Learning of demonstrated Grasping Skills by stereoscopic tracking of human hand configuration

Markus Hueser*, Tim Baier† and Jianwei Zhang‡
Group Technical Aspects of Multi Modal Systems
Hamburg University, Hamburg, Germany
*Email: hueser@informatik.uni-hamburg.de
†Email: tbaier@informatik.uni-hamburg.de
‡Email: zhang@informatik.uni-hamburg.de

***Abstract*— In this paper a novel approach to learning by demonstration (LbD) is presented. A multimodal service robot is taught grasping skills by a human instructor who demonstrates a grasping action. Our approach contributes novel solutions to the aspects of robustly tracking the demonstrator's hands in real time as well as to the transformation of tracking results into grasping skills. To track the demonstrator's hands in stereoscopic images a Mean-Shift-like algorithm is adapted. For the very first time this algorithm is applied to local binary patterns (LBP) and color histograms. To retrieve the hand configuration we use view-based Principal Component Analysis (PCA). To develop grasping skills from tracking results the robot repetitively tracks the demonstrator's grasping actions and transforms the results into three-dimensional self organizing maps (SOMs). The SOMs give a spatial description of the collected data and serve as data structures for a reinforcement learning (RL) algorithm which optimizes trajectories for use by the robot. The approach is applied to a multimodal service robot. Experiments show the effectiveness of the LBP-enhanced Mean-Shift-like tracking and the robustness of LbD based on SOMs and RL**

I. INTRODUCTION

Grasping objects is a basic skill for service robots. It is crucial to manipulation tasks and interaction with the environment. In most industrial applications it is solved via teaching-by-doing or static programs. Additionally, the use of robots is very complex so that only a few experts are able to handle them. However, for a service robot scenario which demands robust and also adaptive behavior in non-static environments like e.g. office environments, it is important to further simplify interaction between robots and humans.

To learn grasping actions by demonstration, it is important that the robot detects an action and learns how to reproduce this action. In the case of grasping a certain object, the robot has to learn an approach trajectory, the tag point, the type of grasp and an adequate force for a stable grasp. These parameters should preferably be learned in real-world scenarios. That means learning by demonstration (LbD) can be done considering e.g. video, audio and laser range data of a human demonstrator performing a grasping task. The data should be captured by the robot itself without restricting the environment by the demands of special hardware like data gloves, head-mounted eye trackers or special sensor installations. E.g., in [1], [2], [3] it is shown that through the integration of multi-modal techniques for human-robot-interaction (HRI) simple, robust and reliable handling in real-world scenarios can be guaranteed.

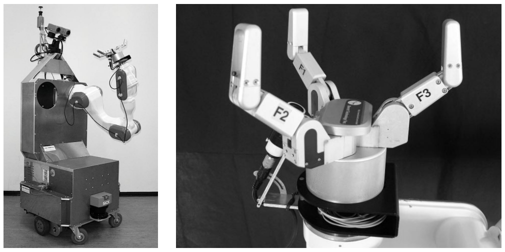
This paper is organized as follows: Section II covers related research. Section III describes the scenario and the setup for the experiments. Section IV presents our new approach to LbD and addresses three different aspects of LbD. First, the design of the interface for HRI is considered. Second, stereoscopic real-time tracking of the demonstrator's hand configuration through Mean-Shift-like tracking [4], [5], [6] based on local binary patterns (LBP) [7] and color histograms is presented. And last, the learning of grasping skills itself by transformation of hand tracking results into three-dimensional self organizing maps (SOMs) [8] and reinforcement learning (RL) [9] is introduced. Experiments are given in Section V. Conclusions and future work are presented in Section VI.

II. RELATED RESEARCH

A lot of work has been done in the field of LbD (see [10] for a recent overview): [11] presents a method for extracting the goals and constraints of a demonstrated task to determine the best imitation strategy, in this case for the selective reproduction of simple reaching tasks. Human trajectories are captured with magnetic field sensors and stereo-cameras and encoded into Hidden Markov Models (HMMs). According to a metric which measures the quality of the reproduction based on invariants in time, an optimal trajectory is generated.

