

“iCub, clean the table!” A robot learning from demonstration approach using Deep Neural Networks

Jaeseok Kim¹, Nino CAuli², Pedro Vicente², Bruno Damas^{2,3}, Filippo Cavallo¹ and José Santos-Victor²

Abstract—Autonomous service robots have become a key research topic in robotics, particularly for household chores. A typical home scenario is highly unconstrained and a service robot needs to adapt constantly to new situations. In this paper, we address the problem of autonomous cleaning tasks in uncontrolled environments. In our approach, a human instructor uses kinesthetic demonstrations to teach a robot how to perform different cleaning tasks on a table. Then, we use Task Parametrized Gaussian Mixture Models (TP-GMMs) to encode the demonstrations variability, while providing appropriate generalization abilities. TP-GMMs extend Gaussian Mixture Models with an auxiliary set of reference frames, in order to extrapolate the demonstrations to different task parameters such as movement locations, amplitude or orientations. However, the reference frames (that parametrize TP-GMMs) can be very difficult to extract in practice, as it may require segmenting the cluttered images of the working table-top. Instead, in this work the reference frames are automatically extracted from robot camera images, using a deep neural network that was trained during human demonstrations of a cleaning task. This approach has two main benefits: (i) it takes the human completely out of the loop while performing complex cleaning tasks; and (ii) the network is able to identify the specific task to be performed directly from image data, thus also enabling automatic task selection from a set of previously demonstrated tasks. The system was implemented on the iCub humanoid robot. During the tests, the robot was able to successfully clean a table with two different types of dirt (wiping a marker’s scribble or sweeping clusters of lentils).

I. INTRODUCTION

Nowadays an increasing number of human resources is required to guarantee an adequate support to a growing elderly population and, as a consequence, autonomous service robots capable of responding to these demands are becoming a key research topic in the robotics field [1]. Even if the social interaction abilities of service robots are reaching a mature stage [2], [3], there is still a long way to go with respect to the physical interaction and manipulation abilities of such robots [4], [5]. There are many categories of household chores, such as doing the laundry or preparing meals: in this paper we take a closer look at cleaning tasks and how to perform them in a fully autonomous way using the iCub robot [6] (Fig. 1) to demonstrate our main results.

There is much research work in robotic cleaning tasks: Okada *et al.*, for instance, apply an inverse kinematics based programming approach to compute whole-body motions for

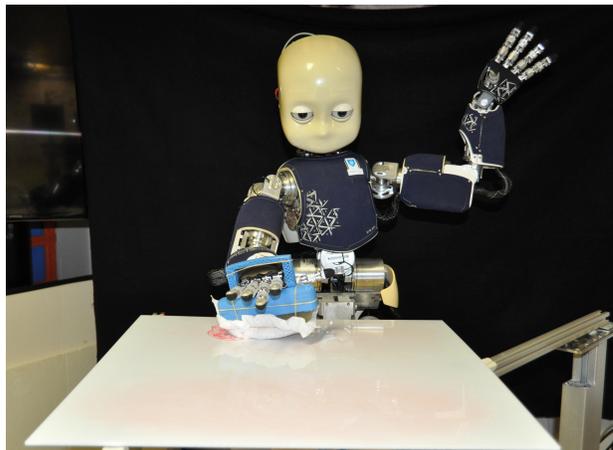


Fig. 1: The iCub humanoid robot using a tool to clean the table in front of him.

the tasks of dish washing, sweeping and vacuuming the floor using a humanoid robot [7], [8]. Kunze *et al.* have a simulation-based approach for everyday manipulation using a temporal projection system [9] and apply their architecture to a table sponge wiping task. In [10] a null-space optimization approach generates cleaning trajectories efficiently in terms of manipulation effort and completion time, while keeping the robot body fixed. Leidner *et al.* propose whole-body control strategies for wiping tasks based on human task knowledge [11], parameterized by high-level hybrid reasoning mechanisms that determine relevant control parameters. The same authors present the application of a coverage path planning algorithm to a wiping task that is divided in sub-tasks such as absorbing, collecting and skimming [12].

