Gestalt-Based Action Segmentation for Robot Task Learning

Michael Pardowitz*, Robert Haschke*, Jochen Steil† and Helge Ritter*†

Abstract— **In Programming by Demonstration (PbD) systems, the problem of *task segmentation and task decomposition* has not been addressed with satisfactory attention. In this article we propose a method relying on psychological gestalt theories originally developed for visual perception and apply it to the domain of action segmentation.**

We propose a computational model for gestalt-based segmentation called Competitive Layer Model (CLM). The CLM relies on features mutually supporting or inhibiting each other to form segments by competition. We analyze how *gestalt laws for actions* can be learned from human demonstrations and how they can be beneficial to the CLM segmentation method. We validate our approach with two reported experiments on action sequences and present the results obtained from those experiments.

I. Introduction

Programming a humanoid robot to achieve an individual task is a complex problem. One promising way to ease this is to equip cognitive robots with task learning abilities, that lets them learn a task from demonstrations of naive (non-expert) users. This paradigm is widely known as *Programming by Demonstration (PbD)* or *Imitation Learning*. Although several systems for Programming by Demonstration have been proposed (see [1], [2] 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: we extend the idea of perceptual grouping via gestalt rules from the domain of visual patterns into the domain of spatio-temporal processes arising from actions. Gestalt theory was successfully applied to the task of image segmentation in computer vision (see section II for an overview) and some Gestalt rules (e.g. such as "common fate") were already addressing the issue of forming perceptual groups within temporally changing patterns. It thus seems natural to extend this line of thinking more deeply into the realm of task decomposition and to explore the power of Gestalt laws for characterizing *good action primitives* for task decomposition. Such an approach can connect the so far primarily perception-oriented Gestalt approach with more current ideas on the pivotal role of the action-perception loop for representing and decomposing interactions.

While the original Gestalt approach relied heavily on explicit, "rule-like" characterizations of Gestalt principles [3] the early Gestaltists were already keenly aware that not everything that makes a good Gestalt is necessarily expressible in a crisp, rule-like linguistic format. Subsequent approaches exploiting field-like concepts for implementing Gestalt processes were reflecting this awareness. Generalizing Gestalt processes from the purely visual into the (inter-)action domain, we expect the significance of such more implicit representations to become even stronger. Therefore, we do not attempt to formulate any explicit rule-like Gestalt principles and instead employ from the outset a learning method for extracting implicit Gestalt principles implemented as "field-like" interactions that are learnt within a layered neural network (CLM, cf. below) from data.

The following section will review the literature on action and image segmentation. After that, section III will introduce the computational model for gestalt based action segmentation used in the experiments in this paper. Section IV describes a learning method to construct such models from human demonstrations in a supervised learning setup. Section V comments on the hardware setup and preprocessing steps to experimentally validate our model, and section VI reports the results obtained from these experiments. Finally, we conclude this paper with a discussion and an outlook on future work.

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.

[4] 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 [5]. [6] performs unsupervised

* Neuroinformatics Group, Bielefeld University, Universitätsstr. 25, D-33615 Bielefeld, Germany. Email: {mpardowi, rhaschke, helge}@techfak.uni-bielefeld.de

† CoR-Lab, Bielefeld University, Universitätsstr. 25, D-33615 Bielefeld, Germany. Email: jsteil@cor-lab.uni-bielefeld.de

This work has been conducted within and funded by the German collaborative research center " SFB 673: Alignment in Communication" granted by Deutsche Forschungsgemeinschaft.

The authors would like to kindly thank Christof Elbrechter from the Neuroinformatics Group, University of Bielefeld, for providing the framework and algorithm implementations for the vision system used in this paper.

clustering using Vector Quantisation (VQ) to segment the basic actions (codes) for a discrete HMM. This method is refined in [7] 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 which 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 [9]. A similar way is proposed in [10] where a user demonstration is segmented based on the most likely primitives performed in each timestep. [11] applies a similar method using a winner-takes-all selection of the most probable behavior to segment a sequence of navigation tasks.

