Proceedings of the 21st International Conference on Global Research and Education (Inter-Academia 2024)

A Simple Learning Algorithm for Contact-Rich Robotic Grasping

M. Przybylski[a,*], J. Klimaszewski[a] and K. Wildner[b]

[a] *Faculty of Mechatronics, Institute of Automatic Control and Robotics, Warsaw University of Technology, św. A. Boboli 8, 02-525 Warsaw, Poland*
[b] *Faculty of Mechatronics, Institute of Metrology and Biomedical Engineering, Warsaw University of Technology, św. A. Boboli 8, 02-525 Warsaw, Poland*

Doi: 10.12693/APhysPolA.146.497 *e-mail: maciej.przybylski@pw.edu.pl

This paper presents the preliminary results of the work on a control algorithm for a two-finger gripper equipped with an electronic skin (e-skin). The e-skin measures the magnitude and location of the pressure applied to it. Contact localization allowed the development of a reliable control algorithm for robotic grasping. The main contribution of this work is the learning algorithm that adjusts the pose of the gripper during the pre-grasp approach step based on contact information. The algorithm was tested on different objects and showed comparable grasping reliability to the vision-based approach. The developed tactile sensor-rich gripper with a dedicated control algorithm may find applications in various fields, from industrial robotics to advanced interactive robots.

topics: robotic manipulation, contact-rich manipulation, electronic skin, learning algorithm

1. Introduction

A fast-growing interest in contact-rich robotic manipulation has been observed recently, especially in the context of *reinforcement learning* (RL) [1]. In contrast to vision-only-based approaches, contact-rich grasping allows robots to complete demanding manipulation tasks, such as assembling with micrometer precision [2].

A contact-rich robotic manipulation may assume the use of different sensors, such as force and tactile sensors, as well as robot motor load, e.g., current. External force sensors attached between a gripper and a wrist, like OnRobot HEX, provide high-precision measurements but are relatively expensive and do not provide much information about the quality of a grip, whereas tactile sensors, such as an *electronic skin* (e-skin) [3, 4], are cheap and provide rich information about touchpoints and touch forces.

Using the elastic e-skin technology [3, 4] and commercial servo motors, an inexpensive two-finger gripper has been developed (Fig. 1). The gripper with e-skin-coated fingers was mounted on a typical six-degree-of-freedom manipulator, i.e., Easy Robots ES5. The experimental setup is shown in Fig. 2.

The main focus of our work was on developing the algorithm using contact data, which was an e-skin pressure map, fingers' motor position, and fingers' motor load. The aim was to develop a method that, based on contact information, can adjust the gripper's pose to perform a successful grasp, which is when the object can be lifted.

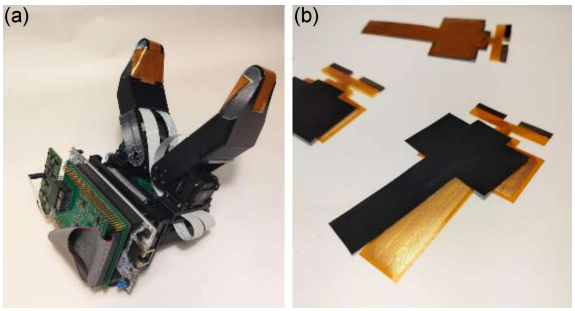

Fig. 1. The tactile-sensor-rich gripper (a) features two mechanical fingers, each covered with electronic skin (b).

The RGBD camera was used to detect an object and provide its pose in 3D space, which helped to automate the experiment and served as a baseline for contact-based grasping. However, visual data was not used in the main contact-based grasping experiment.

