Autonomous Object Detection and Grasping Using Deep Learning for Design of an Intelligent Assistive Robot Manipulation System

Sanzhar Rakhimkul, Anton Kim, Askarbek Pazylbekov and Almas Shintemirov

***Abstract*— Assistive robot solutions are mostly designed as robot helpers with robotic arms and aim to assist disabled and elderly people with carrying out basic activities of daily life such as reaching household objects, i.e. cups, feeding with spoon, opening a drawer/fridge doors, etc., However, commercial assistive robotic arms with joystick control require extensive and tiring hand motor skill training that limits robot's practical usage by patients with disabilities. The main objective of this work is to present the methodology for designing an intelligent human-machine interface for a commercial joystick controlled assistive robotic arm realizing shared autonomy and supervisory control modes. Preliminary results on a RGB-D based object detection and position estimation system development using publicly available YOLOv3 and CenterNet deep learning models implementation of an autonomous object grasping mode by the Kinova Jaco robotic arm are described in detail and experimentally demonstrated.**

I. Introduction

The world population is facing a serious problem of constant increase in number of people with special needs and disabilities. According to the World Report on Disability by World Health Organization, 15% of total world's population, which constitutes approximately one billion people, have special needs, whereas 300 million people possess severe disabilities that limit people's ability to perform activities of daily life, such that they are unable to take care of themselves [1]. As reported by the United Nations Disability Statistics for the Republic of Kazakhstan, 4.6% of men and 3.4% of women experience disabilities that affect on a person's mobility, orientation and self-care abilities [2]. Furthermore, it is also important to consider the fact that the overall population is aging. For instance, the National Institutes of Health (NIH) recently estimated that in next three decades the number of people above 65 year old will be doubled and will reach 88 million people [3]. Even though such sharp aging is not the case for developing countries, in Kazakhstan it is possible to observe gradual rise of the aging index from 25.7 in 2014 to 25.9 in 2017, while elder people compose 11% of the country's total population [4], [5]. Since this segment of the population has higher risk for acquiring disability or demand for special needs, it is vital to include elder people to the target group of potential assistive robotics technology users.

This research was funded under the Nazarbayev University faculty development grant project "Development of an Intelligent Assistive Robot Manipulation System for Improving the Quality of Life of Disabled People in Kazakhstan".

All authors are with the Department of Robotics and Mechatronics, School of Engineering and Digital Sciences, Nazarbayev University, Nur-Sultan (Astana), Kazakhstan.

Corresponding author: A. Shintemirov, ashintemirov@nu.edu.kz.

Currently, the market of assistive robotics arms can present different solutions for people with special needs, allowing to decrease reliance on the assistance of external carers [6]. One of the most prominent examples is Kinova Jaco 6 or 7 degrees-of-freedom (DOF) assistive robotic arm equipped with a 3-fingered adaptive gripper (www.kinovarobotics.com). The lightweight robotic arm structure (5.2 kg) allows it to be used as part of wheelchair mounted robotic arm (WMRA) assistive systems. The robot is controlled using a 3-axis joystick with a 2-button head for switching between the robot and gripper control modes [7]. However, execution of joystick control sequences to achieve desired arm reaches and intuitive object grasps requires extensive and tiring hand motor skills training that limits effective practical usage of the robot by unprepared users, especially in the presence of hand and upper limb disabilities [8]. Aiming to improve usability of this and other existing commercial assistive solutions by adding additional levels of robot autonomy a lot of research efforts are applied to develop intelligent human-machine interfaces (HMI) realizing shared or supervisory control modes of assistive robots [9].

In the shared control, i.e. semi-autonomous, mode a user directly operates a robotic arm through, for instance, a joystick or a body interface whereas the system recognizes user intentions and adjusts the robot motion for more accurate task execution [10]–[12]. Alternatively, in a supervisory mode, a robotic arm automatically executes various predefined tasks with aid of an external visual sensory system for target object detection and position/pose estimation upon receiving high-level commands from a user via, for example, vision-, voice-based or brain-machine interfaces [13]–[17]. The choice of the control modes largely depend on the user's upper limb and hand residual functional capabilities.

This study presents preliminary results on development of an intelligent HMI for the Jaco assistive robotic arm realizing shared and supervisory control modes aiming to provide disabled users with the ability to comfortably control the robots with minimal efforts. The paper is structured as follows: an overall project methodology with related work is presented followed by a review of artificial neural network models applied for object classification using RGB/RGB-D data. The choice of two most suitable RGB-image based object classification deep learning models with a depth data-based object position estimation algorithm are then described in detail and evaluated experimentally in an autonomous object grasping task executed by the Jaco robotic arm.