Several methods try to avoid the segmentation problem: [12] lets the user define the segmentation with explicit verbal commands that directly guide the robot through a demonstration. [13] and [14] do not decompose a task demonstration at all but search for direct mapping functions between input and output trajectories.

In the domain of computer vision the segmentation problem has drawn continuous interest over several decades. In particular, gestalt-based approaches accomplished complex image segmentation tasks as reported in [15]. In this domain one can coarsely divide direct probabilistic models [16] from neural methods like in [17], [18].

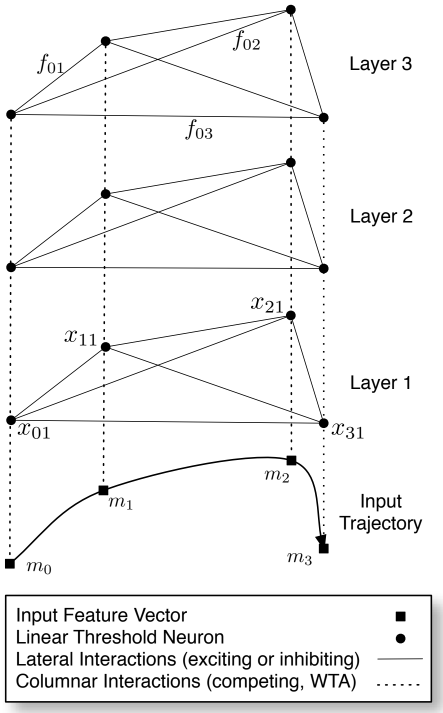


Fig. 1. The Competitive Layer Model Architecture for an input trajectory with $L = 3, N = 4$. The neurons are shown as black dots. The solid and dotted lines represent lateral and columnar interactions respectively. Lateral connections are labelled with their according weights $f_{tt'}$. The input features $m_0 \dots m_3$, the neurons of layer 1 and the lateral interactions for neuron 0 in layer 3 are labeled with the corresponding notation

III. The Competitive Layer Model

The Competitive Layer Model (CLM) consists of L neuron layers. Each layer $\alpha = 1 \dots L$ acts as a "feature map" associating its positions t with feature value combinations m_t from the chosen feature space V. In our specific application, t represents the point of time where a feature vector m_t was recorded. Figure 1 shows a simple example of a CLM with $L = 3$ and $N = 4$: Four input feature vectors are extracted from an input trajectory. A single neuron represents that trajectory segment in each layer. The neurons are connected laterally and columnarly. Identical positions t in different layers share the same input line and receive a (usually scalar) input activity h_t. In the simplest case we assume that the (prespecified) feature maps are layer-independent (no α index) and are implemented in a discretized form by N linear threshold neurons with activities $x_{t,\alpha}$ located at the same set of discrete positions r in each layer α. Therefore, the system consists of L identical feature maps that can be activated in parallel.

The idea of the CLM is to use this coding redundancy to introduce a competitive dynamics between layers for the coding of features in its input pattern h_t. The outcome of this competition partitions the features comprising the input into disjoint groups, with each group being characterized as one subset of features coded by activity in the same "winning layer".

The competitive dynamics is based on two different sets of connections between the neurons of the CLM. Two timesteps t and t' belong to the same segment if they show simultaneous activities $x_{t\alpha} > 0$ and $x_{t'\alpha} > 0$ in a certain layer α. In each layer, the neurons are fully connected with symmetric weights $f_{tt'} = f_{t't}$. The values of $f_{tt'}$ establish semantic coherence of features. Positive values indicate feature compatibility through excitatory connections while negative values express incompatibility through inhibitory connections. Two features t, t' with high positive values for $f_{tt'}$ are more likely to belong to the same action segment than two features with negative values for$f_{tt'}$.

