10.24425/acs.2019.131228

Archives of Control Sciences
Volume 29(LXV), 2019
No. 4, *pages 603–615*

Human–robot cooperation in sorting of randomly distributed objects

EDWARD JEZIERSKI, PIOTR ŁUCZAK, PAWEŁ SMYCZYŃSKI and DARIUSZ ZARYCHTA

The paper presents the possibilities of teaching a robot controller to perform operations of autonomous segregation of objects differing in features that can be identified using a vision system. Objects can be arranged freely on the robot scene also covered with others. In the learning phase, a robot operator presents the segregation method by moving subsequent objects held in a human hand, e.g. a red object to container A, a green object to container B, etc. The robot system, after recognizing the idea of segregation that is being done using the vision system, continues this work in an autonomous way, until all identified objects will be removed from robotic scene. There are no restrictions on the dimensions, shapes and placement of containers collecting segregated objects. The developed algorithms were verified on a test bench equipped with two modern robots KUKA LBR iiwa 14 R820.

Key words: human–robot cooperation, vision system, collaborative robots

1. Introduction

This article discusses the complexities of teaching a system which has the capability to repeat actions performed by humans just by observation. Presented solution concerns the scenario where objects are scattered randomly on a pedestal located inside a robotic workspace. The objects should be sorted according to their colors. Similar tasks are increasingly useful and their popularity among researchers is rising [6, 8, 12]. Fruits and vegetables sorting systems are examples of some of the most important applications [9, 11]. Due to similarity to the human learning process, learning by observation is the most desired way of teaching, however, in most cases tasks are predefined in some kind of programming mode [10].

The authors are with Institute of Automatic Control, Lodz University of Technology, Stefanowskiego 18/22, 90-924 Łódź. The corresponding author is E. Jezierski, E-mail: edward.jezierski@p.lodz.pl.

Received 09.02.2019. Revised 01.11.2019.

Researchers focus on two main aspects of a task: the first step is correct object recognition, the second one is matching robot reaction which changes, for example state or location of an object. In both cases continuous cooperation with human is possible. Various hardware setups have been developed, ranging from simple single camera solutions to more advanced systems using RGB-D images [5, 7].

Methodology of solving these problems is very rich, in many cases semantic relations between objects are taken into account. Interesting approach has been presented in [5], where object affordances (also introduced in [6, 7]) determine the range of possible interactions with an object. Manipulations such as taking the object out of a container or putting the object in different place can be recognized from demonstration performed by a human operator. Important part of object classification was assigning the "possible to grab" property which is crucial in defining the extent of robot-object interaction. It is also interesting, that some kind of segmentation is used both in recognition of the object structure and performed tasks.

2. Kube test stand

Kube test stand is designed for researching human-robot interaction supervised by vision system. Functionally, stand is divided into two subsystems: 1) executive part mounted inside cubic 2.5 m × 2.5 m × 2.5 m frame, 2) operator stand for ambidextrous control with both vision and force feedback. Test stand complies with ISO/TS15066:2016: Robots and robotic devices – Collaborative robots norm.

Executive elements consists of two Kuka LBR iiwa 14 R820 robotic arms equipped with universal tool mounts. It is therefore possible to mount various pneumatic and/or electric effectors. Vision system consisting of multiple cameras observes a whole scene the test stand. Kube test stand is a common workspace for humans and LBR iiwa arms. There are 8 MANTA G223C cameras, which are mounted on the edges of a cube. Maximum resolution is 2048×1088 pixels. Image is acquired via GigE interface and two quad-channel acquisition cards. One of them has additional FPGA chip which can be used to speed up basic, initial steps of image analysis. At maximum resolution, frame rate of 50 fps is achievable, although it depends on selected operation mode. Vision system is augmented with two 40 kHz lamps in order to provide constant lighting. Programmatically, test stand is aided with MVTEC Halcon 12 libraries.

Further information's about Kube test stand and its capabilities can be found in publication [3]. Additionally, the reference [13] shows possibilities of direct human-robot interaction of installed robots.