Typically, calibrating the e-skin's sensor positions would be the first step in developing an engineered-in rule-based control program for grasping. However, this was overcome by developing a self-calibrating data-driven algorithm that learns grasp position corrections from real-world experiments,

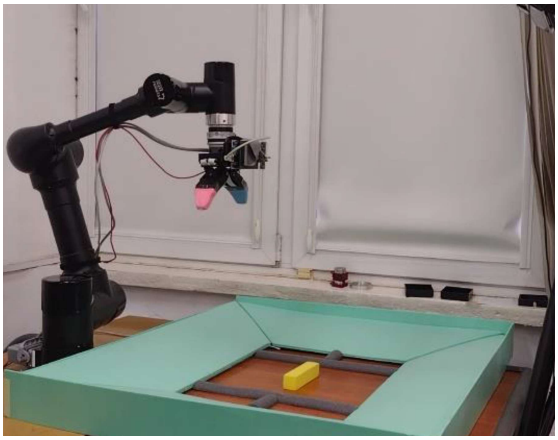

Fig. 2. The experimental setup.

Fig. 3. The gripper and the objects used for testing registered by the RGBD camera. The detected color segments are highlighted.

which is the main contribution of this work. The algorithm was tested on various objects, showing a contact-based grasp reliability comparable to the vision-based approach.

2. Experimental setup

The developed experimental setup for robotic manipulation (Fig. 2) consists of four main components: (i) the 6-degree-of-freedom Easy Robots ES5 manipulator, (ii) the e-skin-coated two-finger gripper with an electronic driver, (iii) the PC machine, and (iv) the RGBD camera. The camera observed the workspace, which was the rectangular region in the center of the "sandbox" visible in Fig. 2. A simple color segmentation allowed for the detection of colored fingertips (red and blue) and the yellow blocks (Fig. 3). The color-based object detection was sufficient to bring the effector to the vicinity of the objects.

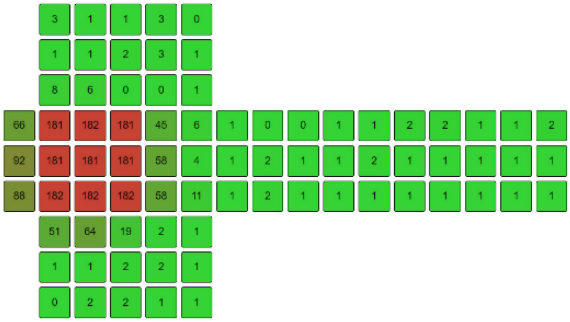

Fig. 4. The e-skin finger patch layout with a visualization of example readings caused by an object pressed on the skin (red-colored cells).

The tactile-sensor-rich two-finger gripper (Fig. 1) is the key element of the setup. Each finger is actuated by a Dynamixel 12A servo motor. The gripper works in an open/closed mode. In the closed position, the motors are commanded to achieve a fixed position, regardless of the size of the object. However, the servo motor drivers allow for compliant motion control, which helps to limit the applied force, consequently reducing the risk of damage and overheating.

The fingers are coated with e-skin patches. Each patch of the e-skin consists of 78 touch sensors arranged in a grid pattern. The shape of the e-skin patch is illustrated in Fig. 4.

An e-skin touch sensor is a *force-sensitive resistor* (FSR) that is approximately 5×5 mm^2. Detailed information on the principles of e-skin operation can be found in [3, 4]. The e-skin's electronic driver, located on the gripper's base, publishes measurements over Ethernet as 2D arrays of natural numbers, one value for each touch sensor and one 2D array for each of the two fingers. The measured values are in internal units proportional to the magnitude of an applied force (Fig. 4).

3. Grasp learning algorithm

The presented grasp learning algorithm assumes that the gripper is close to an object and that only a small pose correction is required. Specifically, the adjustment range was set to ± 50, ± 50, ± 12.5 mm in the X, Y, Z directions, respectively, and $\pm 90°$ around the Z axis, which was perpendicular to the table surface and was aligned with the wrist rotation axis. The correction is applied with respect to the initial grasp position, which is the center of mass of a 3D point cloud representing the detected object.

As the main goal is to evaluate whether contact-based grasp adjustment is comparable with the vision-based approach, the grasp learning algorithm

is presented in two variants: a contact-based (the main variant) and a vision-based one (the baseline). Although it is natural to combine contact features and visual features together, we do not present such a variant of the algorithm to not obscure the role of contact data.