An approach to LbD, which is built on the closed-world assumption and emphasizes the exact observation of the human operator, is presented in [12]. The authors acquire accurate scene data by modeling the whole work cell and using stereo cameras as well as data gloves. In the work cell dual-arm manipulation tasks are demonstrated for humanoid robots [13]. The manipulations are classified by segmentation into subtasks and their mapping and execution on the robot is shown.

As shown in [10], [11], [12], [13] for the reason of robust and accurate data acquisition, data gloves and magnetic sensors are often used to observe the demonstrator's actions. But techniques for contact-free and easy-to-use HRI interfaces are important to simplify the application of LbD techniques. One key problem is the tracking of the human hands, especially for the imitation of manipulation tasks. There are many approaches to this topic, e.g. in [14] an approach to 3D-tracking of



(a) Service robot (b) Grasping tool

Fig. 1. Multi modal service robot and grasping tool

(a) Demonstration phase (b) Grasping phase

Fig. 2. Scenario

a human head and hand is described. It is used to recognize pointing gestures with calibrated stereo cameras. Color and disparity data are integrated into a multi-hypothesis tracking framework, and a HMM is used to classify pointing gestures. Another approach to 3D-hand tracking and articulated finger motion is presented in [15]. A 3D geometric model of a hand is used to generate contours which are matched with monocular edge and skin color images. The tracking is formulated as state estimation, where the model parameters define the internal state of the human hand, which is estimated from image observations. A hierarchical Bayesian filter is developed to allow integration of temporal information.

In [16] a method for imitation learning based on visuo-somatic mapping, ranging from the observed demonstrator's posture to reminding the self posture, is presented. This happens via mapping from the self-motion observation to the self posture of a robot. The mapping from postures to posture space and the mapping from trajectories in posture space onto a motion-segment space is both done by SOMs. Then optical flows from the demonstrator's motions are mapped onto a flow-segment space where flow data is connected with the corresponding motion segments in motion-segment space. The connection with self motion is done via Hebbian Learning.

An abstraction mechanism is used in [17] to focus on features of the demonstration which are important to the imitating robot. It is based on the pairing of multiple inverse models operating in parallel with corresponding forward models to create an execution-prediction sequence and determine which inverse model corresponds to the generation of the observed action. It is shown that recognition with abstraction outperforms recognition without abstraction.

A brief overview of the special field of robot grasping is given in [18]. E.g., [19] presents a self-valuing learning technique based on RL to learn how to grasp unfamiliar objects. Methods for computing hand configurations for precision and pinch grasps are presented in [20].

III. Scenario

The proposed approach is used on a multimodal mobile service robot (see Fig. 1(a)). This robot consists of a mobile platform equipped with a PC and currently one Mitsubishi PA10-6C manipulator. As grasping tool, a three-finger hand from Barrett Technologies Inc. is used (see Fig. 1(b)). The hand has eight degrees of freedom (DOF). Each finger has two joints which are coupled via a TorqueSwitch™ [22]. Two fingers are linked by a spread joint. An active vision system with a stereo-camera system and a pan-tilt unit is mounted on the robot. The robot is able to navigate and move in an office environment. The exact localization capabilities [23] are important for grasping skills, because it allows the robot to adjust its position up to an accuracy of $\pm 1cm$ if it is too far away from the target object to be grasped.

Our scenario for the task of LbD consists of a human demonstrator standing opposite the service robot and a table that is placed between them (see Fig. 2(a)). Several objects are placed on the table. These objects are trained for recognition in advance and offline with a method based on Scale Invariant Features [21]. The user demonstrates a grasping skill by saying "start" and reaching out his hand to the object he wants to grasp. Then he grasps the object and says "stop". The robot observes the performed action several times to collect sufficient data. Then the robot learns the skill and tries to grasp the object itself in the demonstrated way (see Fig. 2(b)).