3. Experiment setup

3.1. Robotic scene and its parameters

Inside Kube robotic station a container with unsorted object and empty containers for segregated objects were placed on top of a pedestal table. The container with unsorted object was inlaid with polyurethane foam, and was placed with its wider side directed towards camera. On its opposite side three white target containers were placed. During experiment plastic balls with the same sizes and masses but different colors were used as object for sorting process.

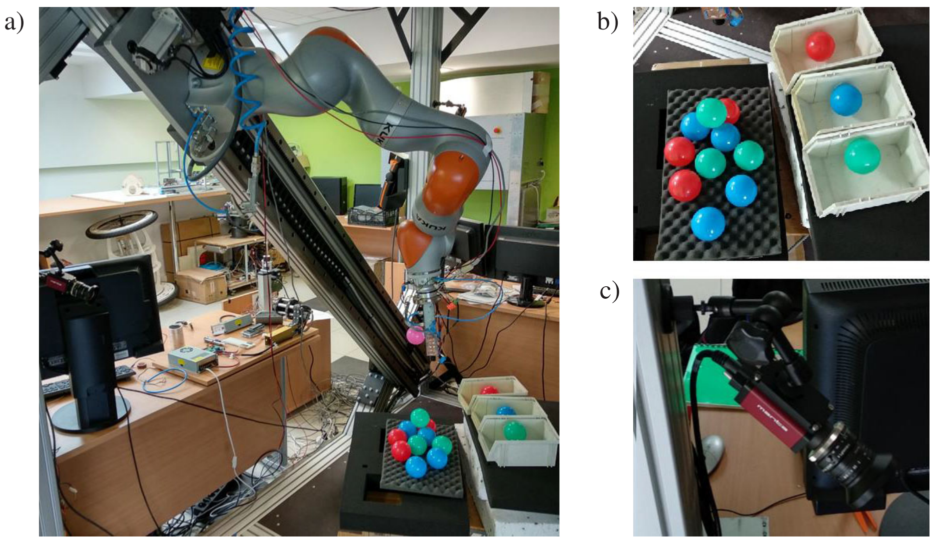

Figure 1: Robotic test stand. a) general overview, b) unsorted container (left), target containers (right), c) Manta G-223C camera

Area of the container with unsorted object were 380 mm × 230 mm, top of each target containers size was 190 mm × 125 mm.

3.2. Sorting method

In order to develop human-robot collaboration in environment with multiple objects, certain experiments were conducted. The goal was to development a method of joining the robot into the object sorting process performed by human, which means recognize sorting method and support or fully replace human in performing it. This generally stated objective causes that system controller should perform the following tasks:

- recognition and listing of features of multiple objects placed on robotic scene with position estimation of selected object in a chosen reference frame;

- operator's hand detection;
- release phase of a grabbed object detection and position recording at this moment;
- robot's tool tracking;
- control of robotic arm movement which means guiding effector to position where it is possible to grab an object, to move the object over a correct container and to release the object.

Additionally system must be able to perform searching for object repeatedly until all objects are placed in designated containers.

The above requirements were reasons for introduction of some simplifications. It was assumed that objects have simple geometrical shapes and identical dimensions – the only distinguishing feature differing between objects was color. This paper refers to results of experiments where differently colored balls with diameter of 60 mm served as sorted objects. A pneumatic sucker with a micromechanical grasp detector was used for grabbing and carrying. These simplified assumptions allowed to concentrate on designing precise experiment scenarios, creating dedicated and simple to use tools for scene analysis with use of only one camera and developing rules according to which robot controller executes tasks.

Robot control program is a finite-state machine with 6 states and 1 input signal. In the present design there are multiple subprocesses of application which are the source of an input signal. However outputs of all of them are stacked on single common message queue which is a sole input of the finite-state machine. There are 6 states:

- S1 – "Direct control" – Initial states. Manipulator executes relative moves based on commands from vision system in base coordinates frame.
- S2 – "Grabbing" – Lowering tool in a straight line with the pneumatic gripper activated until object is grabbed and moving back to specified working height.
- S3 – "Waiting" – State of waiting after action is finished.
- S4 – "Carrying" – State following picking up of a ball when target drop destination is known. In this state the robot automatically moves to dropping position and releases a ball.
- S5 – "Learning" – Robot moves the same way as during "direct control" but the grabber is activated. When the state is finished the dropping position is recorded.
- S6 – "Releasing" – State following "learning", during this stage a ball is released and dropped, then the robot moves back over container with unsorted balls.