The measured contact-related values were an e-skin pressure map (a 2D array of size 9×16), a motor angular position, and a motor load for each of the two fingers. All measured values were uncalibrated sensor-specific values, except for angular finger positions represented in radians.

More formally, a contact-related observation recorded at time t is a tuple

$$o_t^c = \left(f_t^1, f_t^2, e_t^1, e_t^2, A_t^1, A_t^2\right), \tag{1}$$

where f_t^1, f_t^2 are finger positions [rad], e_t^1, e_t^2 are finger efforts (finger servo motor loads in internal units), and A_t^1, A_t^2 are pressure maps for the fingers 1 and 2, respectively.

Whereas finger positions and efforts are scalars, pressure measurements are matrices; hence, extracting feature vectors from these matrices is necessary. 2D pressure maps can be thought of as images, and spatial image moments can be computed according to

$$m^{ji} = \sum_{x,y} A\left(x,y\right) x^j y^i, \tag{2}$$

where A is a pressure map and x, y are pressure map point coordinates. For each of the two pressure maps, a number of spatial moments were computed and stored in a vector, i.e., $\boldsymbol{m} = [m^{00}, m^{10}, m^{01}, m^{20}, m^{11}, m^{02}, m^{30}, m^{21}, m^{12}, m^{03}]$. Finally, for each tactile observation o_t^c, a tactile feature vector can be computed, i.e., $\boldsymbol{f}_t^c = [f_t^1, f_t^2, e_t^1, e_t^2, \boldsymbol{m}_t^1, \boldsymbol{m}_t^2]$. Feature vectors of observations recorded during a learning phase are stored in F^c.

Tactile feature vector components are of different scales; therefore, feature vectors $\boldsymbol{f}_t^c$ are divided by a standard deviation σ_{F^c} that is recomputed on every new observation, which gives a normalized feature vector $\boldsymbol{f}_t^{nc}$ stored in F^{nc}.

As normalized tactile feature vectors are points in a feature space, the nearest-neighbor search method can be used to find a similar past observation and apply the same grasp pose adjustment, which is the core idea behind the presented algorithm. A grasp pose adjustment is represented as a 3D transform matrix T^{corr} but will also be called an action, denoted by "a".

The algorithm keeps track of past actions and their later reuse (using the same T^{corr}). If action a_j reuses (imitates) action a_i, then a_i is called a parent of a_j, and a_j is a child action of a_i. Such a parent–child tree-like structure is used for an expected reward computation

$$v\left(a_i\right) = \frac{\left(r\left(a_i\right) + \sum_{a_j \in \text{children}(a_i)} r\left(a_j\right)\right)}{\left|\text{children}\left(a_i\right)\right| + 1}, \tag{3}$$

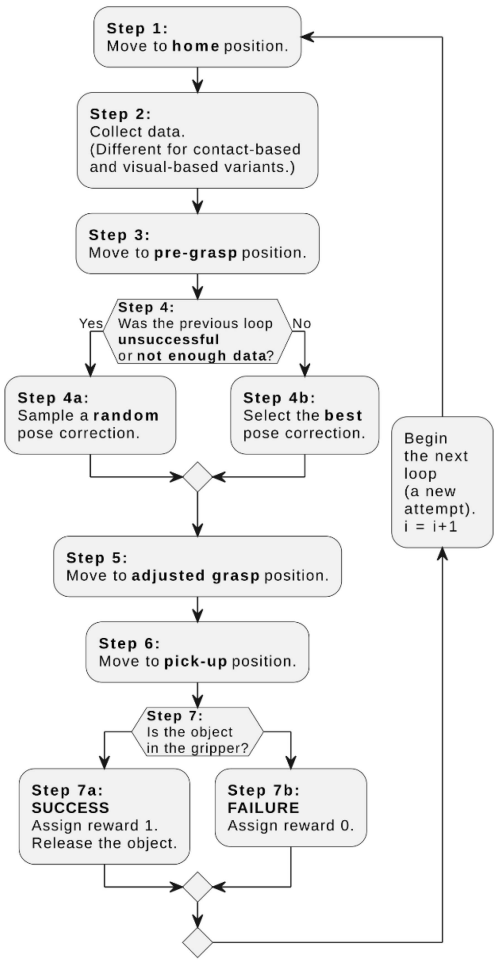

Fig. 5. The flowchart of the algorithm.