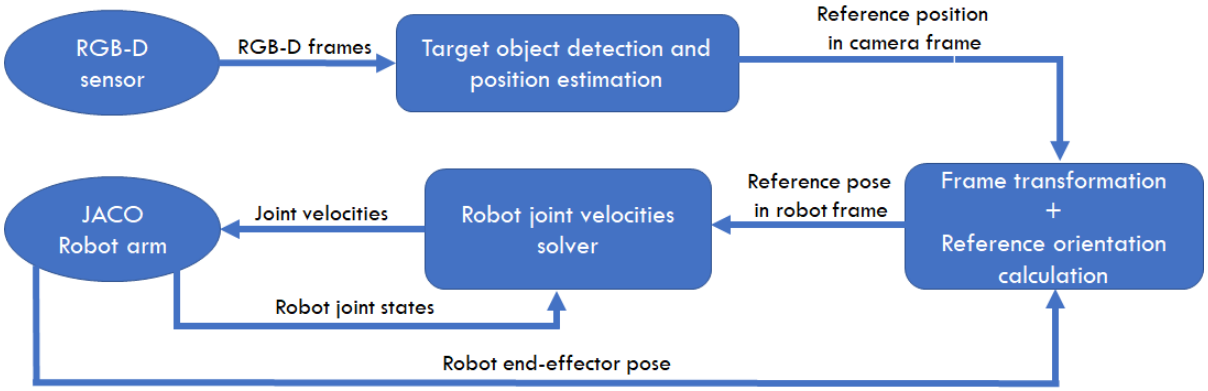


Fig. 1. Control flow chart of the intelligent assistive robot manipulation system in the supervisory control mode.

II. System Overview

A. Project Methodology

The main objective of the project is to enhance the autonomy of the Kinova Jaco 6-DOF assistive robotic arm with a three fingered gripper aiming to ultimately develop an intelligent HMI for comfortable shared autonomy and/or supervisory telecontrol of the manipulator. This will be done through incorporating a combination of machine learning based methods for autonomous target object detection and position/pose estimation based external RGB-D sensor data processing, real-time robot motion planning and robot learning for human intent prediction.

Specifically, the following tasks are pursued in the project:

- Integration of a RGB-D camera sensor and a Jaco robotic arm into the Robot Operating System (ROS) programming environment (`www.ros.org`) on a control PC for real-time robot joint velocity control;
- Development and implementation of a deep learning-based target object detection, classification and depth data processing-based position/pose estimation algorithms with experimental testing on the manipulator;
- Development of a graphical user interface for target object selection on a control PC;
- Developing and implementation of a machine learning-based human intent prediction system for processing user's joystick control actions.

In a shared control mode, the robot motion defined by the user through joystick control can be automatically corrected by the HMI in anticipation of the user's intentions. Acquisition the user's joystick control actions jointly with the robot state feedback data will be the basis for developing human intent prediction algorithms. As human users tend to apply different motions depending on many factors, i.e. intentions, surrounding environment, etc., for executing the same tasks, classical estimation techniques based on kinematic or dynamic models designed for very short-term human motion estimation, may not be suitable for accurate user intent prediction over a longer prediction horizon. On the contrary, methods based on probabilistic machine learning techniques, e.g. hidden Markov models, can accommodate human uncertainty [12], [18], [19]. Final inference of the user's intention, e.g. an identification of a target object and its position/pose estimation for robot grasping, will be made with fusion of RGB and depth data from a ROS integrated RGB-D camera sensor. As a result, a robotic arm motion will be corrected in real-time in the shared control mode, for instance, the robot end-effector will be automatically preshaped to a desired grasp pose for target object grasping, thus correcting the user rough control joystick commands.

Similarly, in the supervisory control mode the robotic arm motion trajectory will be automatically planned and executed for a variety of tasks, e.g. autonomous grasping of a target object detected with a RGB-D system sensor, upon receiving the user's high-level task command. With respect to a selected task the robot will perform a sequence of predefined actions, for example, pour water from a grasped bottle to a cup, bring a grasped cup to the user mouth, etc. This mode is summarized in the control flow chart in Fig. 1 and consisting of several modules discussed in detail in the following sections.