Input signal consists of two elements – label and additional data. Taking movement signal as example – label is recognized as "go to" command and offset value is passed as additional data. Since not all signals require additional information only label is mandatory and additional data is optional. Bellow all possible values of input signal are listed. For clarity of description each command has numerical value assigned:

1. "Grab" – starts grabbing procedure.
2. "Action finished" – informs about finishing of executed action e.g. picking ball.
3. "Known color ID" – informs that color ID is known and has drop position assigned to it, color ID determines target container.
4. "Unknown color ID" – informs that color ID is not known i.e. teaching procedure was not performed for it.
5. "Drop" – release command, turns off pneumatic grabber and records drop position during teaching procedure.
6. "Go to" – relative movement command, relative offset is passed as additional values x, y, z in base coordinates frame.

Fig. 2 describes machine workflow. In the graph there are two cycles C1 and C2 consisting respectively of states S1–S2–S3–S4 and S1–S2–S3–S5–S6. Cycle C1 is a natural cycle of learned machine, whereas C2 describes learning process. Machine executes cycle C2 once for each ball color.

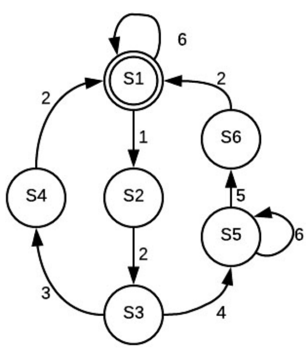

Figure 2: Graph representation of automata

3.3. Learning algorithms

Learning procedure has 2 stages: first stage consist of recording sorting method by means of the vision system, second stage is performed during sorting and comes down to transmitting the sorting method to robot controller. Second stage consists of execution the cycle C2.

During the first stage the vision system recognizes method of sorting. For each color of a ball the human operator shows segregation method by picking up a ball, carrying it over the chosen container and dropping it into the container. The visions system records these actions and calculates final position of a sample ball. This final position is used as reference position for calculation of drop position of the ball which is placed straight above reference position with configurable offset.

Both sorting and learning start in state S1. Effector is guided to position above chosen ball by sending command 6 multiple times, when a tool is in correct position the command 1 – "grab" is sent and machine switches state to S2. After grabbing the command 2 is send to confirm that action is finished and machine state changes to S3, where it awaits information about color of a picked ball. Receiving signal 4 causes changing state to S5, in this state the robotic arm is guided over container assigned to color of picked ball through repetitive sending of movement command (6). When the tool is located in correct position, signal 5 is sent and drop position for current color of object is recorded and machine state is changed to S6. In state S6 pneumatic grabber turns off and manipulator moves back over container with unsorted balls. When signal 2 is received machine goes back to state S1. It finishes the cycle C2 and learning procedure for one color.

3.4. Development tools

Image recognition system has been created using C# language, EmguCV version 3.1.0 library and Visual Studio 2015 environment. Camera settings were changed using GenICam Explorer version 5.4.1.2 program. Connection with camera was established by using libraries delivered with Halcon environment.

Robot software was created in Sunrise Workbench environment and was written in Java. Robot is controlled in SmartServo mode provided by manufacturer which enables asynchronous execution of movement commands. Asynchronism here is taken as possibility to send commands with irregular time intervals and to interrupt currently executed move if new command arrives. Image recognition system and robot controller communicate with UDP. Commands are being sent asynchronously, with confirmation of completion.

3.5. Selected procedures in experiments

Object detection

Image is retrieved from camera with resolution set to 1280×720. Retrieved image is converted from RGB to HSV color space. Converted image is thresholded as many times as there are planned ball colors and result for each color is stored in distinct image buffer. Furthermore, detected areas are filled which will be crucial for raycast algorithm.