where $r(a_i)$ and $r(a_j)$ are immediate rewards (one on success and zero on failure) of i-th and j-th actions, respectively, and children(a_i) returns are a set of child actions of i-th action. In other words, the expected reward of i-th action is the average of its immediate reward and the immediate rewards of its child actions.

The contact-based grasp learning algorithm (Fig. 5) performs the following steps in a loop (i-th loop is also called i-th attempt):

1. Move to the home position above the table. The robot is off the camera's field of view, which helps to detect the object and its center of mass (Fig. 6a).
2. Move to the uncorrected grasp position, approaching the object in a top–down direction, and close the gripper. A tactile observation is collected, and a normalized tactile feature vector $\boldsymbol{f}_i^{nc}$ is computed (Fig. 6b). Open the gripper.
3. Move to the pre-grasp position above the object (shifted by 25 mm along the Z axis from the object's center of mass).

4. If the previous attempt was unsuccessful or the number of attempts is smaller than two, then perform step 4a, else perform step 4b.
   a. Assign random pose correction to T_i^{corr}.
   b. For $\boldsymbol{f}_i^{nc}$, find $k = 4$ nearest observations in F^{nc}, select the one with the highest expected reward according to (3), and use the corresponding pose correction as T_i^{corr}.
5. Move from the pre-grasp pose by T_i^{corr} and close the gripper (Fig. 6c).
6. Move to the pick-up position above the table, and store $\boldsymbol{f}_i^{nc}$ in F^{nc} and T_i^{corr} in T^{corr} set.
7. If the object is in the gripper, then do step 7a, else step 7b.
   a. Report SUCCESS, assign reward of one to i-th attempt, i.e., $r(a_i) = 1$, and release the grasped object (Fig. 6d).
   b. Report FAILURE, assign reward of zero to i-th attempt, i.e., $r(a_i) = 0$.

In step 7 of the algorithm, the presence of an object, i.e., the successful grasp, is checked with a simple rule on the gripper aperture, i.e., the gripper aperture is greater than 5% of the full opening. Such a simple rule can be used knowing that the gripper is at a certain height above the table, and, of course, it is not sufficient if the object is still on the table.

The vision-based variant relies on the same grasping algorithm, but in step 2, it computes visual features, and the robot does not move from the home position.

The visual features are the central image moments computed for the masked depth image of the object detected in step 1, according to

$$mu^{ji} = \sum_{x,y} D\,(x,y)\,(x-\overline{x})^j\,(y-\overline{y})^i, \tag{4}$$

where D is a masked depth image of an object, x, y are image point coordinates, and $(\overline{x}, \overline{y})$ is the mass center of the object's mask. Then, the visual features vector is defined as $\boldsymbol{f}_t^v = [mu_t^{20}, mu_t^{11}, mu_t^{02}, mu_t^{30}, mu_t^{21}, mu_t^{12}, mu_t^{03}]$. Consequently, $\boldsymbol{f}_t^v$ are stored in F^v and normalized feature vectors $\boldsymbol{f}_t^{nv}$ are stored in F^{nv}.

In both variants, the proposed algorithm is an example of instance-based learning, which in step 5, by using K nearest neighbors, is similar to the locally weighted learning approach proposed in [5] and utilized by VINN [6]. However, it always uses the original grasp offset without any control averaging. Also, the algorithm samples pose corrections in a continuous action space. Thus, the algorithm can compete with state-of-the-art RL algorithms, such as TD3 (twin delayed deep deterministic policy gradient) [7], which require large datasets to train actor and critic neural models. Moreover, the algorithm does not distinguish between the learning and testing phases. Therefore, it can adjust to new objects or new unseen observations online, which is a form of continual learning [8]. Finally, recorded observations and actions can be saved and restored; thus, there is no need for learning from scratch.

