, projecting "B" samples onto "B" manifold), the latent projection order matches the observation order and the sum of successor distances is minimal (blue lines). Otherwise (e.g. bottom row, projecting "B" samples onto "Z" manifold), the successor distances significantly differ from the matching case (compare overall length of blue lines in top and bottom schema).

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. For further details on Structured UKR, plese refer to [17], [19], [18].

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 [18] 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.

IV. 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 simpler and more descriptive data to facilitate the understanding of the method itself and the evaluation results, respectively. The data – depicted in Fig. 2 – consists of several (6D position/orientation) trajectories of a hand that draws the letters 'A', 'B', 'C', and 'D' in the air. Position/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 ARToolkit system (http://artoolkit.sourceforge.net).

V. Structured UKR for recognising movements

To enable the recognition of observed movements on the basis of trained Structured UKR manifolds, we introduce features in the manifold domain which express the similarity of the represented movement and the observation (described by data samples $\mathbf{Y}^o = \{\mathbf{y}^o_1, \ldots, \mathbf{y}^o_M\}$). To this end, we consider the latent projections $\mathbf{x}^o_t = \mathbf{g}(\mathbf{y}^o_t)$ of these samples together with their reprojections to observation space $\hat{\mathbf{y}}^o_t = \mathbf{f}(\mathbf{x}^o_t)$ (cp. Sec.III). We can then define two different features for the evaluation of the similarity as a basis for the recognition:

(a) The compatibility of single observations $\mathbf{y}^o_t$ with the manifold is reflected in its distance to its UKR reconstruction: $\Delta(t) = \| \mathbf{y}^o_t - \hat{\mathbf{y}}^o_t \|$. Depending on the degree of similarity of observed and represented data, this distance varies significantly, as visualised in Fig. 3. On this basis, we denote the *reconstruction compatibility* as:

$$C_{rec}(t;\mathbf{Y}) = -1 + 2 \cdot \exp\left(-\Delta^T \mathbf{V}^{-1} \Delta\right) \qquad (5)$$

where $\mathbf{V}(\mathbf{Y})^{-1} = \mathrm{diag}(\frac{1}{\mathrm{var}(\mathbf{y}_{\cdot,1})}, \ldots, \frac{1}{\mathrm{var}(\mathbf{y}_{\cdot,d})})$ is used to weigh the dimensions according to the corresponding variance in the UKR training data $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$. In this formulation, $C_{rec}(t;\mathbf{Y})$ can take values in the interval $[-1;+1]$ whereas -1 reflects worst and $+1$ best compatibility with the underlying UKR manifold.

(b) The chronological compatibility of observations in terms of the considered candidate manifold is reflected in the distances of the latent projections of successive observations, as shown in Fig. 4: if observed and represented sequence are similar, then the order of the latent representations equals the order of the observations and the sum of distances between latent projections of series of successive data points is small (Fig. 4, top) and can be very large otherwise (Fig. 4, bottom).

For a pair of successive observations $\mathbf{y}^o_{t-h-1}$ and $\mathbf{y}^o_{t-h}$, we denote the compatibility with the considered candidate manifold as:

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

where $\delta_{t-h} = mod_\pi(\mathbf{x}^o_{t-h-1} - \mathbf{x}^o_{t-h})$ is the *directed* temporal difference between the latent projections of $\mathbf{y}^o_{t-h}$ and its predecessor (taking the period π of the time dimension into account). Note that the *directed* difference enables a distinction between represented and chronologically reversed direction of trajectories. Here, C_{rec} is considered to also take the compatibility of the historic observations into account, which would be neglected otherwise.

For the history of length H of an observation $\mathbf{y}^o_t$, we then denote the *history compatibility* as:

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

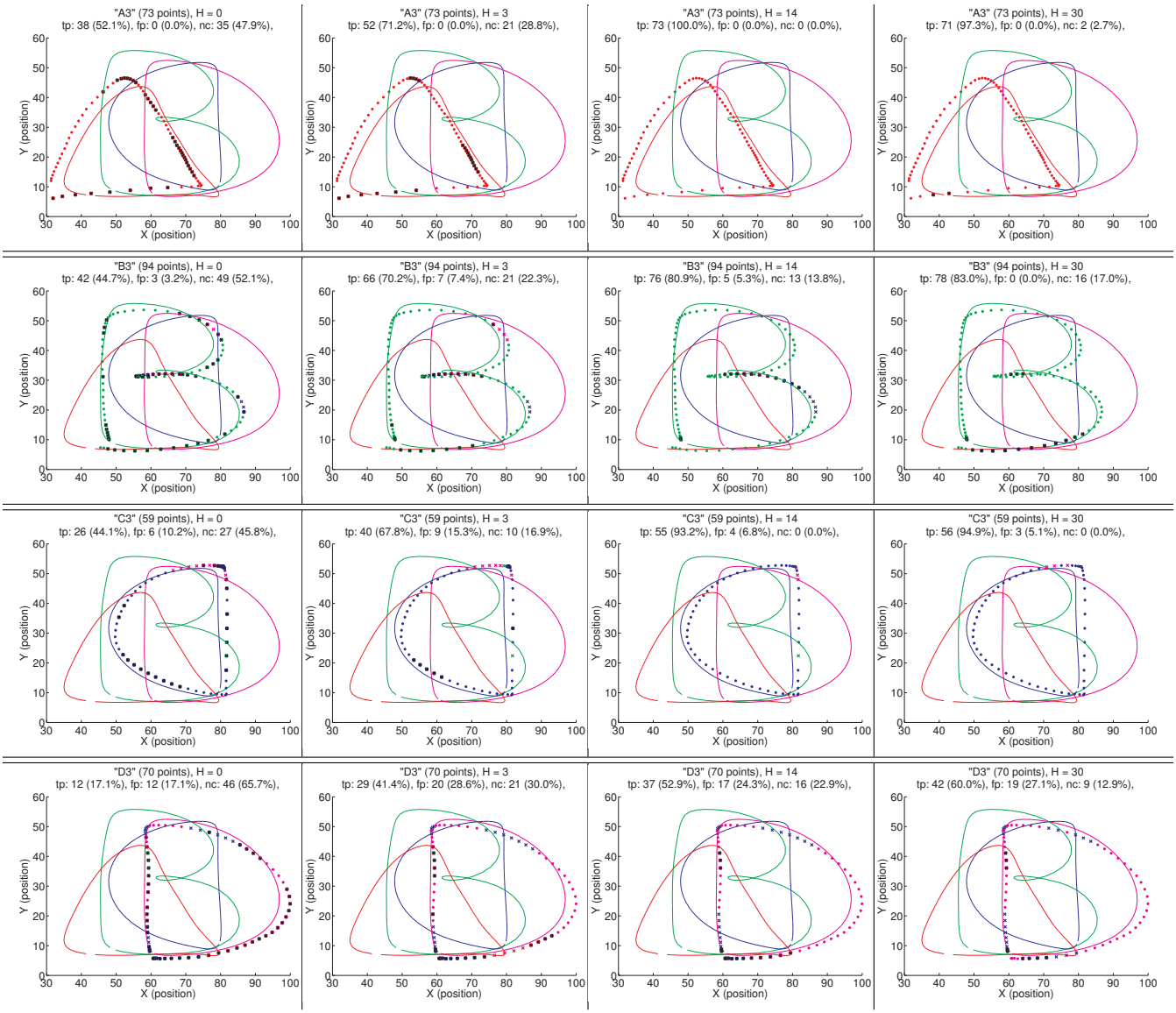

Fig. 5. Classification results for trajectories of single letters. Solid lines represent the four considered candidate UKR manifolds: 'A'(red), 'B'(green), 'C'(blue), and 'D'(magenta). Points and crosses denote the observations of the test trajectory whereas coloured *points* depict true positive ('tp') and coloured *crosses* false positive ('fp') classifications. Here, the colours encode the true class corresponding to the candidates (s.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). **(1st row)** Results for an observation of 'A' for different history lengths $H = 0, 3, 14$, and 30. **(2nd row)** Results for 'B' for $H = 0, 3, 14$, and 30. **(3rd row)** Results for 'C' for $H = 0, 3, 14$, and 30. **(4th row)** Results for 'D' for $H = 0, 3, 14$, and 30.

where $\gamma \in [0; 1]$ is the discount factor for historic observations. Like C_{rec}, C_{hist} can take values in $[-1; +1]$ whereas -1 reflects worst and $+1$ best compatibility with the underlying UKR manifold.