Ball detection and measurement is based on the results of raycast algorithm performed on each ball individually. For object such as balls, their center of gravity is optimal starting position for raycasting. Decision was made to be able to provide them even then, when objects slightly overlap. It was accomplished with thresholded distance transform with L2 norm which has the ability to separate adjacent objects by shrinking. Then it is assumed that centers of gravity of shrunken objects are in close proximity to the centers of balls they represent and can be used as starting points for raycasting. Raycasting is performed on filled image before applying distance transform. After that, ray array is sorted in ascending order and around 50% of rays from the middle are used to calculate mean length. Mean length represents the radius of ball as seen in current frame and is vulnerable to distortions. To decrease the magnitude of errors, new radius length is averaged with lengths memorized during previous cycles.

For each detected object, center, radius and color identifier are stored.

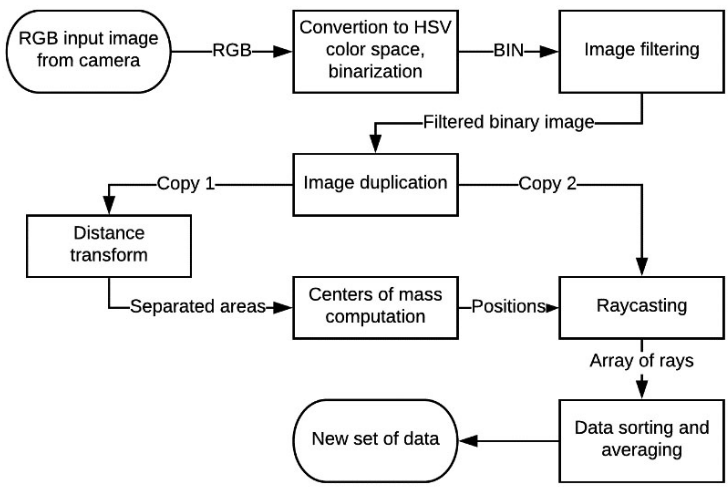

Figure 3: Operation diagram of the object recognition algorithm

Multiple objects detection

After every cycle of algorithm, new set of data is compared with the previous set. For each color individually, all possible distances between centers of newly and previously detected objects are calculated. If a nearby old object is found, then new object is treated as translated old object. The ability to preserve the context of each object enables its tracking even then there are many similarly looking objects. Previously mentioned distance transform has the capability to separate slightly overlapping objects which enables multiple detection.

Additionally, algorithm detects collision when objects of the same color overlap. Overlapping is defined as distance between objects centers being less than sum of their radiuses. Collisions, as well as other cases of overlapping are resolved by specifying grabbing order.

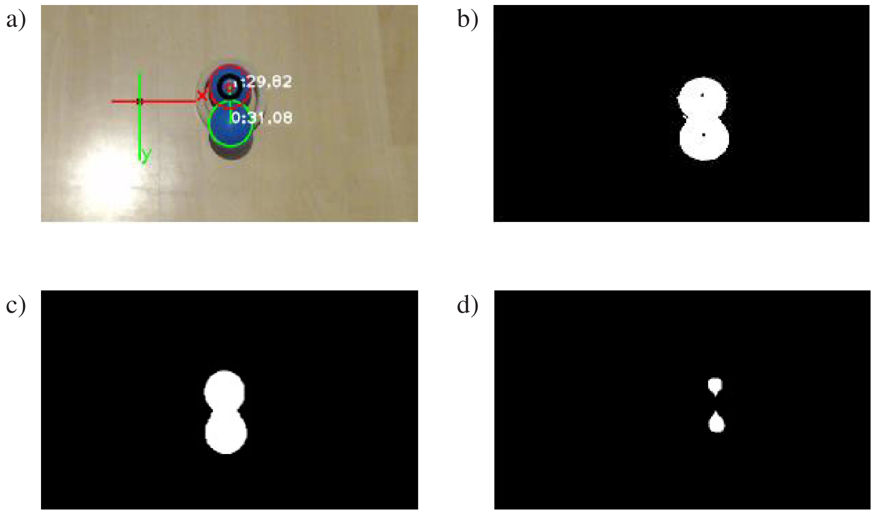

Figure 4: Separation of overlapping objects: a) input image with markers, b) binary image, c) filtered and filled binary image, d) distance transform result: approximate position of object centers

Grabbing order depends on calculated ball radiuses and they are picked in descending order. Longest radius corresponds to ball being closest to the camera which eliminates the risk of being blocked out by another ball. This strategy is capable of unveiling remote objects gradually and resolving collisions. That makes this approach valid during whole sorting process.

