

Low Cost Robot Arm with Visual Guided Positioning

Petra Đurović, Ratko Grbić, Robert Cupec and Damir Filko

Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Osijek, Croatia

pdurovic@etfos.hr, rgrbic@etfos.hr, rcupec@etfos.hr, dfilko@etfos.hr

Abstract – Low cost robotic solutions are of great importance for improvement and development of robotics. In this paper, two visually guided low cost robot arms are proposed. The proposed system performs automatic hand-eye calibration and, after the calibration, positions its end effector above the object of interest using visual servoing based on off the shelf marker tracker. The presented experiments demonstrate positioning accuracy of the proposed setup.

I. INTRODUCTION

Robot arms are today widely used in industrial automation. Their accuracy, reliability and flexibility make them unavoidable components of many production processes. However, high cost of industrial robot arms hinders their wide application in households and education. The prices of the smallest industrial robot arms with sub-millimeter repeatability are incomparably higher than prices of average household machines. Therefore, low cost robotic solutions are of the great importance for the further development of robotics, since they broaden the population of robot users and also expand the base of researchers in this field.

In this paper, development of low cost robot arms is considered. The proposed solution consists of only basic components, which allow implementation of a vision-based robot manipulation system. The presented research is biologically inspired in the sense that hand-eye coordination, typical for living beings, is used to achieve positioning accuracy instead of relying on proprioceptive sensors in the robot joints, common in a vast majority of industrial robots. The goal of this investigation is to find out what accuracy can be achieved with such a simple configuration.

The paper is structured as follows. Section 2 provides a short overview of related research. Section 3 explains components of the proposed vision-based robot manipulation system and implementation of the calibration and visual servoing algorithms, while Section 4 presents the performed experiments and their results. Finally, Conclusion comments on the results and offers ideas for future work.

II. RELATED RESEARCH

There exist many researches where visual servoing is applied for control of a robotic arm. In some of them, industrial robots [1] are being used and eye-in-hand [2], or hybrid eye-to-hand/eye-in-hand [3] configuration is applied. Also, variety of hand-eye calibration methods are applied, such as convex linear matrix inequality [4]. Our setup consists of a low-cost robotic arm and an off-the-shelf camera in eye-to-hand configuration. This setup, which consists of significantly less expensive components in comparison to the hardware used in the aforementioned related research, together with an evaluation of the absolute positioning accuracy achieved by the proposed system represents the contribution of this paper.

III. VISION BASED ROBOT MANIPULATION SYSTEM

A. Components

Since the main goal of this paper is to provide an insight in the positioning accuracy which can be achieved by low cost vision guided robot manipulators, all



Figure 1. Robot Arm mounted on the stand

components are acquired from the lower price specter. The robot manipulation system shown in Figure 1 consists of a robot arm, a camera mounted on a stand, a marker used for visual servoing installed on the robot's end effector and a vision-based robot control software.

Two robots are considered: Dobot Arm V1.0 and a custom made robot arm in SCARA configuration, named VICRA (Vision Controlled Robot Arm). The price of the Dobot Arm was 900\$ and the manufacturing of the mechanical construction which connects the robot arm with a camera on a pan-tilt head costed 320 EUR. The development and manufacturing of VICRA prototype costed approximately 3000 EUR.

The Dobot package includes Dobot Arm, Dobot controller, 5 pieces of effectors, power adaptor, USB cable, toolkit, extension cable and base. Dobot Arm has 4 axes and 0.5 kg payload with maximum reach of 320 mm and +/- 0.2 mm position repeatability. Dobot Arm has three joints with stepper motors: base, rear arm and forearm, and the fourth joint with rotation servo. It weighs 3 kg and it is made of aluminum alloy. Dobot Arm is controlled by Dobot Controller which is based on Arduino Mega 2560 and FPGA board placed inside a control box. It can communicate with a PC via USB-UART. A Dobot Arm is shown in Figure 2. The Dobot arm and the RGB-D camera are mounted on a stand thereby forming a compact vision-based robot manipulation unit. The stand was designed in SketchUp [13] and built in a local workshop. The stand shown in Figure 1 was designed to provide secure camera holding in adequate position w.r.t. the robotic arm. The camera is positioned in such a way that the robot arm workspace is completely contained inside the camera's field of view. It is located sufficiently close to the robot's workspace in order to achieve precise measurement of objects which the robot should manipulate with. In the same time, the distance to the closest object must be greater than the lower bound of the camera range. Furthermore, the marker used for visual servoing must always be clearly