The combination of (a) and (b) to one overall compatibility measure using $\lambda \in [0; 1]$ as weighting factor yields:

$$C = \lambda C_{rec} + (1 - \lambda) C_{hist} \qquad \in [-1; +1]. \tag{8}$$

Like this, C provides a measure for the compatibility of an observation together with its history and the considered candidate manifold. In other words, C realises a measure to quantify the appropriateness of a candidate manifold to reproduce the observation together with the observed 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.

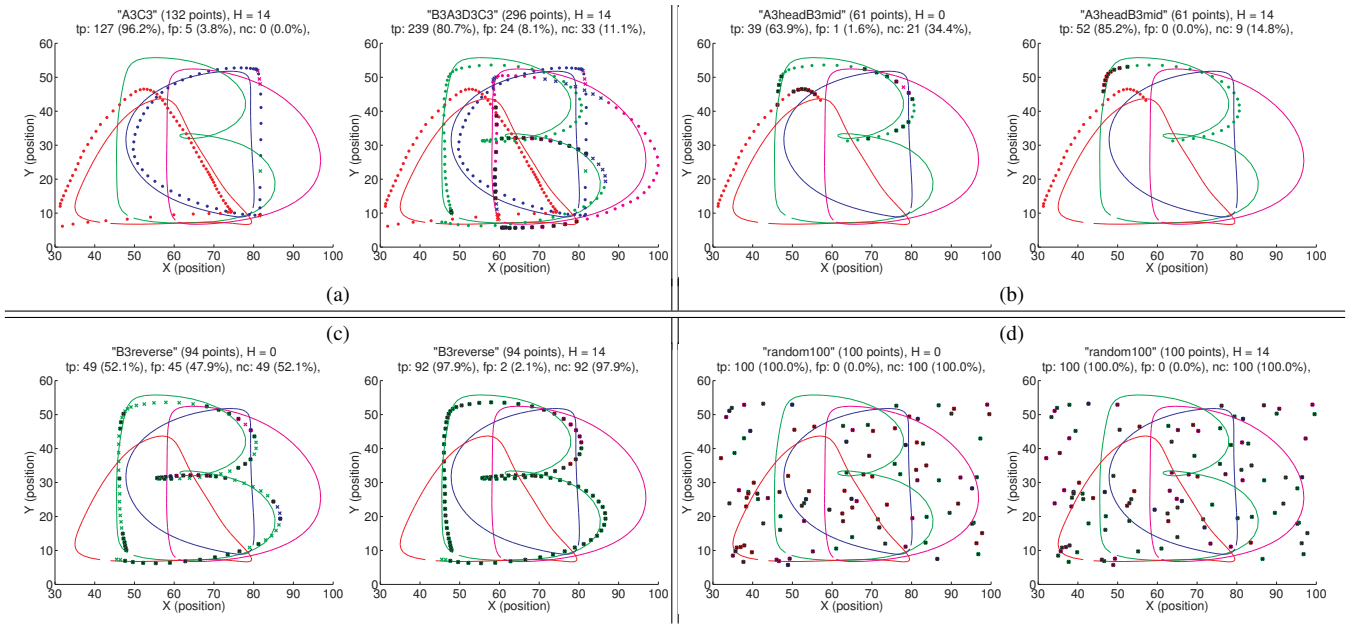

Fig. 6. Segmentation/classification results for history lengths $H = 0, 14$ for (legend: see Fig. 5): **(a)** concatenated whole-letter trajectories: (left) for 'A' and 'C'; (right) for 'B', 'A', 'D', and 'C'. **(b)** concatenated first part of the 'A' trajectory and middle part of the 'B' trajectory. **(c)** reversed 'B' trajectory (data from Fig.5(b) in reverse order). **(d)** 100 random observations.

VI. 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' described in Sec. III. The represented trajectories are shown in Figs. 5 and 6 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.

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 5 depict 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. 5, 1st row), without history, already 52% of the observations are true positive (TP) classifications and no false positives (FP) occur 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. 5, 2nd row) 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. 5, 3rd row), 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. 5, 4th row) 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, for the application on trajectories consisting of observations that only belong to one class, one can state that longer histories are beneficial for the classification result.

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 each observation is processed separately.