Determining object positions in space is based on assumption that all balls have the same radiuses, which is known and constitutes the first of three calibrating parameters. The second and the third parameters are established during calibration process, which consists in putting a ball in front of the camera at a known distance and retrieving radius in pixels as seen by the camera. Calibration distance and radius are the second and the third parameters.

Measurement and filtering of the calculated radius is possible because the shape of a ball casted on an image is equivalent of a cross-section going through the center of said ball. Because of that, center and radius of the ball can be calculated as the center and radius of a circular cross-section visible on an image. Utilizing calibration data and simple geometry, position in camera coordinate system can be computed.

Transformation from camera to robot coordinate system is determined during the station calibration procedure which produces transformation matrix containing parameters of transformation.

Correct computation of translation of a suction cup over the targeted ball requires only knowing the position of the marker ball (recognized by distinct color) attached to the robot and the targeted ball. It is assumed that robotic tool orientation does not change and the only way to pick up the object is to do it vertically from above. Additional translation vectors, describing translations between marker and center of suction cup and targeted height over the center of the ball before triggering grabbing procedure, are predefined constants.

4. Accuracy measurements

4.1. Test setup

In order to achieve reliable operation, estimation of object position has to be as accurate as possible. To measure it, comparative experiment has been conducted. Using the same hardware setup, results of proposed ball location algorithm has been compared to the results given by ARTag markers [2].

Firstly, eight testing points has been selected inside the test bench and location of origin of the reference coordinate system has been selected. Next step consists of placing balls or ARTags on testing points and retrieving their locations as seen by camera in camera coordinate frame. Three scenarios has been compared: proposed algorithm without and with camera calibration in order to remove distortion and ARTag markers position detection where camera calibration is mandatory.

Secondly, received positions has been transformed to reference coordinate frames and compared with known test points locations in order to determine accuracy.

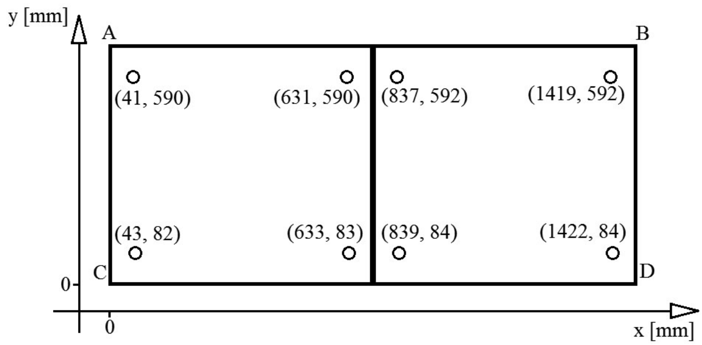

Figure 5: Test points locations and reference coordinate system

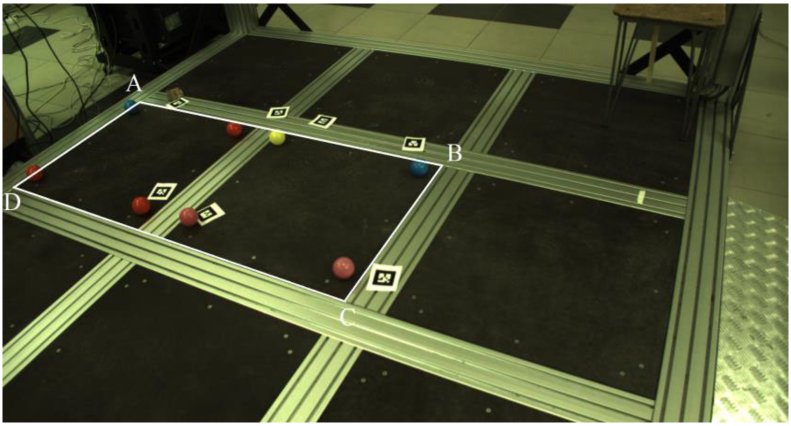

Figure 6: Localization of objects on real test stand, balls are placed at points and ARTag markers are visible