B. Object Detection and Position/Pose Estimation

It is generally known that development of an effective vision-based object detection and position/pose estimation software module requires designing and training a custom or adoption of an existing pre-trained deep learning model. Current state-of-the-art deep learning research offers a wide variety of pre-trained models based on RGB, RGB-D or depth data sets, upon which corresponding HMI modules can be developed. For instance, Maturana et al. developed a convolutional neural network (CNN), called VoxNet, for object recognition and 6-DOF pose estimation using 3D point cloud data generated from RGB-D cameras, LiDARs and 3D CAD models [20]. The proposed VoxNet demonstrated better accuracy rate than other models, such as UFL+SVM and GFH+SVM and outperformed them using different data sets, i.e. ModelNet, NYUv2, and Sydney Urban Objects. Xiang et al. [21] proposed a generic framework for object 6D pose estimation, named the Pose CNN, which was

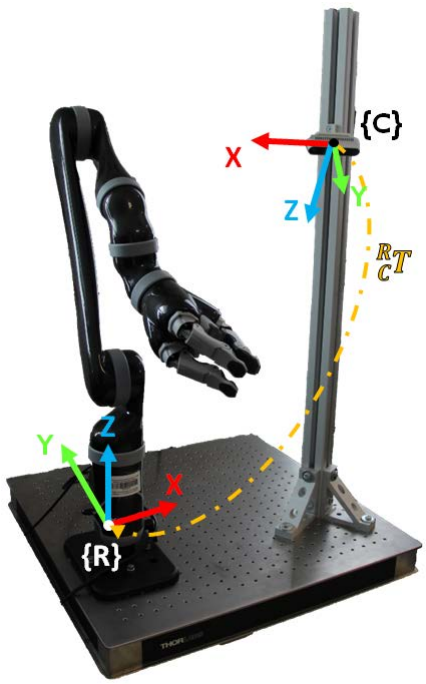


Fig. 2. Laboratory experimental setup with identified reference frames of the Jaco robotic arm {R} and the Intel RealSense RGB-D camera sensor {C} and relative transformation $^{R}_{C}T$ between the frames.

designed to overcome the limitations of other methods, e.g. an occlusion problem, that significantly reduced recognition performances or, in case of the methods based on establishing 2D-3D correspondences for 6D pose estimation, inability to handle symmetric objects. Tremblay et al. [22] presented the Deep Object Pose Estimation (DOPE) neural network that was trained on synthetic data, i.e. object 3D CAD models with object distance and lighting conditions similar to real conditions. This significantly decreased the level of training complexity. It was shown that the DOPE neural network trained using mixed synthetic and real object image data is more precise and robust in data generalization and accuracy thresholds compared to the Pose CNN. An alternative approach proposed by Redmon et al. [23] for object recognition and localization uses a highly deep neural network and casual RGB cameras, called as YOLOv3 object recognition system. It was shown that using a scaling mechanism and sliding windows, it is possible to achieve high accuracy and high frame rate without using high-cost special RGB-D cameras or laser scanners. Alternatively, Duan et al. [24] introduced a model for real-time object detection and recognition, named CenterNet. This model is based on exploring and estimation of the region that is near to geometric center of an object, with one extra keypoint, used for object bounding box estimation.

C. Experimental Setup

Figure 2 presents the experimental laboratory setup of the developed intelligent assistive robot manipulation system consisting of the Kinova Jaco robotic arm, an Intel RealSense D435 RGB-D sensor (`www.intelrealsense.com/depth-camera-d435/`) fixed on a vertical rod using a 3D printed cantilever and directed towards the front working zone of the manipulator.

The Jaco robotic arm is interfaced to a control PC through an USB interface (not shown in the figure) and is controlled in the joint velocity control mode from ROS. The robot inverse kinematics solver is realized using a model predictive control based solver modified from the approach [25]. The details of the solver implementation will be presented in a future publication of the authors.

Once a target object is detected and its coordinates are estimated in camera frame C using the presented RGB-D data processing approach, they are transformed into robot frame R, as denoted in Fig. 2. The robot motion trajectory is then planned to reach the target object and executed by the robot. The transformation matrix $^{R}_{O}T$ describing relation between the frames can be computed using the presented below simple four point calibration procedure based on relative transformations. At first, four arbitrary points are chosen in the robot working space and their coordinates $[R_1, R_2, R_3, R_4]$ and $[C_1, C_2, C_3, C_4]$ are recorded in both the robot and the camera frames correspondingly. Three out of the four points should be in one plane and the fourth one is outside the plane. Then, assuming one of the points to be the origin of a global frame and denoting it as O, the transformation matrices from this point to both the robot frame, i.e. $^{R}_{O}T$, and to the camera frame, i.e. $^{C}_{O}T$, are calculated as follows.