IV. Learning by demonstration

We designed a multi-modal interface for HRI which uses video and audio data gathered by the robot's multi-camera active vision system and microphones. This passive setup doesn't interfere with the environment besides the robot itself. To track the demonstrator's hands in stereoscopic images in real time, a Expectation-Maximization-like algorithm [6] is adapted. In a completely novel approach this EM-like algorithm is applied to LBP and color histograms. Speech recognition is used to mark start and end points of tracked trajectories. In addition, the hand configuration is retrieved by application of view-based Principal Component Analysis [24]. To actually learn the grasping skills from tracking results we developed a new approach which learns grasping skills by repetitive tracking of the demonstrator's grasping actions and by transforming the tracking results into a three-dimensional SOM. The topology of the SOM is arranged to correspond to the three dimensional space in which the demonstrator's hand is tracked. The hand tracking results are fed into the SOM. After convergence, the SOM gives a spatial description of the collected data and serves as the input data structure for a RL algorithm. The RL algorithm finds trajectories and hand configurations optimized for use by the robot arm and hand. The generated grasp is stored within an object representation

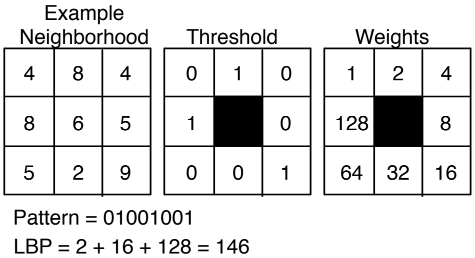
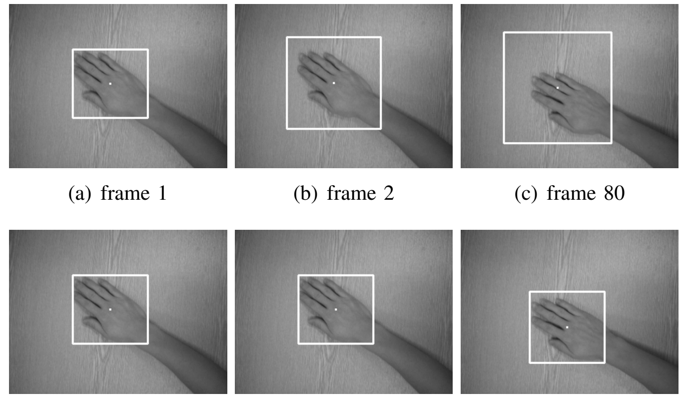


Fig. 3. Example: Local Binary Pattern



(a) frame 1 (b) frame 2 (c) frame 80

(d) frame 1 (e) frame 2 (f) frame 80

Fig. 4. Exemplary comparison of color (Fig. 4(a)-4(c)) and color-texture-based (Fig. 4(d)-4(f)) hand tracking over wooden surface. Color-only-based tracking looses track immediately after initialization whereas color-texture-based tracking does not.

and represents the learned grasping skill.
The approach, its implementation on the multi-model service robot as well as experiments will be presented below.

A. Tracking the demonstrator's hands

To learn a grasping skill by demonstration, it is necessary to know how the demonstrator grasps an object. This means that the trajectory as well as the configuration of the hand, e.g. the articulation of the fingers, has to be known.
For the purpose of position estimation of the hand we rely on an EM-like algorithm [6] which is an extension to Mean-Shift procedures like [4] and [5]. Mean-Shift procedures are efficient techniques for tracking 2D blobs. The EM-like algorithm can robustly track objects based on color-histograms through simultaneous estimation of the position of the local mode and the covariance matrix that describes the approximate shape of the local mode of a kernel-based estimate of a density function. But as for most color-based blob tracking algorithms, robustness decreases if the color feature can not discriminate between the tracked object and background. Therefore we applied LBP as an additional feature to the tracking algorithm. The LBP is a gray-scale-invariant texture analysis operator and derived from a general definition of texture in a local neighborhood. The basic idea is a binary code that describes the local texture which is built by thresholding a neighborhood by the gray value of its center (see Fig. 3). It encodes local primitives like e.g. curved edges, spots, flat areas, etc. Excellent results in terms of accuracy and computational complexity have been shown e.g. by application of LBP to face recognition [25] or moving object detection [26]. Because LBP describes image texture and can also be expressed in the form of histograms, it lends itself to integration into the EM-like tracking algorithm. By combining color and texture features, the robustness of the tracking algorithm is further increased. The improvement in robustness is exemplarily shown in Fig. 4. We used three-dimensional histograms of color and texture discretized to 8 bins per dimension as input for the tracking algorithm. First and second dimensions describe the hue and saturation channels of the HSV color space. The third dimension describes the LBP feature. The main steps of the EM-like tracking algorithm according to [6] are:

Input: object model $\vec{o} = [o_1, \ldots, o_M]^T$(a color-texture histogram of the object with M bins), initial object position $\vec{\theta}^{(0)}$ and shape defined by covariance matrix $V^{(0)}$

1. Compute the M bins r_m of the color-texture histogram of the current region defined by $\vec{\theta}^{(k)}$ and $V^{(k)}$ using
$$r_m(\vec{\theta}^{(k)}, V^{(k)}) = \sum_{i=1}^{N_V} \mathcal{N}(\vec{x}_i; \vec{\theta}, V)\delta\left[b(\vec{x}_i) - m\right]$$
with N_V the number of pixels of the current region, $\mathcal{N}(\vec{x}; \vec{\theta}, V)$ a gaussian probability function, $b(\vec{x}_i)$: $R^2 \to 1, \ldots, M$ a function that assigns a pixel value at location $\vec{x}_i$ to its histogram bin
2. Calculate weights using
$$\omega_i = \sum_{m=1}^{M} \sqrt{\frac{o_m}{r_m(\vec{\theta}^{(k)}, V^{(k)})}}\delta\left[b(\vec{x}_i) - m\right]$$
3. Calculate q_i-s using
$$q_i = \frac{\omega_i \mathcal{N}(x_i; \vec{\theta}^{(k)}, V^{(k)})}{\sum_{i=1}^{N} \omega_i \mathcal{N}(x_i; \vec{\theta}^{(k)}, V^{(k)})}$$
4. Calculate new position estimate $\vec{\theta}^{(k+1)}$ using
$$\vec{\theta}^{(k+1)} = \sum_{i=1}^{N} q_i \vec{x}_i = \frac{\sum_{i=1}^{N} \vec{x}_i \omega_i \mathcal{N}(x_i; \vec{\theta}^{(k)}, V^{(k)})}{\sum_{i=1}^{N} \omega_i \mathcal{N}(x_i; \vec{\theta}^{(k)}, V^{(k)})}$$
5. Calculate new variance estimate (shape) $V^{(k+1)}$ using
$$\vec{V}^{k+1} = \beta \sum_{i=1}^{N} q_i (\vec{x}_i - \vec{\theta}^{(k)})(\vec{x}_i - \vec{\theta}^{(k)})^T$$
6. If no new pixels are introduced using the new elliptical region stop, otherwise set $k \leftarrow k+1$ and go to 1

Tracking of the user's hand is done in both stereo images. To compute the hand's world coordinates, the stereo correspondence problem is solved by approximation of the center points of both tracked hand regions as the corresponding points. The tracking results are fed into a particle filter to compensate for the rough approximation of the hand's position in 3D space. Initialization of the tracking algorithm is done by skin color detection.