The purpose of the layered arrangement and the columnar interactions in the CLM is to enforce a dynamical assignment of the input features to layers that respects the contextual information stored in the lateral interactions $f_{tt'}$. This assignment segments the input into partitions of matching features which links each feature t with its unique label $\alpha(r)$. A columnar Winner-Takes-All (WTA) circuit realizes this unique assignment using mutual symmetric inhibitory interactions with strength $J > 0$ between neural activations $x_{t\alpha}$ and $x_{t\beta}$. Due to the WTA coupling, only one neuron from a single layer can be active in every column, as soon as a stable equilibrium state of the CLM is reached.

Equation (1) combines the inputs with the lateral and columnar interactions into the CLM dynamics (see [18]):

$$\dot{x}_{t\alpha} = -x_{t\alpha} + \sigma\left(J(h_t - \sum_{\beta} x_{t\beta}) + \sum_{t'} f_{tt'} x_{t'\alpha} \right). \quad (1)$$

Here is $\sigma(x) = \max(0, x)$ and h_t is the significance of the detected feature t as obtained by some preprocessing steps. For simplicity, we assume all h_t to be equal to one in this paper.

A process that updates the neural activations according to the dynamics of equation (1) converges towards several possible stable states, as shown in [18]. These stable states all satisfy the consistency conditions

$$\sum_{t'} f_{tt'} x_{t'\beta} \leq \sum_{t'} f_{tt'} x_{t'\hat{\alpha}(r)}, \quad (2)$$

which indicate the assignment of a feature t to the layer $\hat{\alpha}(r)$ with the highest lateral support for that feature. This corresponds to the layer that already contains the most features t' compatible with t. Since every column t has only a single $\hat{\alpha}(r)$, $\hat{\alpha}$ establishes a partitioning of features into disjunctive sets of mutually compatible features, called segments.

Compatibility of features is coded in the lateral inhibitory or excitatory weights $f_{tt'}$. The correct choice of the lateral weights determines the quality of the partitioning. The next section describes, how good lateral interactions can be learned.

IV. Determining the Lateral Interactions

In the last section, we used the compatibility function $f_{tt'} = f(m_t, m_{t'})$ which quantifies the preference to bind similar features with positive values and separate dissimilar features by negative values. This section describes, how the interactions can be learned from a labelled training sequence.

Assuming that we have a consistent labelling $\hat{\alpha}(r)$ for a training sequence of length N (that is: $t = 1, \ldots, N$), we can construct one attractor state by

$$y_{t\hat{\alpha}(r)} = 1 \text{ and } y_{t\beta} = 0 \quad \forall t, \forall \beta \neq \hat{\alpha}(t).$$

The objective is to learn $f_{tt'}$ such that it satisfies the target inequalities (2) for the $y_{t\beta}$:

$$\sum_{t'} f_{tt'} y_{t'\beta} \leq \sum_{t'} f_{tt'} y_{t'\hat{\alpha}(r)} \quad (3)$$

As was shown in [18], these target inequalities are estimated by a matrix $\hat{F} = (\hat{f}_{tt'})$, which is given by

$$\left(\hat{f}_{tt'}\right) = \hat{F} = \sum_{\gamma} \sum_{\mu \neq \gamma} (\mathbf{y}_\gamma - \mathbf{y}_\mu)(\mathbf{y}_\gamma - \mathbf{y}_\mu)^T. \quad (4)$$

Here, the vectors $\mathbf{y}_\gamma$ and $\mathbf{y}_\mu$ represent all components in the γth and μth layer, that is $\mathbf{y}_\gamma = (y_{1\gamma}, \ldots, y_{N\gamma})^T$.