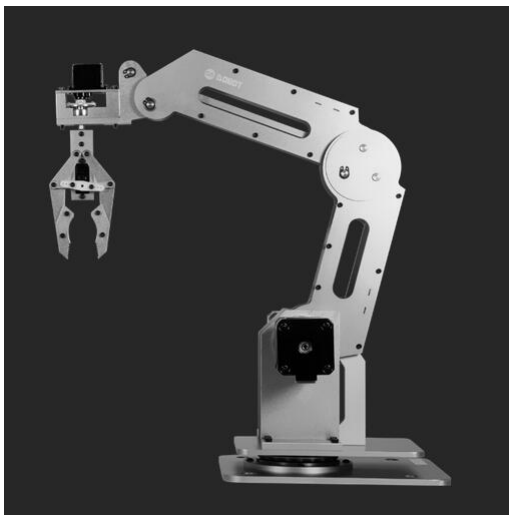


Figure 2. Dobot Arm V1.0



Figure 3. Robot manipulation system with VICRA

visible.

VICRA, shown in Figure 3, is a custom made robot arm in SCARA configuration. It has one translational and three rotational joints. The first three joints, which position the tool, are driven by stepper motors, while the fourth joint, which defines the tool orientation, is driven by a DC servo motor. The currently used tool is a gripper driven by a servo motor, but it can easily be replaced by alternative tools. The first translational joint enables vertical reach of approximately 0.6 m and the two rotational joints enable horizontal reach of approximately 0.6 m. The weight of the robot is approximately 8 kg, which makes it suitable for mounting on a mobile platform. The robot is controlled by Arduino Mega 2560 with RAMPS 1.4 shield which communicates with a PC via USB. The robot is designed for a pan-tilt head with a camera to be mounted on the top of the first axis, in order to support computer vision-based control. Also, inexpensive micro-switches are used as end-stop for all three axes.

Each of the two considered robot arms has its advantages and drawbacks. VICRA is more expensive than Dobot, but it has greater payload and wider operative range. Also, because of its configuration, in specific positions, Dobot's joints interfere with camera's view of the marker used for visual feedback. On the other hand, VICRA provides undisturbed view of the marker in each position.

Both considered robot configurations are four axes robot arms, whose absolute positioning is achieved by a 3D perception sensor being the only sensor applied. The first three axes of the robot are driven by stepper motors without encoders and the fourth axis is driven by a DC servo motor. The robots are controlled by an Arduino-based controller and the applied 3D perception sensor is

an affordable off-the-shelf RGB-D camera, Orbbec Astra S, costing 150 US\$.

Although a common RGB-camera can be used for implementation of visual feedback, a RGB-D camera Orbbec Astra S is used in the considered system because we wanted to build a system with 3D vision capability, since results of recent computer vision research clearly demonstrate advantages of 3D vision systems [6]–[10]. Orbbec Astra S is a camera optimized for short-range use cases, from 0.35 m to 2.5 m. It is compatible with existing OpenNI [14] applications and provided with Astra SDK software. Astra SDK is used for acquiring image from the camera which is then processed by algorithms presented in Sections III.B and III.C. Even though presented manipulation module currently doesn't use depth obtained from the RGB-D camera, it is planned to use this information in future versions.

Both robot manipulators are based on stepper motors and no absolute encoders or any other proprioceptive sensor are used in this research for measuring absolute position. Stepper motors are controlled by series of impulses defining relative changes in joint angles. Hence, in order to achieve absolute positioning, the initial position must be known. Furthermore, stepper motors could lose impulses, which could lead to wrong positioning. In order to achieve precise absolute positioning, visual servoing using RGB-D camera is applied.

Visual servoing is implemented by detection and localization of a marker mounted on the robot arm close to the end-effector. Marker detection and pose estimation is implemented using ArUco library for augmented reality [11] based on Open CV [12]. Since an object of interest is not always captured in the current camera field of view, the camera is mounted on a pan-tilt head. This way, the camera is able to rotate up and down, left and right and find the right angle to successfully capture objects of interest. The pan-tilt head also allows navigation of mobile robot manipulator, which can be created by mounting of the presented vision guided robot arm on a mobile platform. In the experiments reported in this paper, the capabilities of the pan-tilt head are not used, i.e. the camera orientation remains fixed during the experiment.