1. Assume one point as an origin of the robot frame

$$R_O = R_1. \tag{1}$$

2. Define the x-axis with respect to R_O as follows

$$R_X = \frac{R_2 - R_O}{\|R_2 - R_O\|}. \tag{2}$$

3. Calculate vector $R'_Y = R_3 - R_O$, and remove its component parallel to R_X, so that R''_Y is orthogonal to R_X as below

$$R''_Y = R'_Y - \frac{R_X R'_Y}{\|R_X\|^2} R_X. \tag{3}$$

4. Define y-axis as norm of R''_Y as follows

$$R_Y = \frac{R''_Y}{\left\|R''_Y\right\|} \tag{4}$$

5. Following steps are performed to define z-axis

$$R'_Z = R_4 - R_O. \tag{5}$$

$$R''_Z = R'_Z - \frac{R_X R'_Z}{\|R_X\|^2} R_X; \tag{6}$$

$$R^{(3)}_Z = R''_Z - \frac{R_Y R''_Z}{\|R_Y\|^2} R_Y; \tag{7}$$

$$R_Z = \frac{R^{(3)}_Z}{\left\|R^{(3)}_Z\right\|}. \tag{8}$$

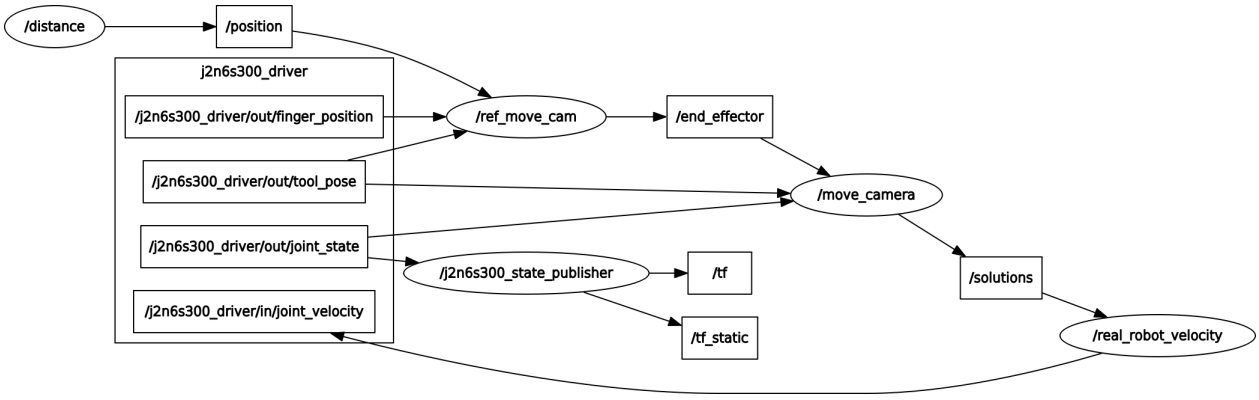


Fig. 3. Schematic of the laboratory setup control system implemented in ROS.

6. The transformation matrix ${}^{R}_{O}T$ from frame O to robot frame R is constructed as below

$$ {}^{R}_{O}T = \begin{bmatrix} R_X & R_Y & R_Z & R_O \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{9} $$

7. Similar procedure is repeated to compute the transformation matrix ${}^{C}_{O}T$ from frame O to camera frame C.

8. Finally, following the post-multiplication rule for relative transformations, we calculate the transformation matrix from the camera to the robot frame as follows

$$ {}^{R}_{C}T = {}^{R}_{O}T \; {}^{O}_{C}T = {}^{R}_{O}T \; {}^{C}_{O}T^{-}1. \tag{10} $$

All system hardware components, data processing and control algorithms are integrated into the ROS, the worlds most popular open-source middle-ware platform for intelligent robotics research. Figure 3 illustrates the details of the system's supervisory control modes implementation in ROS replicating the proposed control flow chart presented in Fig. 1. The */distance* node calculates the position of target object in Cartesian coordinates in the camera's reference frame and sends it to */position* topic. The node */ref_move_cam* receives target reference position and performs relative transformation, as it mathematically described in Section II-C. Then, reference orientation is calculated and the desired pose of end-effector is published to the */end_effector* topic. The node */move_camera* is a joint velocity solver which generates array of target joint positions and joint velocities and publishes them into */solutions* topic. Finally, node */real_robot_velocity* splits this array into two: joint velocities and joint position, and sends joint velocities array to the robot.