So far, the discrete interaction matrix $\hat{F}$ obtained by (4) is defined only on the feature values present in the training sequence. Therefore, we have to generalize it to a real interaction function defined on the full feature domain. Several approaches for this generalization have been discussed in [17], [18]. Here we follow the approach in [18] to decompose the interaction function $f_{tt'}$ into a linear combination of a set of K symmetric basis interaction functions $g^j_{tt'} = g^j(m_t, m_{t'})$ which are defined on the whole feature space such that

$$f_{tt'} = \sum_{j=1}^{K} c_j g^j_{tt'}.$$

A detailed analysis (see [18]) reveals that under the condition that the basis interactions are assumed to be binary step functions ($g^j_{tt'} \in \{0, 1\}$) that describe a disjunct partitioning of the feature space ($g^j_{tt'} g^i_{tt'} = \delta_{ij}$),

$$c_j = \sum_{t,t'} \hat{f}_{tt'} g^j_{tt'} \quad (5)$$

yields a practical estimate.

Following the conditions above, the choice of the basis interaction functions $g^j_{tt'}$ has to satisfy two constraints: symmetry and disjunction partitioning. To satisfy the symmetry constraint we transform the feature space into a generalized proximity space $\mathrm{D} = \mathbf{R}^P$

$$\mathbf{d}_{tt'} = \left((m_{t1} - m_{t'1})^2, \ldots, (m_{tP} - m_{t'P})^2\right)$$

with m_{ti} denoting the ith component of the feature vector m_t.

We then map each proximity vector $\mathbf{d}_{tt'}$ to a multidimensional Voronoi map with K cells and a prototype $\tilde{\mathbf{d}}_j$ corresponding to each cell, such that each Voronoi cell is defined as

$$V_j = \left\{ (m_t, m_{t'}) | \forall i \neq j : ||\mathbf{d}_{tt'} - \tilde{\mathbf{d}}_j|| \leq ||\mathbf{d}_{tt'} - \tilde{\mathbf{d}}_i|| \right\}.$$

Since a Voronoi tesselation results in a disjunct partitioning, a choice of

$$g^j_{tt'} = \begin{cases} 1, & (m_t, m_{t'}) \in V_j \\ 0, & (m_t, m_{t'}) \notin V_j \end{cases} \quad (6)$$

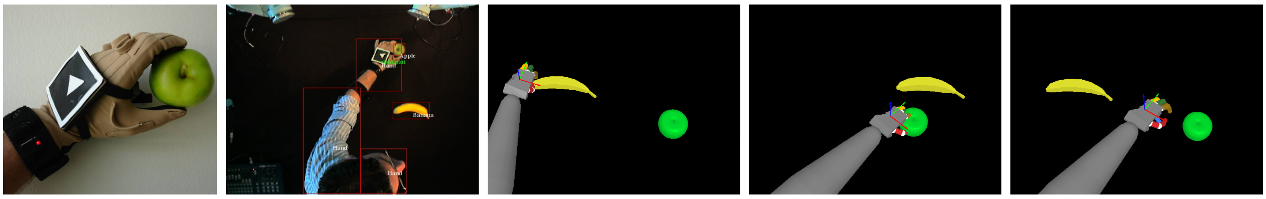
satisfies all conditions to apply equation (5).

The representation of the basis functions as Voronoi cells in the symmetric proximity space enables us to learn interaction functions $f_{tt'}$ from a relatively small data set and achieve good generalization results. In order to obtain the interaction functions for a new sequence, we have to compute the proximity vector for each feature pair, search for its nearest prototype vector $\tilde{\mathbf{d}}_j$ and return the interaction coefficient c_j of this prototype.