The control software consists of functions for controlling robot's joints [15], built in Python, and functions for visual servoing written in C++.

B. Hand-Eye Calibration

Systems where a robotic "hand" is guided by a camera, which represents "eyes", are called Hand-eye systems. There are two main types of these systems: (i) the eye-in-hand configuration, where the camera is mounted on the arm, close to its end effector, and moves along with the end-effector, and (ii) the eye-to-hand configuration, where the camera is stationary and overlooks the robot's movements. In this paper, the

second configuration is used, primarily because the camera was too big to be mounted on the applied robot arms. In order to be able to coordinate the robot movement with the information obtained from the camera images, the pose of the robot reference frame S_R w.r.t. the camera reference frame S_C must be determined. A calibration procedure used to compute this pose is known as hand-eye calibration.

To perform a hand-eye calibration, a set of images of a calibration object is acquired. The calibration object used in the research presented in this paper is a marker mounted close to the end effector of the robot arm, which is also used for visual servoing.

First, let's introduce the notation used in the rest of this paper. In this paper, ${}^B\mathbf{T}_A$ denotes the homogeneous transformation matrix representing the pose of a reference frame S_A w.r.t. a reference frame S_B . Analogously, ${}^B\mathbf{R}_A$ and ${}^B\mathbf{t}_A$ denote rotation matrix and translation vector defining the orientation and position of S_A w.r.t. S_B respectively. A rotation matrix and a translation vector are included in a homogeneous transformation matrix as follows

$${}^B\mathbf{T}_A = \begin{bmatrix} {}^B\mathbf{R}_A & | & {}^B\mathbf{t}_A \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix}.$$

Furthermore, the following notation is used for reference frames: R represents robot, C camera, G tool, and M marker reference frame.

The pose of S_R w.r.t. S_C can be computed from three marker positions, referred to in this section as position 0, 1 and 2. Assuming that the marker is clearly visible, the applied vision software provides us with 3D position of the marker denoted by ${}^C\mathbf{t}_M(k)$, for each robot position $k = 0, 1, 2$.

If the robot moves from position 0 to position 1 along the x-axis of its reference frame, then the marker positions ${}^C\mathbf{t}_M(0)$ and ${}^C\mathbf{t}_M(1)$ define the x-axis of the robot reference frame viewed from the camera reference frame, i.e. unit vector

$${}^C\mathbf{x}_R = \frac{{}^C\mathbf{t}_M(1) - {}^C\mathbf{t}_M(0)}{\|{}^C\mathbf{t}_M(1) - {}^C\mathbf{t}_M(0)\|}, \quad (1)$$

represents the x-axis of S_R w.r.t. S_C .

Furthermore, if the robot moves from position 1 to position 2, where the both positions have the same z-coordinate in the robot reference frame, then the marker positions ${}^C\mathbf{t}_M(1)$ and ${}^C\mathbf{t}_M(2)$ define a vector perpendicular to the z-axis of the robot reference frame. Consequently, unit vector

$${}^C\mathbf{z}_R = \frac{{}^C\mathbf{x}_R \times \mathbf{b}}{\|{}^C\mathbf{x}_R \times \mathbf{b}\|}, \quad (2)$$

where

$$\mathbf{b} = {}^c\mathbf{t}_M(2) - {}^c\mathbf{t}_M(1), \quad (3)$$

represents the z-axis of S_R w.r.t. S_C . Vectors ${}^c\mathbf{x}_R$ and ${}^c\mathbf{z}_R$ completely define the orientation of S_R w.r.t. S_C , which can be represented by rotation matrix

$${}^c\mathbf{R}_R = \begin{bmatrix} {}^c\mathbf{x}_R & {}^c\mathbf{y}_R & {}^c\mathbf{z}_R \end{bmatrix} \quad (4)$$

where

$${}^c\mathbf{y}_R = {}^c\mathbf{z}_R \times {}^c\mathbf{x}_R. \quad (5)$$

The position of the origin of S_R w.r.t. S_C can be computed from ${}^c\mathbf{t}_M(0)$, the previously determined rotation matrix ${}^c\mathbf{R}_R$ and the position of the marker w.r.t. the robot end effector ${}^G\mathbf{t}_M$, which is expected to be known in advance. The relation between ${}^c\mathbf{t}_M(0)$ and the corresponding position of the end effector w.r.t. robot reference frame can be described by the following equation

$$\begin{bmatrix} {}^c\mathbf{t}_M(0) \\ 1 \end{bmatrix} = {}^c\mathbf{T}_R {}^R\mathbf{T}_G \begin{bmatrix} {}^G\mathbf{t}_M \\ 1 \end{bmatrix}. \quad (6)$$

