# Robot Programming Interface with a Neural Scene Understanding System [⋆]

Bartłomiej Kulecki[0000−0002−2820−8212] and
Dominik Belter[0000−0003−3002−9747]

Institute of Robotics and Machine Intelligence, Poznan University of Technology,
ul. Piotrowo 3A, 60-965 Poznan, Poland
bartlomiej.kulecki@put.poznan.pl

**Abstract.** In this paper, we deal with the problem of robots programming using natural voice and gestures. We utilize voice recognition, objects detection, and human pose estimation modules to implement a new human-robot interface that allows defining the motion of the robot. The proposed interface enables robot programming without specially designed hardware interfaces. Finally, we present the efficiency of the proposed interface compared to the standard programming method.

**Keywords:** Human-robot interface · gesture recognition · 3D perception

## 1 Introduction

Cooperative robots become more popular in the modern industry where they support the production process. Once programmed, they perform the repetitive task for days or months. Re-programming the robot is a process that requires specialized domain knowledge. New robot programming interfaces enable operators to program the robot to perform complex tasks. In this research, we explore the problem of programming the cooperative robot using natural language. The main goal of the system is the interpretation of the operator's intentions from the camera images and voice commands.

Convolutional Neural Networks advanced scene understanding. To make robots more flexible and enable fast and natural motion programming, we propose the application of 3D perception and CNN for context understanding. We mimic interpersonal communication that is based on words and gestures. The operator uses his hands to select the object for grasping or asks the robot to pass the object to his hand. The voice commands trigger defined actions of the robot.

Most human-robot interaction systems take advantage of gesture recognition [6]. Neural networks are commonly used for hand/human pose detection and gesture classification [6]. The recognized gestures are later mapped with commands that correspond to the defined motion of the robot. Also, voice commands

[⋆] This work was supported by the National Centre for Research and Development (NCBR) through project LIDER/33/0176/L-8/16/NCBR/2017 and Poznan University of Technology project number 04/45/SBAD.

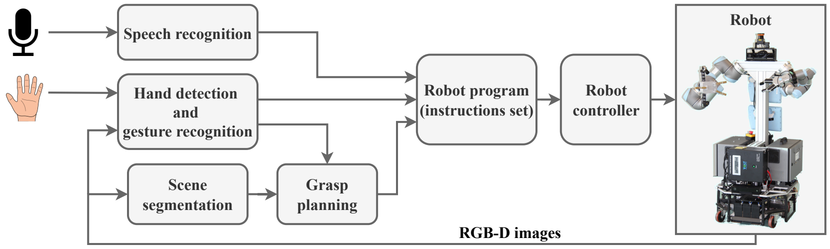

Fig. 1: Architecture of the proposed robot programming system

simplify the communication between the human operator and the robot [1] but the visual scene understanding enables the robot to interact with *a priori* unknown and continuously changing environment [1, 7]. The recognized commands can be later used to directly perform or schedule the repetitive tasks [2, 5]. In this paper, we integrate hand gestures recognition, voice commands with scene understanding to implement a new natural human-robot programming interface.

## 2 Natural Human-Robot Interaction

The architecture of the proposed system is presented in Fig. 1. The user interacts with the robot by gestures and voice commands. RGB-D images are utilized to determine the hand configuration and perform scene segmentation. Recognized voice commands, hand poses, and objects are used to plan the robot's motion.

### 2.1 Hand Detection and Gesture Recognition

The hand detection on the RGB image is performed using the state-of-the-art system for hand detection and tracking - Mediapipe Hands [8]. The output from the module is a set of 21 hand landmark points, consisting of 3 coordinates: x, y, and relative depth. We propose a method that recognizes gestures to allow effective interaction between the user and the robot. We defined two gestures represented by class names: "pointing" and "outreached". The set of 63 coordinates obtained from the hand detector is used as a feature vector for classification. First, we collected the training data set consisting of 9700 samples (5500 for the first and 4200 for the second class). Next, we trained four classification models: Random Forest, Support Vector Classifier, Ridge Classifier, and K-Nearest Neighbors. The final system utilizes the K-NN algorithm that has 99% accuracy on the test dataset containing 2400 samples. The recognized gesture type defines the robot's behavior. If the predicted class is "pointing", the program searches for the selected object. If the recognized gesture is "outreached" the target pose for object release is defined in the operator's hand.