Following [18], equation (5) can be rewritten as

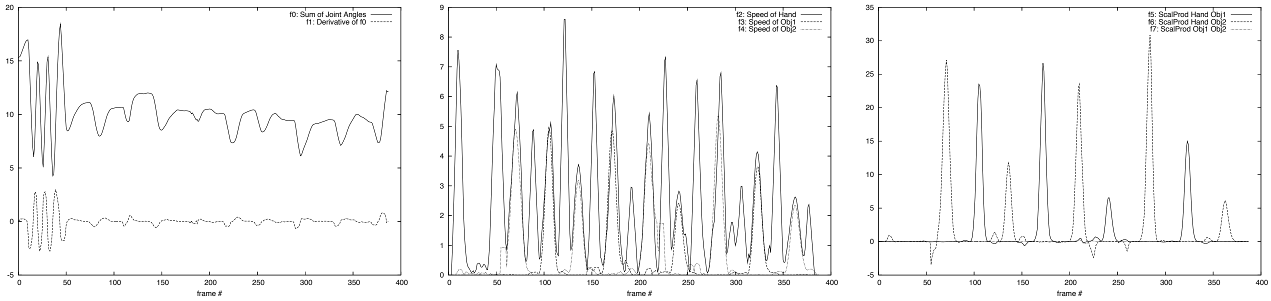
$$c_j = \sum_{t,t' | \alpha(t) = \alpha(t')} g^j_{tt'} - \lambda \sum_{t,t' | \alpha(t) \neq \alpha(t')} g^j_{tt'}.$$

Here, λ is a scaling factor that effects the grouping behavior of the CLM: Higher values of λ result in higher values for the interaction function, that is a higher attraction. This eventually leads to fewer but larger groups. Lower values of λ, in turn, result in a more fine-grained segmentation.



(a) Sensors (b) Camera view (c) Frame #135 (d) Frame #172 (e) Frame #229

Fig. 2. Sensors and Key Frames. (a) Sensing devices: A Cyberglove for hand posture tracking together with a ARToolkit Marker mounted on the back of the hand. (b) The marker and the object are tracked with an overhead camera. Marker and color blob tracking is performed. (c-e) Key frames of recorded sequence. Visualization of hand and object positions and joint angles of the hand.



(a) Sum of Joint Angles and Derivative (b) Absolute Speed of Hand and Objects (c) Scalar Products of Directions of Hand / Object

Fig. 3. Extracted features. (a) f_0, f_1 (b) $f_2 - f_4$ (c) $f_5 - f_7$. See text for explanation.

V. Experimental Setup and Procedure

In order to obtain data from human task demonstrations, we used the following setup: An Immersion Cyberglove [19] senses the finger joint angles in order to capture the human hand posture and transmits them via a bluetooth connection to a computer. An ARToolKit Marker[1] was fixed with elastic straps on the back of the hand (see figure 2a). A Sony DFW-VL500 Firewire camera with a resolution of 640x480 pixels tracked the scene from an overview perspective (see figure 2b). This setting allowed us to locate a person's hand in 6D-space (from the ARTookKilt Marker) together with its posture (from the Cyberglove sensor readings) and estimate the actions the user performs. Figure 2(a) shows this setup.

Equipped with the sensors described above, the user faced an environment which contained various objects (i.e. apples, bananas). To obtain information on the position and movements, we used methods for tracking colored blobs to record object movements in the same overhead camera images that were used for the ARToolKit tracking. The results are shown in figure 2(b).

Several sequences with lengths N varying between 350 and 400 have been recorded with this sensor setup. Three frames from the sequence used for the experiments described in section VI are displayed in figure 2 (c-e) as a 3D reconstruction of the obtained data. The hand position and posture together with the object positions yielded enough information to extract the following features (see also figure 3):

1) The sum of the joint angles of the human hand:
$$f_0(t) = \sum_i \theta_i(t).$$
This gives a measure for the degree of opening/closing of the hand.
2) The derivative of f_0: $f_1(t) = \dot{f}_0(t)$. This gives large values at times when the user opens or closes a grip.
3) The velocities of the hand and the objects
$$f_2(t) = |\overrightarrow{v}_{hand}(t)|, f_3 = |\overrightarrow{v}_{obj1}(t)|, f_4 = |\overrightarrow{v}_{obj2}(t)|.$$
The velocities tend to remain positive during connected segments and disappear only at points where the goal context of the user changes.
4) The co-occurrence of parallel movements is calculated using the scalar product of the movements of the hand with an object or between the two objects respectively:
$$f_5(t) = \overrightarrow{v}_{hand}(t)^T \cdot \overrightarrow{v}_{obj1}(t)$$
$$f_6(t) = \overrightarrow{v}_{hand}(t)^T \cdot \overrightarrow{v}_{obj2}(t)$$
$$f_7(t) = \overrightarrow{v}_{obj1}(t)^T \cdot \overrightarrow{v}_{obj2}(t)$$
These features give large values for segments where an object moves in the same direction as the hand, or the objects move parallel to each other.
5) The frame number: $f_8(t) = t$. This allows us to explicitly take into account the time dimension, which leads to more continuous segments and not creating too many unlinked fragments.