While such approaches can succeed in highly controlled environments, the presence of disturbances or uncertainty usually requires some sort of adaptive behavior. If we want a robot to interact with people in a (at least) partially unknown environment we must ensure some robustness and adaptability to the variability of situations and perceptions presented to the robot in an actual scenario. With this learning setting in mind, Inéz *et al.* present a Markov Decision Process to search actions for cleaning, based on dirty areas, using a learning heuristic, based on a fixed set of rules, to find optimized actions [13]. However, using such fixed, pre-defined set of rules limits the applicability of the method to very simple tasks. Resorting to the reinforcement learning (RL) framework provides a more flexible approach to the cleaning problem: the work of Cruz *et al.* is such an example,

¹BioRobotics Institute, Scuola Superiore Sant’Anna, Pisa, Italy. j.kim@sssup.it, filippo.cavallo@santannapisa.it

²Institute for Systems and Robotics, Instituto Superior Tecnico, Universidade de Lisboa, Portugal {ncauli, pvicente, bdamas, jasv}@isr.tecnico.ulisboa.pt

³CINAV — Centro de Investigação Naval, Almada, Portugal

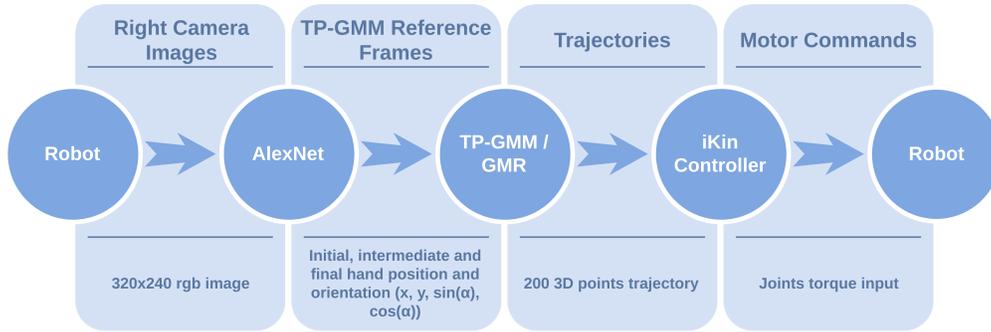


Fig. 2: System Architecture. Each circle represents a module of the overall architecture and the bottom of the figure shows the data flow between modules.

where an interactive RL approach for removing objects on a table is presented, using object affordances [14].

The major problem with such RL schemes is the slow convergence rate and the need for many training episodes to achieve a high success rate: this is a critical issue in real-life, non-simulated scenarios. The Learning from Demonstration (LfD) paradigm is an alternative to a pure RL scheme, where the human directly presents to the robot some correct demonstrations for the task to fulfill, that can eventually be used to provide a good starting point to an RL scheme. In this context, Kormushev *et al.* present a method to clean a surface by kinesthetic teaching [15]: however, this pure imitation learning scheme is only able to mimic previously seen movements.

To adapt the cleaning task to flexible situations a task-parameterized Gaussian mixture model framework is proposed in [16]. Alizadeh *et al.* develop a system able to solve a dust sweeping task through kinesthetic teaching with the task-parameterized Gaussian mixture model (TP-GMM) framework and Gaussian Mixture Regression (GMR) [17]. The authors define a set of reference frames consisting of the starting position and orientation of the hand, the position and orientation of the objects to swipe, and the end position and orientation. After learning, the model is able to generalize to unseen movements, characterized by different reference frames defined by the locations of the objects that serve as markers. It is also shown in that paper the robustness of the proposed method to the absence of some of those reference frames while performing the task.

This paper builds upon the previous work of Alizadeh *et al.* [17]: while their architecture is flexible enough to be able to generalize to unseen situations, it relies heavily on assigning reference frames that implicitly define the movement to perform. However, in a real-life scenario it is not reasonable to rely on the physical presence of such markers for tasks like wiping and sweeping, where the distribution of dirt in the floor or the table turns this task into an ill-posed problem, as this requires a strong human intervention. We developed a learning architecture that is able to encode and reproduce human demonstrations for a given task, using the TP-GMM framework, without the need to use markers of any kind to signal the task reference frames.

Given a set of demonstrations, associated with a given task, two deep neural networks learn to assign such reference frames to the images acquired from the cameras installed in the robot head. In our setup, these frames correspond to initial, intermediate and final positions and orientations of the provided demonstrations, and are calculated automatically as a post-processing step of the demonstrated data. Also, we show that the networks are able to discriminate between tasks given solely the raw data from the camera after training, *i.e.*, they can associate an image depicting specific types of dirt to corresponding tasks like wiping or sweeping. The robot, controlled by this deep learning architecture, performs the cleaning task autonomously, without human intervention. We provide an experimental validation for this architecture. The data acquired and recorded during task demonstrations using the iCub humanoid robot was gathered in a dataset that is publicly available (see subsection III-C).