Figure 6(a) visualises the segmentation results for concatenated 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 concatenated letters are similar to the results for single-letter trajectories (cp. Fig. 5): for the concatenation of 'A' and 'C', 96% of the observations are correctly classified (Fig. 6(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 6(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 Fig. 6(b, left) does not incorporate history ($H = 0$) and thus yields suboptimal classification results, Fig. 6(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 6(c-d), two other effects are visualised. Figure 6(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. 5(b), but in reverse order. Without history (Fig. 6(c, left, $H = 0$)), the reverse order does not change the results compared to Fig. 5(b, $H = 0$) (different TP/FP values result from NC being considered as the true classification in Fig. 6(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 6(d) demonstrates that random points are correctly rejected, either with or without history (note that the figures are 2D projections of the 6D random data).

VII. 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 motions and 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) due to the definition of the compatibility features in the manifold domain instead of directly in the observation space, it is independent – to a certain extent – of the specific task or observation space characteristics; (2) it is independent of a fixed time window or history length of the observations and can thus be applied on the 'raw' data stream without specific preprocessing 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 ability to recognise (or classify or segment) a specific motion is automatically included in its representation as a Structured UKR manifold for reproduction and synthesis, because 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 *Proc. Int'l Conf. on Control Applications*, volume 1, pages 400–405, 2004.

[2] C. Atkeson and S. Schaal. Robot learning from demonstration. In *Proc. ICML*, 1997.

[3] T. Beth, I. Boesnach, M. Haimerl, J. Moldenhauer, K. Bös, and V. Wank. Characteristics in human motion – from acquisition to analysis. In *Proc. HUMANOIDS*, page 56ff, 2003.

[4] S. Calinon and A. Billard. Stochastic gesture production and recognition model for a humanoid robot. In *Proc. IROS*, 2004.

[5] S. Calinon, F. Guenter, and A. Billard. On learning the statistical representation of a task and generalizing it to various contexts. In *Proc. ICRA*, 2006.

[6] 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 *ISSR*, 1999.

[7] M. Ehrenmann, R. Zöllner, O. Rogalla, S. Vacek, and R. Dillmann. Observation in programming by demonstration: Training and exection environment. In *Proc. HUMANOIDS*, 2003.

[8] S. Iba, C. Paredis, and P. Khosla. Interactive multi-modal robot programming. In *Int'l Symp. on Experimental Robotics*, 2004.

[9] I.T. Jolliffe. *Principal Component Analysis*. Springer, New York, 2nd edition, 2002.

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

[11] 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.

[12] E. A. Nadaraya. On Estimating Regression. *Theory of Probability and Its Appl. (9)*, 1964.

[13] M. Pardowitz, R. Haschke, J. Steil, and H. Ritter. Gestalt-Based Action Segmentation for Robot Task Learning. In *Proc. HUMANOIDS*, 2008.

[14] G. Rizzolatti, M. Fabbri-Destro, and L. Cattaneo. Mirror neurons and their clinical relevance. *Nat Clin Pract Neuro*, 5(1):24–34, 2009.

[15] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, Dec. 2000.

[16] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3:233–242, 1999.

[17] J. Steffen, R. Haschke, and H. Ritter. Towards Dextrous Manipulation Using Manifolds. In *Proc. IROS*, 2008.

[18] 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.

[19] J. Steffen, S. Klanke, S. Vijayakumar, and H. Ritter. Towards Semi-supervised Manifold Learning: UKR with Structural Hints. In *Advances in Self-Organizing Maps (Proc. WSOM'09)*. Springer, 2009.

[20] W. Suleiman, E. Yoshida, F. Kanehiro, J.-P. Laumond, and A. Monin. On human motion imitation by humanoid robot. In *Proc. ICRA*, 2008.

[21] 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.

[22] G. S. Watson. Smooth Regression Analysis. *Sankhya, Ser.A*, 26, 1964.

[23] M. Wertheimer. *A Source Book of Gestalt Psychology*, chapter Laws of Organization in Perceptual Forms, pages 71–88. Harcourt Brace, 1938.

[24] R. Zöllner, T. Asfour, and R. Dillmann. Programming by demonstration: Dual-arm manipulation tasks for humanoid robots. In *Proc. IROS*, 2004.

[25] R. Zöllner, O. Rogalla, R. Dillmann, and J. Zöllner. Dynamic grasp recognition within the framework of programming by demonstration. In *Int'l Workshop on Robot and Human Interactive Communication (Roman)*, 2001.