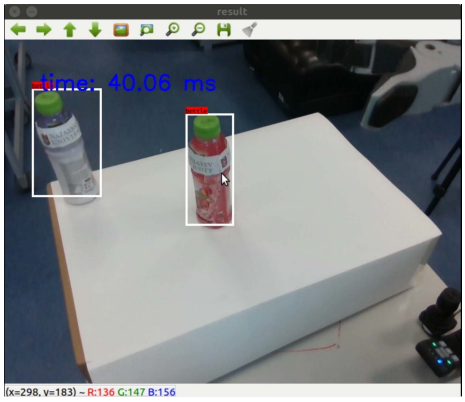
III. Implementation and Results

A. RGB-D Sensor Data Processing Models

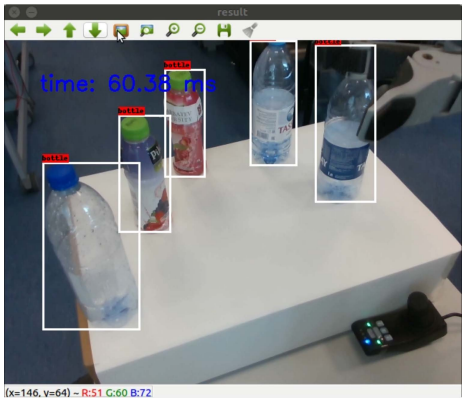
Initial tests with the Intel RealSense D435 RGB-D sensor, utilized in the laboratory setup, revealed that it provides noise depth data. This made unrealistic our initial plans to rely on stable depth data frames for simultaneous RGB and depth data processing. To cope with this problem we decided to test two approaches based on YOLOv3 [23] and CenterNet [24] object recognition models. Thus, a target object is detected in a RGB frame first, then the object position is estimated by taking the average distance value to the target provided from 30 depth frames of the RGB-D sensor.

YOLOv3 is considered as a state-of-art model in the area of object detection with fast inference, even though it can maintain only a base level of accuracy. Despite the fact that there are other models available with higher mAp (mean average precision) for RGB images, the YOLOv3 model has the essential characteristic that is its high-speed frame processing, which makes this model very suitable for real-time robot control applications. The adopted in this project YOLOv3 model requires an input image size of 416x416 that results to 65.86 bn (billion operations per second) floating-point calculations. The model executes within 50 fps on average on a powerful PC workstation with two NVIDIA GTX 1080Ti GPU graphics cards. Equivalently, for WMRA design, the usage of a tiny YOLOv3 model with FLOPS of 5.56 bn should increase fps from 50 to approximately 300 [23]. The application of the YOLOv3 model is beneficial in autonomous embedded system designs based on system-on-module computing platforms with integrated CPU, GPU and flash storage in one module such as, for example, NVIDIA Jetson TX2 board (`www.nvidia.com/en-us/autonomous-machines/embedded-systems/`). Our YOLOv3 model has weights pre-trained on the COCO image dataset with 80 object classes. The average mean precision of the model is fairly accurate, nearly 31.0 mAP. This is twice faster than any other real-time object detection model such as SSD512 [26].

CenterNet has several different pretrained weights with various backbones as image feature extraction tool. They are Hourglass-104, DLA-34, and ResNet-101 deep feature extraction layers that mainly differ by a single frame processing time and a mean average precision parameters. For the project aiming at designing a real-time object detection system we have selected the Deep Layer Aggregation (DLA) 34 highly accurate and fast model that has approximately 25 fps and 37.4% accuracy in mAp. DLA-34 is twice as fast as YOLOv3 with Darknet53 feature extraction and 6.4% AP more accurate [24]. There are two additional versions of CenterNet that use flip test augmentation and multi-scale object detection with different window size. In this project,



(a)



(b)

Fig. 4. Target object recognition using YOLOv3 (a) and CenterNet (b) deep learning models with 40.06 and 60.38 ms processing times, respectively.

the CenterNet with flip augmentation is used providing 17 fps. However for the real-time system design, the CenterNet without flip augmentation is more convenient as final fps for object detection increases to stable 30 frames per second.

Figure 4 and Table I present the comparison of the adopted object recognition models with the results of their experimental testing for a task of a bottle object recognition. As seen from the table the both models demonstrate a high degree of accuracy with a slight difference in processing time suitable for real-time system design. The system requires 0.06 sec and 0.04 sec for the YOLOv3 and the CenterNet based data processing respectively, while the average human visual reaction time is about 0.45 sec [27]. This additional time can be accounted for object selection in a GUI by a user (as shown in Fig. 6).