These features are computed for each frame of a demonstration sequence. The interactions are learned according

[1] see http://artoolkit.sourceforge.net/

to the approach outlined in section IV. After that, two different experiments are conducted:

1) *Exploration of parameter space:* During the training phase and the execution of the CLM dynamics the following parameters were systematically varied:
   - Binary vs. continuous interactions: In the binary experimental condition the interactions were thresholded, that is interactions larger than 0.5 were set to 1 and interactions smaller than 0.5 were set to 0.
   - The scale factor λ was systematically varied in ten equally spaced intervals between the bounds
   $$\frac{(\mathbf{c}^+)^T\mathbf{c}^+}{(\mathbf{c}^-)^T\mathbf{c}^+} < \lambda < \frac{(\mathbf{c}^+)^T\mathbf{c}^-}{(\mathbf{c}^-)^T\mathbf{c}^-}$$
   which are suggested in [18].
   - The number of prototypes K for the basis functions to be learned was varied between 100 and 280 in steps of 20.

   The aim of this experiment was to find a parameter set that yields optimal classification accuracy.
2) *Comparison with a standard classifier:* Here the scale factor λ and the number of prototypes K was fixed to 1.3 and 260, respectively, which was close to the optimum as found by the first experiment. Classification accuracy was compared to a Feed-Forward Neural Network with 9 hidden units that was trained with Backpropagation using a learning rate $\epsilon = 0.2$ and a weight decay term of 0.005 for $10,000$ episodes. In a first experimental condition, the classification rate of the CLM and the Feed-Forward-Classifier was tested on the same demonstration sequence that was used for training. In a second condition it was tested on a new sequence to assess the generalization capabilities of the different classifiers.

After the interaction functions of the CLM are trained with the methods outlined in section IV, the dynamics of the CLM is executed (either on the training data or a new sequence, depending on the experimental conditions) according to the rules stated in section III. When the dynamics have converged, the correct classification rate is determined by comparison with labellings generated by hand.

VI. Results

Figure 4 shows the interactions as learned from the training sequence. From the first visual impression one can easily distinguish the interaction patterns for movements with grasped objects (b), (c) and with empty hand (d). On execution of the CLM dynamics, the assignment converges from complete randomness towards a satisfying segmentation of the input sequence.

Considering the experiment designed to explore the parameter space, figure 5 shows the resulting classification rates plotted versus the parameter λ. Generally, one can conclude that thresholded binary interactions (fig. 5 (b)) achieve better segmentation results even for a smaller number of learned basis functions than the continuous interaction case (fig. 5 (a)). This is apparently due to the greater simplicity of binary functions, which can be approximated more easily. The data plotted in figure 5 additionally suggests that a maximum of segmentation accuracy can be found around a parameter set including a scale factor $\lambda = 1.3$ and the number of function prototypes $K = 260$. Therefore, λ and K were fixed to these values during the second experiment.

The results from the second experiment are recorded in tables I and II. Table I shows the classification rate averaged over 30 training runs on the same data set. The CLM with binary interactions clearly outperforms the continuous CLM. Both the binary and the continuous CLM perform better than the Neural Network trained by the Backpropagation algorithm. This effect becomes even more evident when we take into account the figures for the generalization ability. Both CLMs show a much higher tolerance to new data than the Backpropagation network.