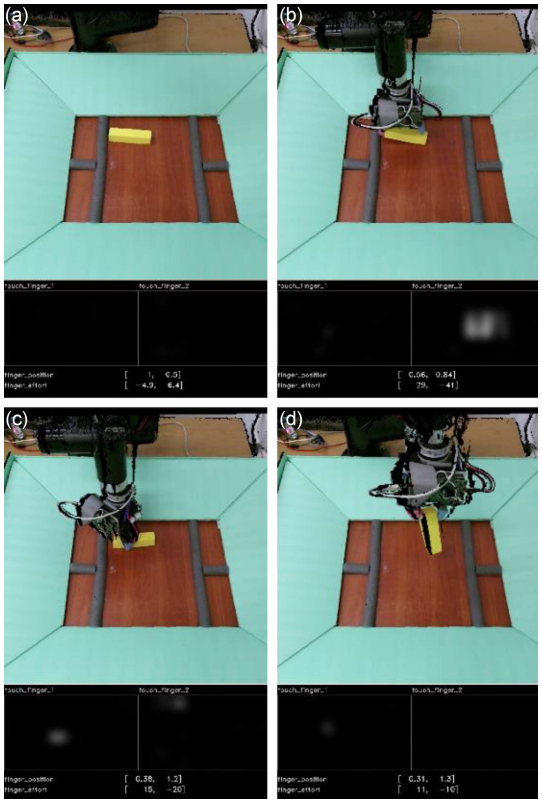

Fig. 6. The data collected in subsequent steps of the grasping algorithm. (a) The robot is in the home position (step 1). (b) The robot approaches and touches the object (step 2). (c) The robot corrects its pose and grasps the object (step 5). (d) The robot picks up the object and checks if the object is in the gripper (step 7).

4. Experiments

In the experiments, the manipulator program performed typical pick-and-place task steps, which are described in Sect. 3, i.e., the grasping algorithm steps. Although objects of different shapes were also tested, this section discusses the experiments with the most challenging one, a long yellow block shown in Fig. 6. The length of the block, which is larger than the aperture of the gripper, does not allow for a successful grasp in some configurations.

After powering up the e-skin driver, initial measurements may show some pressure even when no pressure is being applied. Therefore, the average pressure level was recorded for all e-skin sensors in a no-contact scenario. This average level was then subtracted from the raw measurements in subsequent experiments.

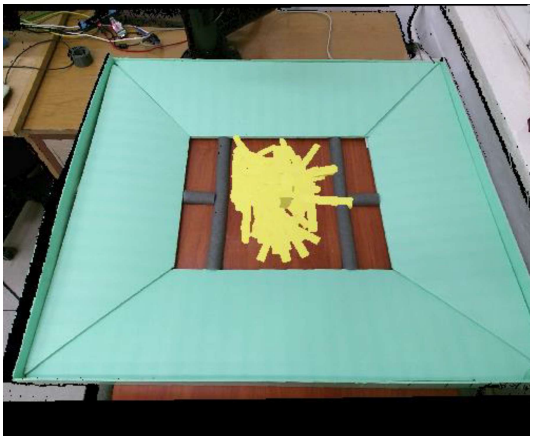

Fig. 7. An artificial image depicting all positions of the block during the contact-based experiment.

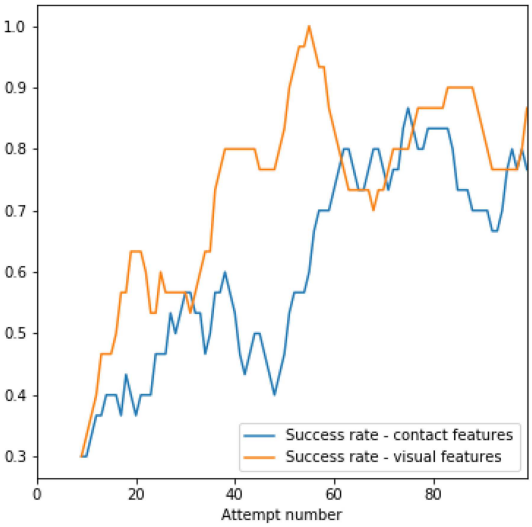

Fig. 8. The success rate of grasping for the contact-based and visual-based variants of the experiment over the course of learning, i.e., the average number of successes in a window of ten consecutive attempts. The values are the averages of three runs for each variant.