Multiple system tests demonstrated that CenterNet consistently outperformed YOLOv3 in the average precision, due to its faster backend image feature extraction module DLA-34 compared with the Darkenet53 employed in YOLOv3. Hovewer, DLA-34 is much complex than Darknet53 with numerous convolutional layers stacked, even though it has less hidden layers. This demonstrates the concept of the performance-complexity tradeoff, where the less complex Darkent53 requires lesser processing time but provides lower accuracy. Thus, it is possible to choose between the models,

TABLE I
COMPARISON OF THE ADOPTED YOLOV3 AND CENTERNET MODELS

Model	Backbone	Input size	Processing time	Reported AP
YOLOv3	Darknet-53	416x416	40.06 ms	31.0
CenterNet	DLA-34	511x511	60.38 ms	37.4

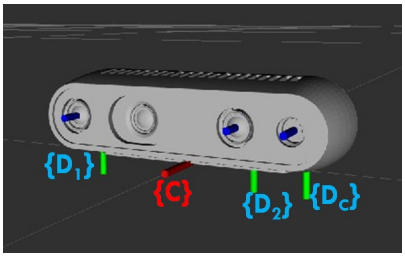


Fig. 5. Reference frames of an Intel RealSense D435 camera.

where the less accurate one is faster, whereas the more accurate model is slower.

B. Target Object Position and Distance Estimation

The Intel RealSense D435 sensor has several internal reference frames as shown in Fig. 5. The camera's base frame C is located in the physical center of the device, whereas two reference frames are aligned with the camera two infra-red sensors $\{D_1\}, \{D_2\}$ and one frame corresponds to a RGB sensor of the device $\{D_C\}$. The base frame is used C is used to perform the relative transformation from the RGB-D camera frame to the robot frame as discussed in Section II-C.

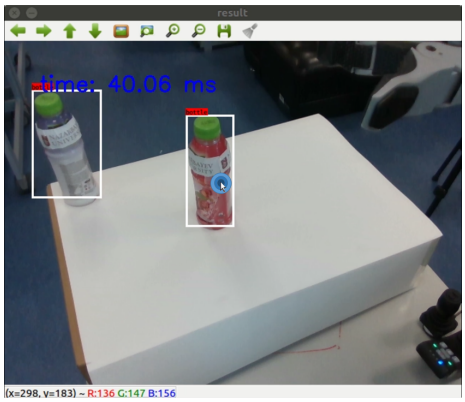
In order to find the object position on a scene, frames $\{D_1\}$ and $\{D_2\}$ are mapped to each other to create a single depth frame $\{D_d\}$.

After the implemented YOLOv3 or CenterNet model localizes a boundary box of an object and its central position in a RGB frame $\{D_C\}$, then the coordinates of the corresponding center pixel in the aligned depth frame $\{D_d\}$ are derived. Finally, the depth coordinates of the detected object are transformed into corresponding $X - Y - Z$ coordinates in the camera frame C using the Point Cloud Library (PCL) (`http://pointclouds.org`).

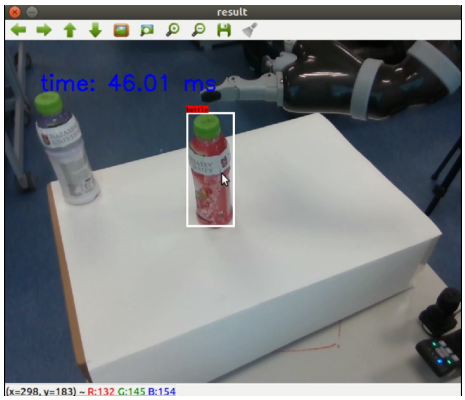
One of the drawbacks of the Intel RealSense RGB-D camera is its constant escalation of depth frame, so it often gives a wrong value of depth. Our approach to solve this problem is to use a centroid inside the boundary box. Firstly, another boundary box is created four times smaller than the original one with the same center. Then, the position for each index inside the new boundary box is estimated and transformed into Cartesian coordinates. Finally, a median for each axis is calculated separately. This method was proved to be successful with high precision.

C. Autonomous Object Grasping Test

Initial experimental testing of the developed intelligent assistive robot manipulation system were done in the supervisory control mode for the task of an autonomous object grasping by the robotic arm.



(a)



(b)

Fig. 6. A user GUI for the system supervisory control mode: a) two bottles are detected and a user selects the red bottle; b) the red bottle is chosen; the robot starts motion towards the selected target object.