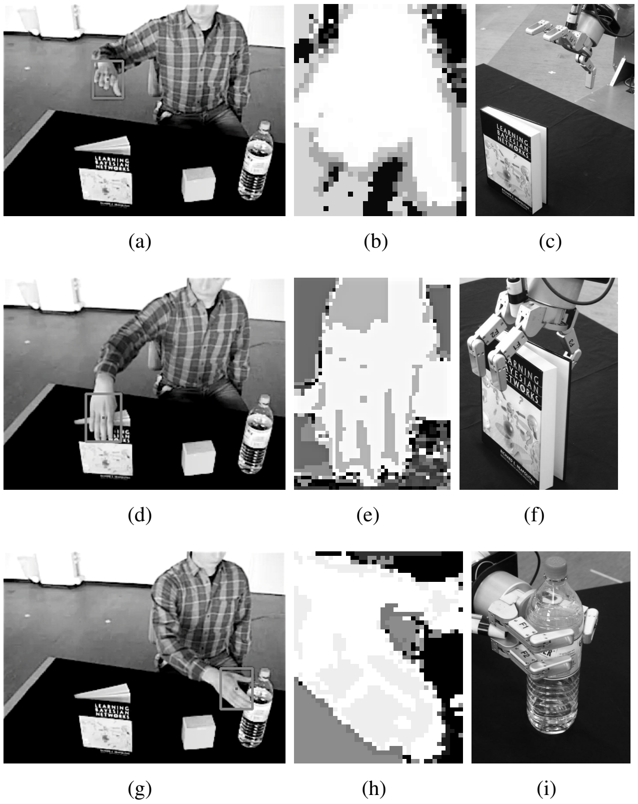


Fig. 5. Examples of the robot's hand configuration selection: The tracked hand performing different grasps is shown in Fig.5(a), Fig.5(d) and Fig.5(g). The weights from step 2 of the tracking algorithm are shown in Fig.5(b), Fig.5(e) and Fig.5(h). They are used as input to the PCA of the shape of the human hand. Classification of the weights leads to the Barrett-Hand configurations as shown in Fig.5(c), 5(f), 5(i)

B. Estimation of the hand posture

Tracking of articulated finger motion in 3D space is a high-dimensional problem. Over 25 DOF can be concerned, and because of the projection of a three-dimensional scene onto a two-dimensional image, a lot of information is lost. The non-rigidity of the human hand and the great number of DOF result in deformations of the human hand during grasping actions and motion in general. But modeling the whole hand with complete articulated finger motion is not necessary for the case of grasping objects. We are only interested in the type of grasp which is used to pick up an object. Therefore we decided to do a view-based approach to detect the posture of the demonstrator's hand. This approach is inspired by the eigenface method [24] and is based on the well-known Principal Component Analysis (PCA). But we use the weights of tracking step number 2 (see Fig.5(e), 5(h) and 5(b)) as input for the PCA, instead of just taking the raw image of the tracked hand region. As can be seen, the weights provide a good segmentation into fore- and background and therefore the PCA yields better classification results using the weights as input. Additionally, the tracking makes normalization of the sample size easy because it gives the exact location and shape of the tracked object. The corresponding mapping from classified hand postures to Barrett configurations is selected offline in advance.

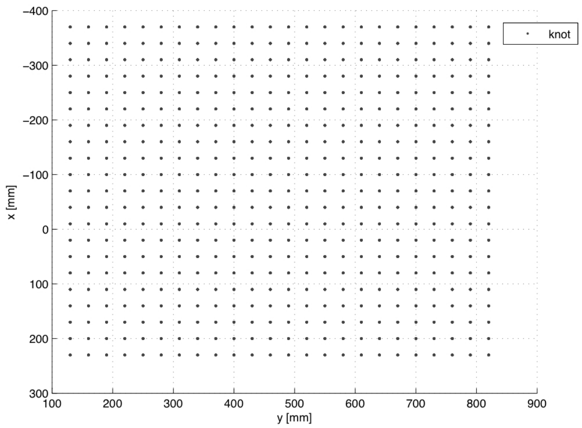
C. Learning the grasping skill

In contrast to robot arms, the repeat accuracy of the human arm is very low. Each time the demonstrator presents a grasping skill, it will be performed differently. Because of tracking and calibration errors, the discrepancies between intended and tracked trajectories are further amplified.