The paper is organized as follows. Section II summarizes the proposed approach for LfD using the TP-GMM and deep neural networks. Section III presents the experimental setup, while in Section IV the experimental results obtained using this setup are presented. Section V concludes the paper and presents some directions for further research.

II. PROPOSED APPROACH

In this paper, we have designed and implemented a system to control a robot for cleaning a planar surface placed in front of it (*e.g.*, wiping a marker’s scribble or sweeping lentils from a table). The robot uses vision (an RGB camera) to distinguish two types of dirt (*i.e.* scribbles and lentils) and perform the correct action to achieve a clean workspace (*i.e.*, wiping or sweeping respectively).

The control system consists of three different modules: i) two convolutional neural networks (CNNs) based on AlexNet [18]; ii) a TP-GMM/GMR block; and iii) the iCub kinematic controller [19] (see Fig. 2 for more details on the system architecture). The two networks predict the TP-GMM parameters from a 320x240 RGB camera’s image. The TP-GMM parameters are defined as the 2D positions (\mathbf{b}) and orientation (\mathbf{A}) of the trajectory at the initial, intermediate and final points:

$$\mathbf{X}_j = \{\mathbf{A}_j, \mathbf{b}_j\} \quad (1)$$

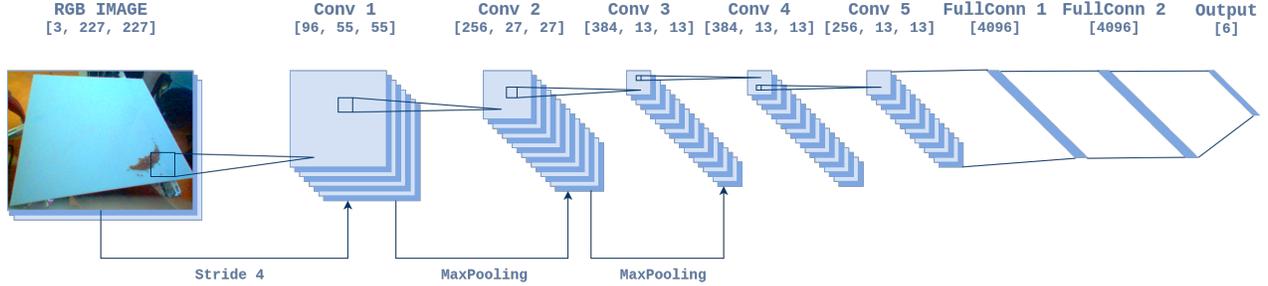


Fig. 3: Architecture of the two networks used to predict the reference frames (initial, intermediate and final positions and orientations). The 6 output for are the x and y positions for the first network and sine and cosine values of the trajectory orientation for the second one.

The reference frame \mathbf{X}_j is represented in the robot reference frame. The outputs of the first network are the 2D positions of the reference frames, while the outputs of the second network are their orientation. Based on predicted TP-GMM parameters, the TP-GMM/GMR module generates the wiping/sweeping hand trajectory composed by 200 3D points. Starting from the generated trajectory, the iCub Cartesian interface (iKin controller) [20] solves the inverse kinematics and sends the motor commands to the robot.

A. Deep neural networks

Two CNNs were implemented in order to learn the task parameters for the Gaussian mixture model directly from the image pixels. In our case, as said before in Eq 1, the TP-GMM parameters are the three reference frames (positions and orientation). The CNNs architecture was devised based on the AlexNet [18] model changing only the output layer. In the proposed networks, the 1000 nodes output layer of the AlexNet is replaced with a fully connected one with 6 nodes. The outputs of the first network are the x and y Cartesian coordinates of the three reference frames, while the second network outputs the trajectory orientation on those points (represented as sine and cosine). In total, we have 6 values for the reference frames positions and 6 values for the respective orientation. The networks take as input a 3 channel (RGB) image resized to a dimension of 227x227 pixels. The networks have a total of 8 layers: 5 convolutional layers and 3 feedforward fully connected layers. To have a more detailed description of the network structure please refer to Fig. 3. In order to train the networks we generated a dataset of images and trajectories using 1000 kinesthetic demonstrations of wiping and sweeping movement. The reader can refer to Section III for a detailed description of this dataset. The networks were trained through supervised learning: the input is the first image recorded at the beginning of a sweeping or wiping movement while the output are three reference frames: initial, intermediate and final. The reference frames were calculated after a postprocessing phase on the raw trajectories.