The YOLOv3 model was implemented in the experiment. Observing the object detection performance it was found out that in the case when the system detected more than one previously specified target objects (e.g. bottles) on a scene, this caused unstable fluctuating movement of the robotic arm trying to approach all target objects at the same time. For this reason, a simple graphical user interface (GUI) was developed to allow a user to choose a target object for autonomous robot grasping.

The developed GUI has a very simple design of a window with a view from the RGB-D sensor showing recognized objects on a scene. Figure 6 demonstrates the user interface window with two bottles wrapped by bounding boxes with labels as a result of the RGB data based object detection. The robot's movement towards a target object is initiated by clicking on a desired object in this GUI window as shown in Fig. 6(a). Consequently, only the selected bounding box is left and the robot starts movement towards the selected object, i.e. a red bottle, as presented in Fig. 6(b).

Figure 7 presents a sequence of camera shots demonstrating the autonomous robot object grasping task execution with the developed object recognition and position estimation framework. Initially the robot in its pre-defined home configuration with an open gripper as shown in Fig. 7(a). Then, once the target object is selected by the user in the GUI and its position is determined, the control system plans and executes the robot arm motion towards the target object (Figs. 7(b) and 7(c)). Subsequently, the object is grasped with the robot gripper from the side position as shown in Fig. 7(d). In overall, multiple test runs with different objects confirmed the validity of the proposed approach for object detected and position estimation. In all cases the system were able to accurately position the robot over the detected objects in automatic mode. The video demonstration of this work is available at the author's research lab website `www.alaris.kz`.

The main problem encountered during implementation process is the occlusion of a target object by the robotic arm. When the robot approach on short distance to a target object and, consequently, part of the target object is closed by the manipulator, an object recognition model can no longer provide stable output. Due to this, the robot control algorithm is not able to generate correct reference pose and, therefore, the manipulator can not reach the desired destination. The temporary solution to this problem adopted in this work is to stop tracking a target object after the distance between the robot gripper and the object drops beyond a certain threshold. In this way the system is able to reach a target object. On the contrary, the drawback of this approach is that the system becomes less adaptive to last minute changes of on a scene, e.g. removal of a target object by another persons, etc. As future work, we plan to tackle this issue and modify the strategy by employing an algorithmic approach.

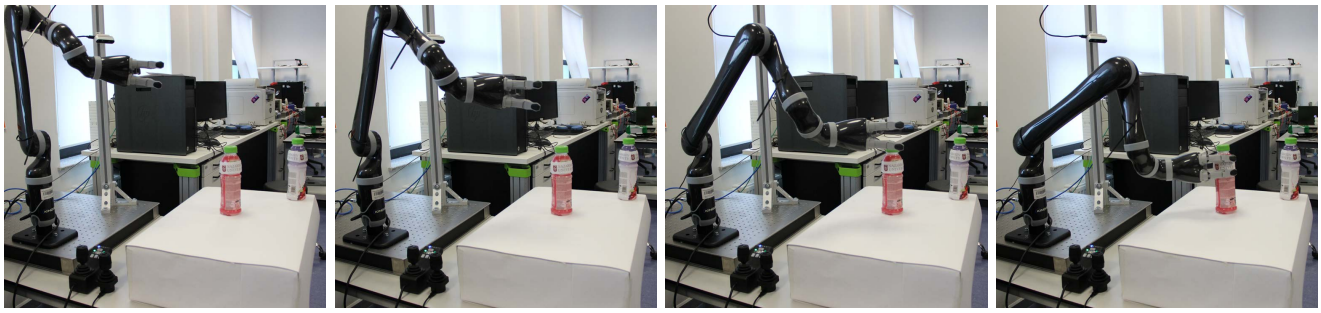
IV. Conclusion

This work presents the project methodology and preliminary results of development of an intelligent assistive manipulation system focusing on realization of the system supervisory control mode. Specifically, the implementation of the RGB-D data based object detection and position estimation is presented in detail for realizing automatic robotic grasping of objects. The proposed system was designed for and experimentally tested on Kinova Jaco assistive robotics arm.

The main problem encountered during this project is occlusion of target object by the manipulator. For the time being, it was decided to apply simple threshold to handle this problem. In the future, it is planned to apply an algorithmic approach to resolve this problem. Also, future work will focus on enhancing the system performance in terms of more accurate robot motion planning and execution, and extension of detected object classes. Furthermore, it is planned to develop and implement a shared autonomy control mode taking into account human intention prediction.