### 2.2 Scene Segmentation and Grasp Planning

The 3D perception module performs scene segmentation based on point cloud processing [4]. The output of this module is a set of 3D-oriented bounding boxes representing detected objects. The grasp planning algorithm [4] utilizes information about the current gesture and pose of the hand to find the object selected by the user. In this process, we use the position of two landmarks detected on the image [3], named INDEX_FINGER_MCP and INDEX_FINGER_TIP. Based on the depth image and camera matrix, we obtain the 3D position of these points and create a pointing vector $P$. In the next step, the root of the finger (MCP point) is connected with the objects' centers to obtain a set of $M$ vectors. To decide which item is selected, we find the object (its $id$) with the minimum distance (smaller than the threshold value) to the pointing vector:

$$\arg\min_{id} \frac{\left\| \overrightarrow{P} \times \overrightarrow{M(id)} \right\|}{\left\| \overrightarrow{P} \right\|}. \tag{1}$$

### 2.3 Voice Commands

In the proposed system, the voice commands are used to confirm intentions expressed with a gesture. Our system [5] utilizes an of-the-shelves module - Google Speech Recognition to convert speech to text. Example commands used during experiments are:

**1)** ***grasp*** - the robot executes the trajectory to grasp the selected objects,
**2)** ***give to hand*** - the robot executes the trajectory to the position over the outreached hand and opens the gripper,
**3)** ***go home*** - moves the robot to the predefined configuration (gripper tilted down).

## 3 Results

The proposed system was tested in two scenarios (Tab. 1). During the first experiment, the robot was programmed using simple commands like *"open/close the gripper"* and *"go to initial configuration"*. If the *"go there"* command is recognized, the robot moves to the position of the detected hand. In the second experiment, the robot grasps the selected object. The example motion execution is presented in Fig. 2. During the experiments, we measured the programming and the execution time. We compared the proposed method with the programming using the visual marker, selecting objects by name (with objects detection), and traditional programming methods (teach pendant and teaching by doing) [5]. The results are presented in Tab. 1. The results show that programming the robot with gestures is more time-consuming than using the pointer but is more intuitive and does not require additional equipment.

## 4 Conclusions and Future Work

In this paper, we propose a system that utilizes voice commands, hand gestures, and 3D perception of the robot to program the execution of complex tasks. The advantage of the proposed system is the lack of additional hardware equipment like markers or pointers. Another benefit is that we can choose any type of object,

Table 1: Comparison of average time per one command in programming sequences for various programming methods.

| test case | average time per one command [s] | | | | |
|---|---|---|---|---|---|
| | teach pendant [5] | TbD [5] | pointer and voice [5] | object detection and voice [5] | gestures and voice |
| 1 | 15.4 | 9.7 | 18.2 | - | 19.7 |
| 2 | - | - | 20.7 | 22.3 | 22.7 |

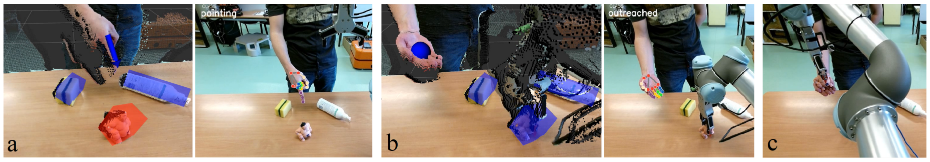

Fig. 2: Example point clouds and images from the camera during the grasping experiment: the user selects the object using the finger (a), the robot grasps the object (b) and passes it to the detected hand (c).

in contrast to the system with object detection, which allows us to select only objects from the training set. In the future, we are going to extend the number of recognized gestures to simplify robot programming.

## References

1. Adamini, R., Antonini, N., Borboni, A., Medici, S., Nuzzi, C., Pagani, R., Pezzaioli, A., Tonola, C.: User-friendly human-robot interaction based on voice commands and visual systems. In: 2021 24th International Conference on Mechatronics Technology (ICMT). pp. 1–5 (2021)
2. Dudek, W., Winiarski, T.: Scheduling of a robot's tasks with the tasker framework. IEEE Access **8**, 161449–161471 (2020)
3. Google_LLC: Mediapipe - hand landmark model. https://google.github.io/mediapipe/solutions/hands.html, [accessed 4-03-2022]
4. Kulecki, B., Młodzikowski, K., Staszak, R., Belter, D.: Practical aspects of detection and grasping objects by a mobile manipulating robot. Industrial Robot **48**(5), 688–699 (Jan 2021)
5. Kulecki, B.: Intuitive robot programming and interaction using RGB-D perception and CNN-based objects detection. In: Szewczyk, R., Zieliński, C., Kaliczyńska, M. (eds.) Automation 2022: New Solutions and Technologies for Automation, Robotics and Measurement Techniques. Springer International Publishing (2022), (in print)
6. Mazhar, O., Ramdani, S., Navarro, B., Passama, R., Cherubini, A.: Towards real-time physical human-robot interaction using skeleton information and hand gestures. In: IEEE/RSJ Int. Conf. on Int. Robots and Systems (IROS). pp. 1–6 (2018)
7. Park, K.B., Choi, S.H., Lee, J.Y., Ghasemi, Y., Mohammed, M., Jeong, H.: Hands-free human–robot interaction using multimodal gestures and deep learning in wearable mixed reality. IEEE Access **9**, 55448–55464 (2021)
8. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.L., Grundmann, M.: Mediapipe hands: On-device real-time hand tracking (2020)