B. Task parameterized Gaussian mixture model

Gaussian Mixture Models provide a convenient and compact way to encode a set of human demonstrations of a particular task or skill, and have been extensively used in the recent years [16], [17], [21]. Given a set of trajectories, corresponding to M human demonstrations of the task to be learned, a GMM can be trained over the complete set of data points acquired during the demonstrations and later provide an average trajectory estimate for the task via Gaussian Mixture Regression, that takes into account the natural variations in human demonstrations. However, to be able to extrapolate the learned skill to other regions of the task space, or to be able to modulate such skill (*e.g.* changing the amplitude of the motion), one has to rely on task parameters that implicitly define the properties of the task, such as auxiliary frames of reference representing initial, intermediate and final points for the movement [22].

This paper builds upon the work of [16] and employ a task parametrized GMM that can extrapolate a trajectory from a set of human demonstrations. In this framework, each demonstration m from a set of M demonstrations consists of a set of T_m samples forming a whole dataset of $\{\xi\}_n$ data points, with $1 \leq n \leq N$ and $N = \sum_m T_m$. Each sample ξ_n is a $D + 1$ dimensional vector containing time t_n and the observed task space variables at that time, and is associated with the observed task parameters $\mathbf{X}_j = \{\mathbf{A}_{m,j}, \mathbf{b}_{m,j}\}$, $1 \leq j \leq P$, that represent the P fixed frames of reference \mathbf{X}_j (rotation matrix $\mathbf{A}_{m,j}$ and origin $\mathbf{b}_{m,j}$) that define the task for each demonstration m .

The parameters to be learned from the data are $\{\pi_i, \mathbf{Z}_{i,j}^\mu, \mathbf{Z}_{i,j}^\Sigma\}$, with $1 \leq i \leq K$, corresponding to weights, means and covariances of a set of P GMMs with K components each, that describe the demonstrated trajectories in each frame of reference j . Given this set of P GMMs, the likelihood of a data point $\{\xi\}_n$ is given by the product of likelihoods in each frame of reference, *i.e.*,

$$p(\xi_n | \cdot) \sim \prod_{j=1}^P \mathcal{N}(\mathbf{A}_{m,j} \mathbf{Z}_{i,j}^\mu + \mathbf{b}_{m,j}, \mathbf{A}_{m,j} \mathbf{Z}_{i,j}^\Sigma \mathbf{A}_{m,j}^T),$$

where each GMM is projected to a global frame of reference and ξ_n comes from demonstration m . For each frame, the

mixture component i that effectively generates the observed data point ξ_n is unknown, and so determining the parameter values that maximize the likelihood of the demonstrated data $\{\xi\}_n$ must resort to an Expectation-Maximization iterative procedure. The E-Step estimates $\gamma_{n,i}$, the probability that mixture component i is responsible for generating training data n , while the M-Step calculates the new values for the parameters given these estimates for the latent variables. For details concerning the model fitting procedure please refer to the original paper [16].

The well known GMR technique is used to reproduce a task after fitting the task parameterized GMM using the Expectation-Maximization (EM) algorithm, where for each time t_n the corresponding task space variables are estimated by marginalizing the model w.r.t. mixture components and reference frames. Once again, please refer to the original paper for details. The interesting point here is that in this process different weights are assigned to different frames of reference, according to the current time of the reproduction, thus effectively capturing the most relevant features of the human demonstrations. These correspond to some invariance of the demonstrations as seen from each frame of reference, encoded in a low variance estimate for the task space variables, taken from the corresponding GMM.

III. EXPERIMENTAL SETUP

The experiments were conducted on a laboratory environment and divided into two sets: i) the *wiping* scene and ii) the *sweeping* scene. The first scene corresponds to a white table tainted with a red marker. The robot should clean it using a special white board eraser (the tool) attached to its right hand. The tool consist of a sponge with dimensions 6x13x5 cm (height, width and depth, respectively). Due to iCub grasping limitations (*i.e.*, lack of motor strength to hold the sponge while applying pressure on the table), the sponge is fixed to the robot hand using a 3D printed support (see Fig. 1). In the *sweeping* scene, the same white table is installed in front of the robot holding the same tool. However, the dirt is simulated as several dry lentils that should be swept towards the gap between the robot and the table, where a garbage box is located. In both scenarios, and during the creation of a dataset of examples, the robot is initialized at a home position and a human instructor teaches it how to clean the dirt using zero torque controllers¹ on the arm, without specifying explicitly the type of dirt in front of the robot.

The whole trajectory is saved during the kinesthetic teaching experiment and a post-processing was done to extract the reference frames (Eq. 1) in order to be able to generalize to different locations of the workspace through the task-parametrized GMM. The image of the right camera is saved before starting the cleaning motion. Moreover, the 6D Cartesian pose (position and orientation) of the robot end-effector during the whole movement is recorded exploiting the iKin library [20].