References

[1] WHO, "World Report on Disability.," 2011.
[2] U. N. Statistics Division, "UN Disability Statistics by Countries," *United Nations*, 2018.
[3] W. He, D. Goodkind, and P. Kowal, "An Aging World: International Population Reports 2015," 2016.



(a) (b) (c) (d)

Fig. 7. Video frames at four different time instants during an autonomous object grasping experiment in the system supervisory control mode.

[4] D. of Statistics of the Republic of Kazakhstan, "The Republic of Kazakhstan Age Indexes in Regions." http://stat.gov.kz/, 2017.

[5] D. of Statistics of the Republic of Kazakhstan, "The Republic of Kazakhstan Country Population by Regions." http://stat.gov.kz/, 2018.

[6] M. Beaudoin, J. Lettre, F. Routhier, P. Archambault, M. Lemay, and I. Glinas, "Long-term use of the JACO robotic arm: a case series," *Disability and Rehabilition Assitive Technologies*, no. 31, pp. 1–9, 2018.

[7] "Kinova Jaco-2 Product Specifications." https://www.kinovarobotics.com/sites/default/files/KINO-2018-Bro-Assistive-ZH_YUL-06-R-Web.pdf, 2019.

[8] D.-S. Vu, U. C. Allard, C. Gosselin, F. Routhier, B. Gosselin, and A. Campeau-Lecours, "Intuitive Adaptive Orientation Control of Assistive Robots for People Living With Upper Limb Disabilities," in *2017 IEEE International Conference on Rehabilitation Robotics (ICORR)*, pp. 795–800, 2017.

[9] P. Beckerle, G. Salvietti, R. Unal, and et. al., "A Human-Robot Interaction Perspective on Assistive and Rehabilitation Robotics," *Frontiers in Neurorobotics*, vol. 11, pp. 1–6, 2017.

[10] D. Gopinath, S. Jain, and B. D. Argall, "Human-in-the-loop optimization of shared autonomy in assistive robotics,"

[11] S. Jain, A. Farshchiansadegh, A. Broad, F. Abdollahi, F. Mussa-Ivaldi, and B. Argall, "Assistive Robotic Manipulation Through Shared Autonomy and a Body-Machine Interface," in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, pp. 526–531, 2015.

[12] K. Khokar, R. Alqasemi, S. Sarkar, K. Reed, and R. Dubey, "A novel telerobotic method for human-in-the-loop assisted grasping based on intention recognition," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4762–4769, 2014.

[13] C. P. Quintero, O. Ramirez, and M. Jagersand, "VIBI: Assistive Vision-Based Interface for Robot Manipulation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4458–4463, 2015.

[14] T. B. Pulikottil, M. Caimmi, M. G. D'Angelo, E. Biffi, S. Pellegrinelli, and L. M. Tosatti, "a voice control system for assistive robotic arms: Preliminary usability tests on patients," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*.

[15] H. Ka, D. Ding, and R. A. Cooper, "ARoMA-V2: Assistive Robotic Manipulation Assistance with Computer Vision and Voice Recognition," in *The 9th Conference on Rehabilitation Engineering and Assistive Technology Society of Korea*, 2015.

[16] S. Schrer, I. Killmann, B. Frank, M. Vlker, L. Fiederer, T. Ball, and W. Burgard, "An Autonomous Robotic Assistant for Drinking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6482–6487, 2015.

[17] F. Arrichiello, P. D. Lillo, D. D. Vito, G. Antonelli, and S. Chiaverini, "Assistive Robot Operated via P300-Based Brain Computer Interface," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6032–6037, 2017.

[18] R. Alqasemi, R. Dubey, and N. Pernalete, "Telemanipulation Assistance Based on Motion Intention Recognition," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1121–1126, 2005.

[19] A. K. Tanwani and S. Calinon, "a generative model for intention recognition and manipulation assistance in teleoperation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[20] D. Maturana and S. Scherer, "VOXNET: A 3D Convolutional Neural Network for Real-Time Object Recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928, IEEE, 2015.

[21] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[22] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv preprint arXiv:1809.10790*, 2018.

[23] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[24] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," *arXiv preprint arXiv:1904.08189*, 2019.

[25] M. Rubagotti, T. Taunyazov, B. Omarali, and A. Shintemirov, "Semi-autonomous robot teleoperation with obstacle avoidance via model predictive control," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2746–2753, 2019.

[26] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[27] S. Thorpe, D. Fize, and C. Marlot, "Spped of processing in the human visual system," *Nature*, vol. 381, pp. 520–522, 1996.