At the beginning of every experiment, a transformation from the RGBD camera frame of reference to the robot's global frame of reference was computed by recording the colored fingertips in four non-coplanar positions. Although the method is approximate, it was sufficient for the experiments.

In both variants of the experiment, three runs of one hundred grasping attempts each were performed. The experimenter could interrupt the main loop of the algorithm and place the object in a desired position. In most cases, however, the algorithm continued after both successful and unsuccessful attempts. After a successful grasp, the object was dropped with random orientation, which helped to achieve a certain degree of randomness among the block positions. All of the block positions observed in the first run of the contact-based variant are shown in Fig. 7, gathered in a single image.

The success rates for the two experiment variants, shown in Fig. 8, are the mean values from the three runs. The success rates are calculated as the average number of successes in a window of ten subsequent attempts. In both variants, the algorithm achieved a success rate of more than eighty percent. In general, the algorithm was able to improve its success rate over time and appeared to be sample efficient, as it learns within tens of attempts.

In conclusion, the results show that the novel e-skin-covered gripper provides sufficient contact information to achieve a grasping success rate comparable to the vision-based approach when used in the same learning algorithm. Moreover, with the proposed learning algorithm, grasp adjustment learning is feasible within a small number of attempts.

5. Conclusions

This paper proposes and evaluates a simple algorithm for robotic grasping learning using a prototypical gripper equipped with an electronic skin that provides contact-rich information. The experimental results show that the learning algorithm can achieve a very good success rate within tens of attempts. Moreover, the success rate obtained for the contact-based experiment is comparable to the success rate of the vision-only variant of the algorithm.

These preliminary results suggest that the learning algorithm and the prototypical e-skin gripper are worthy of further investigation and may be useful in real-world applications when visual data is insufficient, for example, objects are indistinguishable or visual data is simply unavailable.

In future work, the learning algorithm will be compared with state-of-the-art reinforcement learning algorithms in robotic manipulation tasks. Ongoing work also focuses on self-supervised multimodal robotic manipulation learning using the proposed algorithm and the gripper.

To make better use of contact-based measurements, it would be ideal to cover a larger area of the gripper surface with e-skin. At the same time, the gripper should have a more robust and compliant design, capable of withstanding high force and displacement without breaking.

References

[1] Í. Elguea-Aguinaco, A. Serrano-Muñoz, D. Chrysostomou, I. Inziarte-Hidalgo, S. Bøgh, N. Arana-Arexolaleiba, *Robot. Comput.-Integr. Manuf.* **81**, 102517 (2023).

[2] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, R. Tachibana, in: *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2017, p. 819.

[3] J. Klimaszewski, K. Wildner, A. Ostaszewska-Liżewska, M. Władziński, J. Możaryn, *Sensors* **22**, 6122 (2022).

[4] J. Klimaszewski, D. Janczak, P. Piorun, *Sensors* **19**, 4697 (2019).

[5] C.G. Atkeson, A.W. Moore, S. Schaal, in: *Lazy learning*, Ed. D.W. Aha, Springer, 1997, p. 75.

[6] J. Pari, N.M. Shafiullah, S.P. Arunachalam, L. Pinto, arXiv:2112.01511, 2021.

[7] S. Fujimoto, H. Hoof, D. Meger, in: *Int. Conf. on Machine Learning*, 2018, p. 1587.

[8] C. Zhao, J. Xu, R. Peng, X. Chen, K. Mei, X. Lan, in: *2024 IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, 2024, p. 501.