¹We used the "Force Control" iCub module. The code can be found at <https://github.com/robotology/icub-basic-demos/tree/master/demoForceControl>

A. iCub description

The iCub humanoid robot [6] was developed in the context of the EU project RobotCub (2004-2010) and subsequently adopted by more than 30 laboratories worldwide. It has 53 motors that move the hands, arms, head, waist and legs and it has the average size of a 3-year-old child. Its degrees of freedom and human-like appearance are important characteristics that enable the study of human-robot interaction and autonomy in humanoid robots. The stereo vision system (cameras in the eyeballs), proprioception (motor encoders), touch (tactile fingertips and artificial skin) and vestibular sensing (IMU on top of the head) are major features that allow the implementation of the proposed methodology.

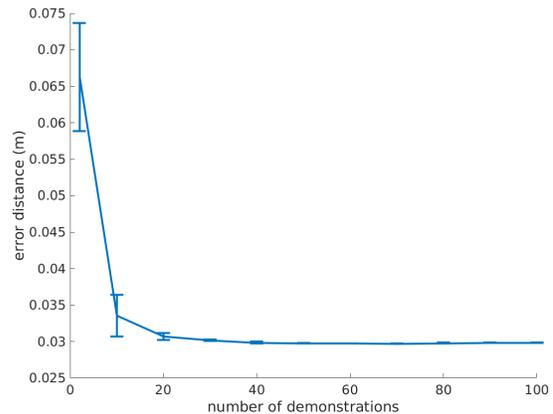


Fig. 4: Variation of the error between GMR generated trajectories and kinesthetic ones, with the number of demonstrations presented to the GMM.

B. System initialization

The iCub is initialized at zero position in the torso joints, with the arm above the head (*i.e.*, unobserved by the cameras) and gazing the table with the workspace totally visible on right camera image. The table is fixed during the whole experiment at a distance of 15 cm in front of the robot and 3 cm below the robot reference frame².

The first reference frame X_1 is acquired when the contact between the tool and the table is attained. Furthermore, the last reference frame X_3 is calculated as the final point in which the hand was in contact with the table. The intermediate frame X_2 is calculated as the median point (in time) between the starting (*i.e.*, the first reference frame) and finishing (*i.e.*, the last reference frame) the movement. In addition, the orientation of the reference frames are computed as the orientation vector between each point and their next adjacent point on the trajectory. These reference frames (position and orientation) are computed for labelling purposes.

²The reader can check <http://wiki.icub.org/wiki/ICubForwardKinematics> for a full description of the iCub reference frames

The networks were trained on a sub-dataset of 700 images while 300 images were used as test set. The networks' supervisory signal consists of 6 labels each (2D position of the reference frames for the first network and their 2D orientation for the second one). The labels are normalized to reduce the difference in dimensionality between the outputs. Also, mean RGB values extracted from the training set images are subtracted to the input. The learning rate was set to 0.001 and the momentum to 0.9. We stopped the training process after 230000 epochs. In order to prevent overfitting, we applied a dropout with ratio 0.5 to the first two fully connected layers of the networks. The networks were implemented using Caffe [23].

The initialization of the TP-GMM consists in defining the reference frames structure and number. In our case the reference frames are 4: the iCub reference frame is the base reference frame and the 2D position and orientation of the initial, intermediate and final points are the remaining three as defined in Eq. (1). The TP-GMM/GMR will generate trajectories starting from the initial point, passing through the intermediate point and ending on the final one. In order to define the number of demonstrations to initialize the TP-GMM we divided the 700 elements of the training set in a 500 for training and 200 for validation. We initialized the TP-GMM on different numbers of random demonstrations and we tested the results on the validation set. Fig. 4 depicts the mean and standard deviation on 10 trials of the error between GMR generated trajectories and kinesthetic ones as they vary in the number of demonstrations. Based on the results of the plot, we set at 70 the number of demonstrations used to initialize the TP-GMM during our experiments. The number of Gaussian components was calculated empirically and set to 5.

C. Dataset

The networks were trained on a dataset created performing kinesthetic demonstrations on the iCub humanoid robot. During each demonstration the robot stands in front of a table with either marker's scribbles or clusters of lentils on top of it. A user performs a cleaning action (wiping for the scribbles and sweeping for the lentils) grabbing the robot hand and moving its right arm using a compliant control mode. The dataset is composed with 1000 cleaning demonstrations (500 for wiping off a scribble and 500 for sweeping the lentils). Each demonstration contains the initial camera image (iCub right camera³) and its respective reference frames.