The both considered robots are constructed in such a way that the z-axis of the tool reference frame S_G is always parallel to the z-axis of S_R . Hence, if position 0 is selected so that the x-axis of S_G is parallel to the x-axis of S_R , then ${}^R\mathbf{R}_G$ is the identity matrix and (6) can be written as

$${}^c\mathbf{t}_M(0) = {}^c\mathbf{R}_R \cdot ({}^G\mathbf{t}_M + {}^R\mathbf{t}_G(0)) + {}^c\mathbf{t}_R.$$

Consequently, vector ${}^c\mathbf{t}_R$ can be computed by

$${}^c\mathbf{t}_R = {}^c\mathbf{t}_M(0) - {}^c\mathbf{R}_R \cdot ({}^G\mathbf{t}_M + {}^R\mathbf{t}_G(0)). \quad (7)$$

The eye-to-hand calibration is performed in the following steps.

Step 1. Position the robot tool at an arbitrary position ${}^R\mathbf{t}_G(0)$ on the x-axis of the robot RF.

Step 2. Capture an image and measure ${}^c\mathbf{t}_M(0)$.

Step 3. Move the robot tool in the x-direction for distance a . The resulting position is ${}^R\mathbf{t}_G(1)$.

Step 4. Capture an image and measure ${}^c\mathbf{t}_M(1)$ using the tracking tool.

Step 5. Move the robot tool in the y-direction for distance b . The resulting position is ${}^R\mathbf{t}_G(2)$.

Step 6. Capture an image and measure ${}^c\mathbf{t}_M(2)$ using the tracking tool.

Step 7. Compute ${}^c\mathbf{R}_R$ using equations (1) – (5) and ${}^c\mathbf{t}_R$ by (7).

C. Visual Servoing

The visual servoing algorithm presented in this section computes changes of joints variables required to move the marker in a wanted position based on the image acquired by the camera, a desired marker position ${}^c\mathbf{t}_{M,r}$ w.r.t. the camera reference frame and the pose of the robot reference frame w.r.t. the camera reference frame ${}^c\mathbf{T}_R$, obtained by calibration. In a practical application, the desired marker position ${}^c\mathbf{t}_{M,r}$ is computed from a pose of an object of interest obtained by an appropriate computer vision software, which is not a subject of this paper. The task of the presented visual servoing algorithm is to position the marker at ${}^c\mathbf{t}_{M,r}$ within a given tolerance. The visual servoing algorithm consists of the following steps.

Step 1. Based on a given image, the current marker position ${}^c\mathbf{t}_M$ w.r.t. the camera is computed using a tracking tool (in the current implementation, this is ArUco software).

Step 2. Compute the current marker position in the robot reference frame by ${}^R\mathbf{t}_M = {}^R\mathbf{T}_C \cdot {}^c\mathbf{t}_M$.

Step 3. Compute joint angles \mathbf{q} from ${}^R\mathbf{t}_M$ by inverse kinematics.

Step 4. Compute the reference marker position in the robot reference frame by ${}^R\mathbf{t}_{M,r} = {}^R\mathbf{T}_C \cdot {}^c\mathbf{t}_{M,r}$.

Step 5. Compute new joint angles \mathbf{q}_r from ${}^R\mathbf{t}_{M,r}$ by inverse kinematics.

Step 6. Compute the required changes in joint angles by $\Delta\mathbf{q} = \mathbf{q}_r - \mathbf{q}$.

Step 7. Send $\Delta\mathbf{q}$ to the robot controller.

These steps are repeated iteratively until the marker reaches the desired position within a given tolerance.