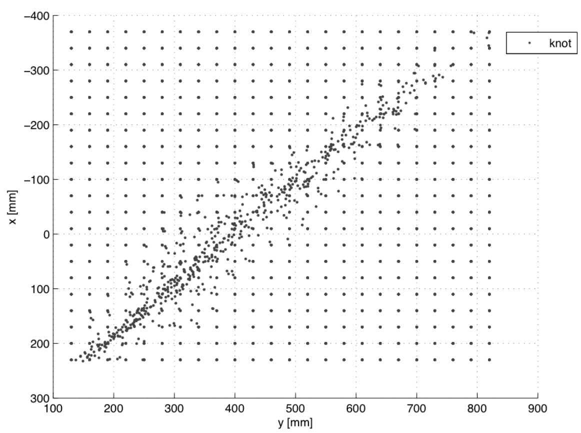
Our approach uses an RL-technique to learn the intended trajectory of the grasping skill. First, the tracking data has to be transferred into a suitable representation for an RL-agent. Therefore, the space used for hand tracking has to be discretized. Each node of the grid corresponds to one discretized point of the input space and therefore represents one state of the RL-agent. Additionally, a state evaluation function which evaluates the quality of the current state is needed. Because the quality of a state corresponds to the amount of trajectories going through it, the evaluation function assigns higher values to states with more trajectories going through. We propose the use of SOMs to generate the state evaluation function. A SOM is organized according to the regular three-dimensional grid world which serves as data structure for the RL-agent. Each node represents one point in the real-world and is connected to its six nearest neighbors (see Fig.6(a)). The collected trajectories are discretized and each point of the discretized trajectories is used as input vector for the SOM. For each input vector the nearest node of the map is searched. This node and its six neighbors (defined by the topology of the SOM) are shifted towards the input vector. Whereas the learning rule says that the smaller a node's distance to the input vector is, the more it is shifted. This leads, after entering all trajectories, to regions with different densities of nodes on the map (see Fig.6(b)). Whereas nodes with lower distances to their neighbors represent nodes with more trajectories going through and nodes with higher distances to their neighbors represent nodes where lesser trajectories going through, respectively. Because the topology of the SOM is preserved during the learning phase, a value of quality can be assigned to each corresponding state of the grid world where higher values of quality are assigned to nodes with low distances to their neighbors.After learning the evaluation function by a SOM, the RL-agent learns the intended trajectory of the grasping skill. The used RL-algorithm is a Q-Learning algorithm [9]. The states for the Q-Learning are the discretized states from the SOMs. The update rule for the Q-Values of each state is given by:

$$Q_{t+1}(s,a) = \sum_{s^{'}} \mathcal{P}^{a}_{ss'} \left(\mathcal{R}^{a}_{ss'} + \gamma \cdot Q_t(s^{'}, a^{'}) \right)$$

Where s denotes the state and a the action. $\mathcal{P}^{a}_{ss'}$ is the probability to get from state s to state $s^{'}$ with action a, and $\mathcal{P}^{a}_{ss'}$ is the given reward according to the selected action and states. γ denotes the learning rate. At this point, the advantage of this method over simple regression becomes clear. The RL-Agent can be set up with different constraints or strategies to



(a) initial SOM with regular distributed knots



(b) final SOM with concentration of knots along the area where data was entered

Fig. 6. Initial and final SOM of book grasping skill (see Fig. 7). For better visualization the knots of the SOM are projected on the xy-plane (ground plane) and the links between the knots are left out)

simulate the movement of the robot arm. E.g., it can optimize time or path length. Or even further: impossible robot arm movements can be excluded from the outset by concerning inverse kinematics. Only the reward function $\mathcal{R}^{a}_{ss'}$ is modified to get different results.

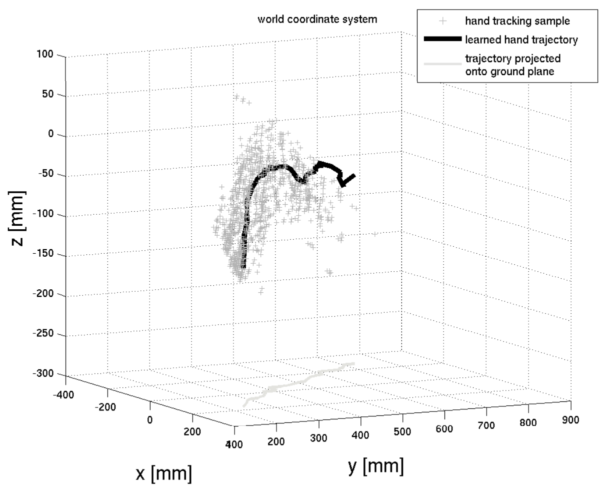
V. Experiments

In the experiments three different objects were grasped. We chose a bottle, a box and a book representing many other objects. The demonstrator grasped each object ten times while being observed by the robot. The robot recorded three sequences, one for each object, of about 1600 vectors of world coordinates. Each sequence was used to train one three-dimensional SOM.