The dataset used on the preparation of this manuscript is public⁴ in order to be possible to replicate the results achieved, and to foster the comparison with other methodologies on cleaning tasks performed by robots.

IV. RESULTS

In this section we analyze some results obtained using the proposed architecture. The results section is divided in two

³RGB images with resolution of 320x240

⁴the dataset is available on the VisLab webpage: <http://vislab.isr.ist.utl.pt/datasets/>

parts: detailed analysis of performances on the test set and early tests of the whole architecture on the real robot.

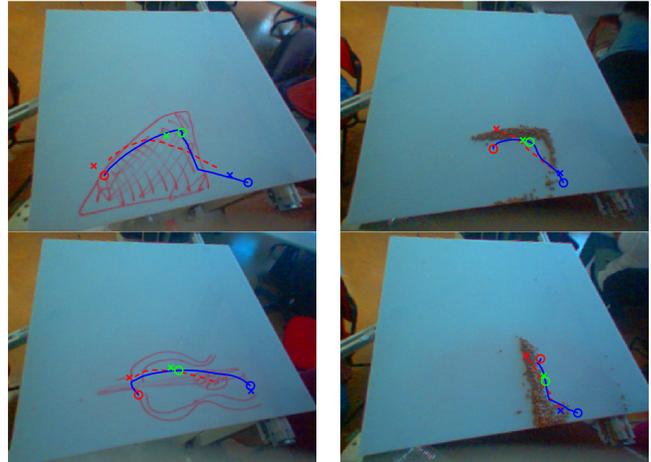


Fig. 5: Examples of system outputs for training (upper row) and test examples (lower row) projected on the input images. Circles are train/test labels (red:initial, green:intermediate, blue:final), crosses are positions network outputs, solid lines are trajectories from demonstration and dashed lines are trajectories generated by the GMR.

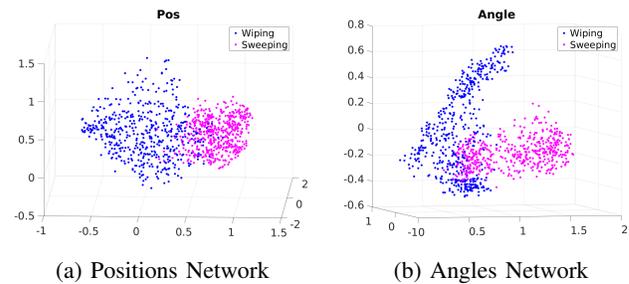


Fig. 6: PCA 3D space projection of 'FullConn 2' layers for positions network (a) and angles network (b). Blue dots represent wiping actions while purple dots represent sweeping actions. Best seen in color.

A. Test set results

We evaluated the system on the 300 images test set extracted from our dataset (see subsection III-C). The Absolute Mean Error (AME) between the test trajectories obtained through kinesthetic demonstrations and the trajectories predicted by the GMR is around 3.3 cm. Taking into account the length of the trajectories in the demonstrations (which have a mean value of 22.4 cm), the system is able to generate a path with a small error and close to the desired ones.

The results on the two CNNs can be seen in Table I which shows the AME between the test reference frames (initial, intermediate and final) and the output of each network. The error of the first network (position) evaluated on the test set is about 5.66 cm. Although the error appears to be high, we need to take in consideration the variability of the kinesthetic demonstrations of our dataset. In Fig. 5 it



Fig. 7: Examples of system outputs during the real robot experiments projected on the input images. The first row shows a sweeping example while the second row shows a wiping one. Each column shows the state of the table after the previous cleaning action and the trajectory that will be executed next. Circles are positions network outputs (red:initial, green:intermediate, blue:final) and dashed lines are trajectories generated by the GMR.

is possible to notice how different are the demonstration trajectories for the same type of actions (*i.e.*, the first images of both rows representing a wiping movement). The networks were able to generalize well despite the error. The angles network has an AME of 0.7621 radians. The high value is mainly due to the error on the orientation at the initial position (1.3234 radians). Analyzing the dataset we noticed that the demonstrators tend to change substantially the initial direction of the movements. This is a ill-posed problem and it is not reasonable to expect accurate predictions to almost random human movement starting positions. Fig. 5 shows how this error does not have a significant effect on the generation of the trajectory performed by the GMR module. In fact, the GMR is able to generate a valid trajectory, with an error of 3.3 cm (around 14.7% of the mean length of the trajectories in the demonstrations).