	μ	σ
FFNN	0.563	0.214
CLM cont.	0.763	0.144
CLM bin.	$\mathbf{0.840}$	0.011

TABLE I

Experiment II: Comparision of classification accurracy on training data set for different methods

	μ	σ
FFNN	0.457	0.172
CLM cont.	0.750	0.143
CLM bin.	$\mathbf{0.801}$	0.148

TABLE II

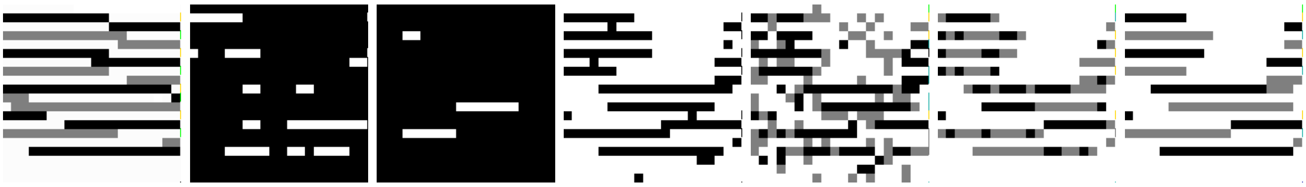
Experiment II: Comparision of generalization ability for different methods: Classification accuracy for unknown test data set

VII. Conclusion and Future Work

In this article we proposed a gestalt-based method for task segmentation. It applied the CLM, previously found well-suited for visual segmentation. We developed methods for learning of the CLM parameters and validated them in an object manipulation scenario.

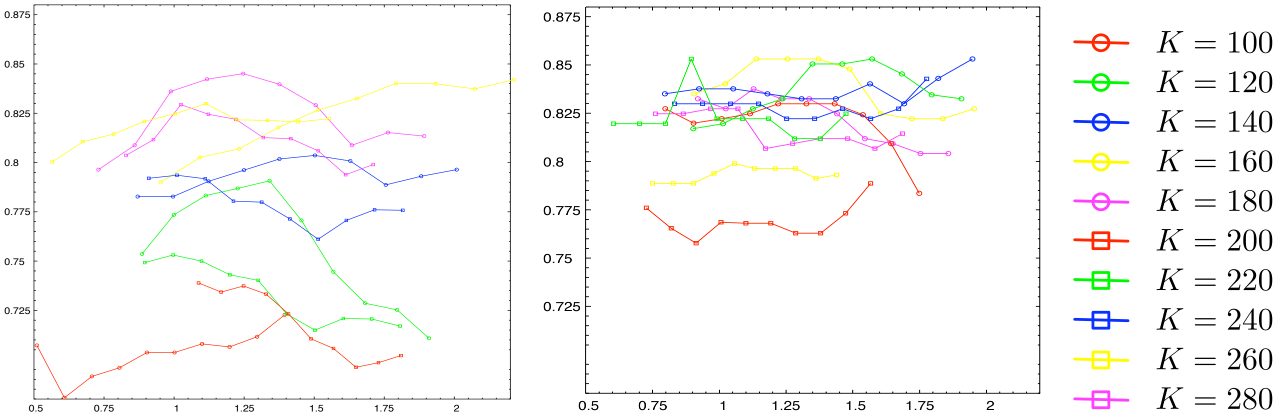
Future work will focus on several different aspects: The full strengths of the competitive layer model, as shown in visual segmentation experiments, still have to be exploited. A background/reject layer might prove useful to reject the assignment to a certain segment in areas of ambiguity, where information is sparse. Different layer classes with different lateral interactions might be practical to segment more complex actions, e.g. rotating a grasped object, unscrewing a bottle or similar fine manipulations.

Moreover, each found segment has to be thoroughly analysed and interpreted to facilitate further learning and a final imitation of the demonstrated movements. Effects and preconditions of actions and temporal sequences of segments can yield important cues for an imitation learning system that will be exploited in further research.