IV. EXPERIMENTAL EVALUATION

In order to test the developed system, two test case scenarios were performed. In the first experiment, the accuracy of the camera calibration was evaluated by comparing the estimation of the tool position measured by vision, the tool position given to the robot and the actual tool position measured manually. In the second experiment, the goal was to test the accuracy of the implemented visual servoing algorithm. The experiments were performed using Dobot Arm. Since the positioning accuracy depends on visual feedback and not on the mechanical construction of the robot, similar results are expected to be obtained by VICRA. Nevertheless, experiments with VICRA are not included in this paper, since its development is still in progress.

In order to facilitate manual measurements, in the first experiment, a laser pointer is mounted instead of a gripping tool. The manually measured positions are considered in this analysis as the ground truth positions. The tool positions given to the robot are referred to in this section as robot positions and the positions measured by

TABLE I. STATISTICAL DATA ANALYSIS OF TOOL COORDINATES MEASURED BY VISION.

	BEFORE CORRECTION	AFTER CORRECTION
Δx_{\max} [mm]	5.94	4.87
Δy_{\max} [mm]	14.61	2.30
Δd_{\max} [mm]	8.93	4.64
σ_x [mm]	1.80	1.74
σ_y [mm]	4.17	0.93
σ_d [mm]	2.61	1.76

vision are referred to as vision measurements. After camera calibration, a square matrix of 10×10 points is given to the robot as reference positions. The robot's task was to move to these points, one by one, wait for the vision system to calculate the tool pose and a human operator to measure the tool position. In order to allow comparison of the measurement obtained by vision to the open-loop robot positioning and the manual measurements, the pose estimated by the vision system is transformed in the robot reference frame. Since the robot has a laser as the end effector, the point projected by the laser on a millimeter paper allowed an easy measurement in the robot reference frame. After all 100 points were processed, a statistical analysis is performed. The results of this analysis are shown in the middle column of Table I and Table II. In Table I, the statistics of the difference between the ground truth positions and the vision measurements is shown, where Δx_{\max} and Δy_{\max} represent the maximal absolute difference in the x- and y-coordinates, Δd_{\max} represents the maximal distance, while σ_x, σ_y , and σ_d represent the standard deviations of the x-coordinates, y-coordinates and distances. The analogous statistical analysis of the difference between the ground truth positions and the tool positions given to the robot is shown in Table II.

The presented analysis shows that the calibration procedure proposed in Section III.B, although fast and simple, provides the transformation ${}^C\mathbf{T}_R$ with significant inaccuracy. We noticed that both the measurements obtained by vision and the robot open-loop positioning have significant bias, which could possibly be reduced by an additional calibration step. In order to estimate to which extent the considered positioning errors represent random noise and to which extent these errors contain a bias, which could be reduced by an additional correction step or a more thorough calibration procedure, we performed the following analysis. We computed values a_x, b_x, a_y and b_y which minimize the following error functions

$$\mathfrak{J}_x(a_x, b_x) = \sum_i (x_{r,i} - x'_i)^2,$$

TABLE II. STATISTICAL DATA ANALYSIS OF TOOL POSITIONS GIVEN TO THE ROBOT.

	BEFORE CORRECTION	AFTER CORRECTION
Δx_{\max} [mm]	7.00	1.86
Δy_{\max} [mm]	2.00	1.25
Δd_{\max} [mm]	7.23	1.75
σ_x [mm]	1.95	0.77
σ_y [mm]	0.61	0.44
σ_d [mm]	1.95	0.78

$$\mathfrak{J}_y(a_y, b_y) = \sum_i (y_{r,i} - y'_i)^2,$$

where $x_{r,i}$ and $y_{r,i}$ are reference values obtained by manual measurement, while x'_i and y'_i are values obtained by linear mapping of the measured values x_i and y_i defined by

$$x'_i = a_x x_i + b_x, \quad y'_i = a_y y_i + b_y. \quad (8)$$

The statistics of the vision error and robot error for the values obtained by mapping (8) is shown in the last column of Table I and Table II. This analysis indicates that the positioning error can be significantly reduced by an additional correction step, such as the linear mapping proposed in this section. If the robot is required to have a good positioning accuracy within an approximately planar workspace, then the presented linear mapping can be used as the final calibration step, resulting in a more accurate positioning. If the robot is required to have accurate positioning in a wider workspace including significant changes in z-direction, then a more thorough calibration procedure should be applied.