TABLE I: Absolute mean errors of positions and angles networks

TEST	initial	intermediate	final	mean
meters	0.0594	0.0572	0.0533	0.0566
radians	1.3234	0.5486	0.4142	0.7621

Initially we conjectured that the networks would be able to distinguish between the two types of dirt. Figure 6 shows the outputs of the two AlexNet’s ‘FullConn 2’ layers projected into a three dimensional space using Principal Component Analysis (PCA). The two plots show the output of the layer after running the networks on the full dataset (1000 examples). Looking at the plots, wiping (blue dots) and sweeping (purple dots) actions are easily separable. Therefore, they produced different reference frames for wiping and sweeping (see Fig. 5 for some examples).

B. Robot experiments

The whole system was then tested on the iCub robot. Various examples of scribbles and lentils clusters were placed

on the table in front of the robot. For each scribble/cluster presented, the robot performed autonomously three consecutive cleaning sequence. Each sequence can be decomposed in various sub-processes: look at the table, predict reference frames and trajectory, generate the motor commands through the iKin interface and execute the wiping/sweeping movement. The goal of the robot was to clean the table at the end of the three sequences. Torso and head of the robot were fixed during the entire experiment. For a more detailed qualitative analysis, the reader can watch the video attached to this article following the experiments done in the robot since the learning from demonstration part until the autonomous execution of the cleaning task⁵. While the results presented in this subsection are still qualitative, we expect to present a thorough quantitative analysis in subsequent work.

Figure 7 depicts two cleaning sequences performed by the iCub during the test (the upper row depicts examples of sweeping, while the lower row of wiping tasks). The images show the table state before each cleaning action and the respective predicted trajectory to be executed. In all the cases the networks and the GMR model predict reasonable trajectories. It is possible to notice that between the second and third iterations the table is poorly cleaned by the robot even if the calculated trajectories pass on top of the dirty area. The reason is a non-optimal orientation of the end effector during the movement, as a higher priority on position than orientation was set in the iCub Cartesian controller in order to better follow the generated trajectories. This resulted in a precise positioning of the hand at the expense of a precise orientation. In future implementations a deeper analysis of the controller parameters will be performed.

V. CONCLUSIONS AND FUTURE WORK

In this article, we have presented a fully autonomous system able to clean a table using learning from demonstration and deep neural networks methodologies. We have shown the

⁵The video can be download in the following link <https://youtu.be/DwGTHU8GyKE>

agent can learn the action implicitly from the network using the image features extracted on the convolutional layers: this is one of the main contributions of our work. No kind of marker or visual clue is used to help the robot, as it relies only on raw sensor data. The trajectories generated by TP-GMM using the reference frames provided by the network generalize well on the test set. After a set of demonstrations the robot is able to perform the cleaning task autonomously, generalizing to unseen situations. For future work, we are planning to enlarge the dataset in order to be richer in examples. Indeed, one of the limitations is the fixed positions of the robot (torso+head) and the fixed position of the table. We intend to project the images on a common image plane using homography based on the height of the table and the robot's joints position. Hopefully this will result in a system able to generalize on table height and robot initial configuration. The dataset will be enlarged adding Perlin noise [24] to the background in order to make the system robust to table texture and illumination. To test the generalization capability of the system we are planning to implement the system on a different robotic platform. Another interesting future improvement is the extension to bimanual cleaning. With the ability to use both arms, the iCub would increase significantly its reaching space. We speculate that would be possible to use demonstrations recorded using the right arm to train the system on the left one. Moreover, we would like to extend our approach to more than two actions.

ACKNOWLEDGMENTS

This work was partially supported by Fundação para a Ciência e a Tecnologia (project UID/EEA/50009/2013 and Grant PD/BD/135115/2017) and the RBCog-Lab research infrastructure. We acknowledge the support of NVIDIA Corporation with the donation of the GPU used for this research.