Afterwards the Q-Learning was trained. The average dimension was approximately about 6000 states. To generate a trajectory from the input data, the Q-Learning algorithm was trained by 1000 iterations. The generated sequence through state space represents the learned trajectory. The learned trajectory for the book is shown in Fig.7, for the box in Fig.8 and for the bottle in Fig.9. The input vectors are overlaid in

(a) frame 12

(b) frame 76



(c)

Fig. 7. Hand tracking results and generated trajectory of book grasping skill

these plots to visualize which path through the input data is chosen by the RL-Agent. As the visualization shows, the RL-Agent generates a trajectory which lies within the recorded movement. For grasping the book the trajectory is curved, for the box it is descending and for the bottle it is ascending according to height. All trajectories are nearly a straight line projected on the xy-plane. The trajectory is optimal according to the length of the movement and the number of visits of a point in the grid.

VI. Conclusion

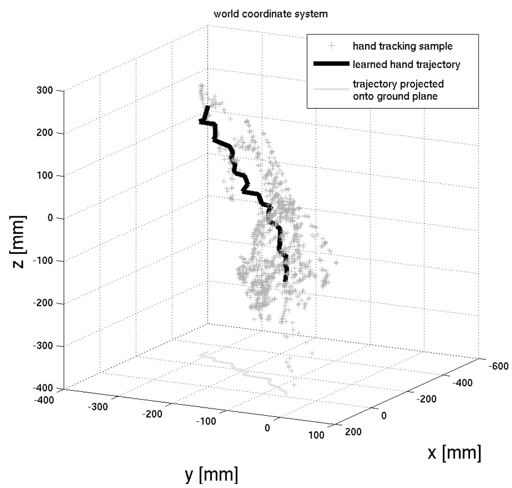
The presented approach describes the usefulness of LbD in the context of human-robot-interaction. It is vision- and audio-based, therefore contact-free and usable without further special hardware like data gloves, etc.

Through the combination of local binary features and color histograms, the robot tracks the demonstrator's hand in real-time. The use of SOMs as an unsupervised learning method in combination with reinforcement learning lets the robot learn the demonstrated grasping trajectories. Both contributions to LbD prove to be robust and efficient as experiments show.

The next step will be to add the ability of learning object handling to the system. Examples of this are pouring water from a bottle into a glass or hand over operations. The integration of new vision sensors with higher resolution and incorporation of more multi-modal sensors are also further research aspects.

(a) frame 560

(b) frame 812

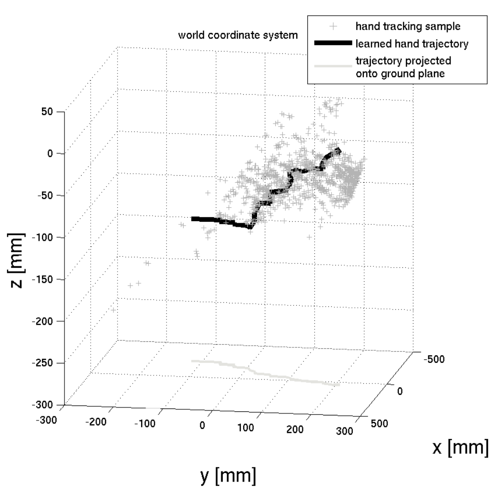


(c)

Fig. 8. Hand tracking results and generated trajectory of box grasping skill

(a) frame 331

(b) frame 769



(c)

Fig. 9. Hand tracking results and generated trajectory of bottle grasping skill

REFERENCES

[1] T. Baier, M. Hüser, J. Zhang, *Integration of Face and Gesture Detection in Natural Language Instructing of Robot in an Assembly Scenario*. Proc. of 11th IEEE Int. Workshop on Robot and Human Interactive Communication ROMAN 2002, Berlin, Sep. 25-27, 2002, pp. 241-246

[2] T. Baier, M. Hüser, J. Zhang, *Instructing an assembly robot in situated natural language and gestures*. Proc. of the 10th Int. Conference on Human-Computer Interaction, Heraklion, Crete, June 2003

[3] T. Baier, M. Hüser, J. Zhang *A multimodal interface to situated assembly robot systems*. Proc. of the IEEE Int. Conference on Robotics, Intelligent Systems and Signal Processing, Changsha, China, 2003

[4] D. Comaniciu, V. Ramesh, P. Meer, *Real-Time Tracking of Non-Rigid Objects using Mean Shift, To appear*. IEEE Conf. on Comp. Vis. and Pat. Rec., Hilton Head Island, South Carolina, 2000.