In the second experiment, visual servoing accuracy was evaluated. A marker which represented an object on the scene, was put in 30 different positions, one by one, in the robot's working region. This marker is referred to in this paper as *target marker*. After capturing an image, the center of the target marker representing the desired position was determined.

The robot initially moved to this position by setting its joints to the values computed by inverse kinematics. If the difference between the robot's position and the target marker after this movement was greater than a given threshold, which was 3 mm in this case, the visual servoing algorithm described in Section III.C was applied. This process is repeated until the positioning error falls below the given threshold. After the robot stopped moving, the distance between the laser point on the millimeter paper, which represents the position of the end effector, and the center of the target marker was measured by the human operator. The results of the

TABLE III. STATISTICAL DATA ANALYSIS OF DISTANCES BETWEEN TARGET AND OBTAINED POSITIONS BASED ON VISUAL SERVOING ALGORITHM

Δd_{\max} [mm]	6
σ_d [mm]	1.42

described experiment are given in Table III, where Δd_{\max} represents the maximum distance between the center of the target marker and the laser position and σ_d represents the standard deviation of this distance.

V. CONCLUSION AND FUTURE WORK

In this paper, a simple vision guided robot manipulation system consisting of a low cost robot arm and an off-the-shelf RGB-D camera is presented. As the main contribution, we emphasize the development of a low-cost solution which can be used by a wide population of developers. Furthermore, an analysis of the positioning accuracy of the developed system achieved by visual servoing without proprioceptive sensors is presented. From the results of this analysis it can be concluded that the positioning error is within 6 mm with standard deviation of 1.42 mm.

The main error source in the proposed positioning system comes from inaccurate determination of the marker distance w.r.t. the camera reference frame. We expect that a significant improvement to the given system can be achieved by fusing the measurement obtained by ArUco software with depth information provided by a RGB-D camera. This will be the topic of our future research.

ACKNOWLEDGMENT

This work has been fully supported by the Croatian Science Foundation under the project number IP-2014-09-3155.

REFERENCES

- [1] S. van Delden and F. Hardy, "Robotic Eye-in-hand Calibration in an Uncalibrated Environment."
- [2] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, pp. 4647–4653.
- [3] V. Lippiello, B. Siciliano, and L. Villani, "Eye-in-hand/eye-to-hand multi-camera visual servoing," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, 2005, pp. 5354–5359.
- [4] J. Heller, D. Henrion, and T. Pajdla, "Hand-eye and robot-world calibration by global polynomial optimization," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 3157–3164.
- [5] A. Aldoma, F. Tombari F, L. Di Stefano and M. Vincze, "A global hypothesis verification method for 3d object recognition," *European Conference on Computer Vision (ECCV)*, 2012.
- [6] C. Papazov and D. Burschka, "An Efficient RANSAC for 3D Object Recognition in Noisy and Occluded Scenes," *Asian Conference on Computer Vision (ACCV)*, Part I, 2010, pp. 135–148.
- [7] S. Hinterstoisser, C. Cagniard, S. Ilic, P. Sturm, N. Navab, P. Fua and V. Lepetit, "Gradient Response Maps for Real-Time Detection of Textureless Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, 2012, pp. 876–888.
- [8] C. A. Mueller, K. Pathak and A. Birk, "Object Shape Categorization in RGBD Images using Hierarchical Graph Constellation Models based on Unsupervisedly Learned Shape Parts described by a Set of Shape Specificity Levels," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Chicago, IL, USA, 2014, pp. 3053–3060.
- [9] R. Detry, C. H. Ek, M. Madry and D. Kragić, "Learning a Dictionary of Prototypical Grasp-predicting Parts from Grasping Experience," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 3157–3164.
- [10] L. Twardon and H. Ritter, Interaction skills for a coat-check robot: identifying and handling the boundary components of clothes, *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3682–3688.
- [11] ArUco: a minimal library for Augmented Reality applications based on OpenCV, <https://www.uco.es/investiga/grupos/ava/node/26>, February 2017.
- [12] A. Kaehler and G. Bradski, *Learning OpenCV 3*, O'Reilly Media, December 2016.
- [13] SketchUp, <https://www.sketchup.com/>, March 2017.
- [14] OpenNI, The standard framework for 3D sensing, <http://www.openni.ru/>, March 2017.
- [15] open-dobot , <https://github.com/maxosprojects/open-dobot>, March 2017