REFERENCES

- [1] F. Cavallo, R. Limosani, A. Manzi, M. Bonaccorsi, R. Esposito, M. Di Rocco, F. Pecora, G. Teti, A. Saffiotti, and P. Dario, "Development of a socially believable multi-robot solution from town to home," *Cognitive Computation*, vol. 6, no. 4, pp. 954–967, 2014.
- [2] W. D. Stiehl, C. Breazeal, K.-H. Han, J. Lieberman, L. Lalla, A. Maymin, J. Salinas, D. Fuentes, R. Toscano, C. H. Tong, *et al.*, "The huggable: a therapeutic robotic companion for relational, affective touch," in *ACM SIGGRAPH 2006 emerging technologies*. ACM, 2006, p. 15.
- [3] C. D. Kidd, W. Taggart, and S. Turkle, "A sociable robot to encourage social interaction among the elderly," in *Proceedings IEEE International Conference on Robotics and Automation. ICRA 2006*. IEEE, 2006, pp. 3972–3976.
- [4] G. Ferri, A. Manzi, P. Salvini, B. Mazzolai, C. Laschi, and P. Dario, "Dustcart, an autonomous robot for door-to-door garbage collection: From dustbot project to the experimentation in the small town of pecioli," in *IEEE International Conference on Robotics and Automation (ICRA), 2011*. IEEE, 2011, pp. 655–660.
- [5] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro, and N. Hagita, "An affective guide robot in a shopping mall," in *4th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2009*. IEEE, 2009, pp. 173–180.
- [6] G. Metta and L. Natale and F. Nori and G. Sandini and D. Vernon and L. Fadiga and C. von Hofsten and K. Rosander and M. Lopes and J. Santos-Victor and A. Bernardino and L. Montesano, "The icub humanoid robot: an open-systems platform for research in cognitive development," *Neural Networks*, vol. 23, 2010.
- [7] K. Okada, T. Ogura, A. Haneda, J. Fujimoto, F. Gravot, and M. Inaba, "Humanoid motion generation system on hrp2-jsk for daily life environment," in *IEEE International Conference on Mechatronics and Automation, 2005*, vol. 4. IEEE, 2005, pp. 1772–1777.
- [8] K. Okada, M. Kojima, Y. Sagawa, T. Ichino, K. Sato, and M. Inaba, "Vision based behavior verification system of humanoid robot for daily environment tasks," in *6th IEEE-RAS International Conference on Humanoid Robots, 2006*. IEEE, 2006, pp. 7–12.
- [9] L. Kunze, M. E. Dolha, E. Guzman, and M. Beetz, "Simulation-based temporal projection of everyday robot object manipulation," in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 107–114.
- [10] J. Hess, G. D. Tipaldi, and W. Burgard, "Null space optimization for effective coverage of 3d surfaces using redundant manipulators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012*. IEEE, 2012, pp. 1923–1928.
- [11] D. Leidner, A. Dietrich, M. Beetz, and A. Albu-Schäffer, "Knowledge-enabled parameterization of whole-body control strategies for compliant service robots," *Autonomous Robots*, vol. 40, no. 3, pp. 519–536, 2016.
- [12] D. Leidner, W. Bejjani, A. Albu-Schäffer, and M. Beetz, "Robotic agents representing, reasoning, and executing wiping tasks for daily household chores," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 1006–1014.
- [13] D. Martínez, G. Alenya, and C. Torras, "Planning robot manipulation to clean planar surfaces," *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 23–32, 2015.
- [14] F. Cruz, S. Magg, C. Weber, and S. Wermter, "Training agents with interactive reinforcement learning and contextual affordances," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 4, pp. 271–284, 2016.
- [15] P. Kormushev, D. N. Nenchev, S. Calinon, and D. G. Caldwell, "Upper-body kinesthetic teaching of a free-standing humanoid robot," in *IEEE International Conference on Robotics and Automation (ICRA), 2011*. IEEE, 2011, pp. 3970–3975.
- [16] S. Calinon, T. Alizadeh, and D. G. Caldwell, "On improving the extrapolation capability of task-parameterized movement models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013*. IEEE, 2013, pp. 610–616.
- [17] T. Alizadeh, S. Calinon, and D. G. Caldwell, "Learning from demonstrations with partially observable task parameters," in *IEEE International Conference on Robotics and Automation (ICRA), 2014*. IEEE, 2014, pp. 3309–3314.
- [18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [19] U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini, "An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 18-22, 2010, Taipei, Taiwan, 2010*, pp. 1668–1674. [Online]. Available: <https://doi.org/10.1109/IROS.2010.5650851>
- [20] U. Pattacini, "Modular cartesian controllers for humanoid robots: Design and implementation on the icub," Ph.D. dissertation, Italian Institute of Technology, 2011.
- [21] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.
- [22] T. Matsubara, S.-H. Hyon, and J. Morimoto, "Real-time stylistic prediction for whole-body human motions," *Neural Networks*, vol. 25, pp. 191–199, 2012.
- [23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.
- [24] K. Perlin, "An image synthesizer," *ACM Siggraph Computer Graphics*, vol. 19, no. 3, pp. 287–296, 1985.