[5] R. Collins, *Mean-shift blob tracking through scale space*. Proc. IEEE Conf. on Comp. Vis. and Pat. Rec., 2003

[6] Z. Zivkovic, B. Kröse, *An EM-like algorithm for color-histogram-based object tracking*. IEEE Conf. on Comp. Vis. and Pat. Rec., 2004

[7] T. Mäenpää, M. Pietikäinen, *Texture analysis with local binary patterns*. In: Chen CH & Wang PSP (eds) Handbook of Pattern Recognition & Computer Vision, 3rd ed, World Scientific, Singapore

[8] T. Kohonen, *Self-Organizing Maps*. Springer, Berlin, Germany, 1997

[9] R. S. Sutton, A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press, 1998

[10] A. Billard, R. Siegwart, *Robot Learning from Demonstration*. Robotics and Autonomous Systems, Vol.47, Issue 2 & 3, 30 June 2004

[11] S. Calinon, F. Guenter, A. Billard, *Goal-Directed Imitation in a Humanoid Robot*. ICRA, Barcelon, April 2005.

[12] M. Ehrenmann, R. Zöllner, O. Rogalla, S. Vacek, R. Dillmann, *Observation in Programming by Demonstration: Training and Execution Environment*, Third IEEE Int. Conference on Humanoid Robots, October 2003, Karlsruhe, Germany

[13] R. Zöllner,T. Asfour, R. Dillmann, *Programming by Demonstration: Dual-Arm Manipulation Tasks for Humanoid Robots*, Proc. IROS, pp.479-488, Sendai, Japan, 2004,

[14] K.Nickel, E.Seemann, R.Stiefelhagen, *3D-Tracking of Heads and Hands for Pointing Gesture Recognition in a Human-Robot Interaction Scenario*. 6th Int. Conf. on Face and Gesture Recognition - FG 2004, Seoul, Korea

[15] B. Stenger, DPhil Thesis: *Model-Based Hand Tracking Using A Hierarchical Bayesian Filter*. Department of Engineering, University of Cambridge, March 2004.

[16] M. Asada, M. Ogino, S. Matsuyama, J. Ooga, *Imitation Learning Based on Visuo-Somatic Mapping*. Proc. of 9th Int. Symposium on Experimental Robotics 2004

[17] M. Johnson, Y. Demiris, *Abstraction in Recognition to Solve the Correspondence Problem for Robot Imitation*. In Proc. of TAROS 2004, pp. 63-70, Essex, 2004

[18] A. Bicchi, V. Kumar, *Robotic grasping and contact: a review*. in: Proceedings of the IEEE Int. Conference on Robotics and Automation, 2000, pp. 348-353

[19] J. Zhang, B. Rössler *Self-valuing learning and generalization with application in visually guided grasping of complex objects*. Journal of Robotics and Autonomous Systems, 47: 117–127, 2004

[20] Ch. Borst, M. Fischer, G. Hirzinger *Calculating hand configurations for precision and pinch grasps*. Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 2002

[21] D. G. Lowe, *Distinctive image features from scale-invariant keypoints*. Int. Journal of Computer Vision, 60, 2 (2004), pp. 91-110

[22] W. T. Townsend, *The BarrettHand Grasper - programmably flexible part handling and assembly*. Industrial Robot: An international Journal, Vol. 27, Nr. 3, 2000, pp.181-188

[23] A. Schneider, D. Westhoff, *Autonomous Navigation and Control of a Mobile Robot in a Cell Culture Laboratory*. Diploma Thesis, Bielefeld, Germany, 2002

[24] M. Turk, A. Pentland, *Eigenfaces for recognition*. J. Cog. Neuro., Vol.3, No.1, pp.71-86, 1991

[25] T. Ahonen,A. Hadid,M. Pietikinen, *Face recognition with local binary patterns*. Computer Vision, ECCV 2004 Proc., Lecture Notes in Computer Science 3021, Springer, 469-481.

[26] M. Heikkil, M. Pietikinen, J. Heikkil, *A texture-based method for detecting moving objects*. Proc. the 15th British Machine Vision Conference (BMVC 2004), London, UK, 1:187-196.