(a) Manual (b) Int. #135 (c) Int. #172 (d) Int. #229 (e) Dyn. t = 1,000 (f) Dyn. t = 100,000 (g) Dyn. t = 200,000

Fig. 4. Interactions and Assignment. Each frame corresponds to a single pixel, with frame #0 in the lower left, and frame #400 in the upper right corner. The first 20 frames form the lowest, the last 20 frames the highest row. (a) Manual segmentation. White segments correspond to hand movements without an object being grasped. Black and grey segments correspond to movements with the banana or the apple in the hand, respectively. (b-d) Interactions $f_{tt'}$ as functions of pixel position t' for different fixed choices of t. White pixels indicate a value of 1, black pixels a value of 0. (b) $r = 135$: Banana grasped (c) $r = 172$: Apple grasped (d) $r = 229$: Movement without an object. (e-g) Execution of CLM-Dynamics (same color coding scheme as for manual segmentation) (e) after $1,000$, (f) after $100,000$ iterations, (g) after $200,000$ iterations (converged result).



(a) Continuous Interactions (b) Binary Interactions (c) Legend

Fig. 5. Results of Experiment I: Exploration of Parameter Space. The values for λ and the resulting classification rates are plotted on the x- and y-axis, respectively. The number of function prototypes was varied according to the color and dot code depicted in the legend (c).

References

[1] S. Schaal, “Is imitation learning the route to humanoid robots?” *Trends in Cognitive Sciences*, vol. 3, pp. 233–242, 1999.

[2] 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*, October 1999, pp. 229–238.

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

[4] 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*, vol. 1, 2004, pp. 400–405.

[5] 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*, 2003, p. 56ff.

[6] 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.

[7] 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.

[8] M. Ehrenmann, R. Zöllner, O. Rogalla, S. Vacek, and R. Dillmann, “Observation in programming by demonstration: Training and exection environment,” in *Humanoids 2003, Karlsruhe/Munich, Germany, October 2003*, 2003.

[9] 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.

[10] D. C. Bentivegna, “Learning from observation using primitives,” Ph.D. dissertation, College of Computing, Georgia Institute of Technology, July 2004.

[11] M. N. Nicolescu and M. J. Mataric, “Learning and interacting in human-robot domains,” *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, vol. 31, no. 5, pp. 419–430, 2001.

[12] S. Iba, C. Paredis, and P. Khosla, “Interactive multi-modal robot programming,” in *9th International Symposium on Experimental Robotics*, June 2004.

[13] C. Atkeson and S. Schaal, “Robot learning from demonstration,” in *Proc. 14th Intl. Conf. on Machine Learning (ICML)*, D. H. F. Jr., Ed. Morgan Kaufmann, 1997, pp. 12–20.

[14] 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.

[15] A. Desolneux, L. Moisan, and J.-M. Morel, “Edge detection by Helmholtz principle,” *Journal of Mathematical Imaging and Vision*, vol. 14, no. 3, pp. 271–284, 2001. [Online]. Available: citeseer.ist.psu.edu/article/desolneux01edge.html

[16] ——, *Seeing, Thinking and Knowing*. Kluwer Academic Publishers, 2004, ch. Gestalt Theory and Computer Vision, pp. 71–101.

[17] S. Weng and J. J. Steil, “Data driven generation of interactions for feature binding and relaxation labeling,” in *ICANN ’02: Proceedings of the International Conference on Artificial Neural Networks*. London, UK: Springer-Verlag, 2002, pp. 432–437.

[18] S. Weng, H. Wersing, J. Steil, and H. Ritter, “Learning lateral interactions for feature binding and sensory segmentation from prototypic basis interactions,” *IEEE TNN*, vol. 17, no. 4, pp. 843–863, 2006.

[19] Immersion, “Cyberglove ii wireless glove.” [Online]. Available: http://www.immersion.com/3d/products/cyber_glove.php