4.2. Position transformation

Ball position is estimated in camera Cartesian coordinates frame. The transformation from camera to reference coordinate frame is a homogeneous transformation. This is shown in Eq. (1).

$$\begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'_i \\ y'_i \\ z'_i \\ 1 \end{bmatrix}, \tag{1}$$

where: x_i, y_i, z_i – position in reference coordinates frame, x'_i, y'_i, z'_i – position in camera coordinates frame.

Equation for transformation of each coordinate has form of a linear function of variables x', y', z'. For this experiment homogeneous transformation constraints relating to rotation matrix were removed in order to simplify linear optimization problem and to make the result as accurate as possible. Additionally we do not need the value of z-direction, since it is not used in the robot control algorithm. As a result this equation can be reduced to optimization problem shown below in Eq. (2).

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix} \begin{bmatrix} x'_i \\ y'_i \\ z'_i \\ 1 \end{bmatrix}. \tag{2}$$

Error e was defined as a sum of square differences for each point as in Eq. (3).

$$e = \sum \left((x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right). \tag{3}$$

The optimization problem was solved using Nedler-Mead algorithm.

4.3. Results

During the experiments 3 different sets of measurements were taken:

- balls positions with uncalibrated camera,
- balls positions with calibrated camera,
- ARTag marker position.

Both sets of balls positions were taken for exactly the same robotic scene with the only difference being distortion removal step in image processing algorithm. The last set was taken as reference with use of ARTags markers which are designed and used for purposes of visual localization. For tag localization ROS ar_track_alvar package was used, which can calculate marker position in camera coordinates frame based on tag parameters and image data.

For each set we performed optimization and computed transformation parameters. Values of maximum and average displacement between real position and calculated position in reference frame are presented in table.

Table 1: Displacement calculation results

	Uncalibrated camera	Calibrated camera	ARTag
e_{max} [mm]	74.0	19.9	18.4
e_{avg} [mm]	38.1	11.7	8.4

As it is shown in the table: measured errors without camera calibration are much bigger with average of 38.1 mm and maximum value of 74.0 mm. Taking into account the results with camera calibration the maximum displacement is comparable with value measured for ARTag markers, however the average value is higher.

It is important to note that sorted objects are placed in much smaller area due to robot movement ranges, so the transformation can be calculated only locally for this area which should result in increased accuracy.

5. Summary

The described system is still in early stages of development and serves as backbone for further development enabling more complex tasks. Conducted experiments confirmed that initial assumptions and resulting system architecture were satisfactory. Moreover, effectiveness of human-robot interaction in this model case was checked and the assumption that such system would be useful in monotonous tasks, such as recycling or packing, was made. The achieved accuracy proved to be satisfactory, although it is slightly worse than marker accuracy.

Further stages of development will focus on improving various subsystems, most notably, better teaching methods. Current state of the art increasingly utilizes methods based on machine learning which often yield very promising results.

Good review of the current state of the art is the article by Soni Chernova and Andrei Thomaz [1]. In many cases, identifying correct structure of a complex task is an exceptionally hard task. Our team will also explore this topic in future work.

References

[1] S. Chernova and A.L. Thomaz: *Robot Learning from Human Teachers Series: Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool Publishers, 2014.

[2] M. Fiala: ARTag, a fiducial marker system using digital techniques. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego, California USA, 20–25 June 2005, Vol. 2, 590–596.

[3] G. Granosik *et al.*: *Kube – platforma robotyczna dla badań naukowych i prac wdrożeniowych*. In: Postępy robotyki, tom I. Red. K. Tchoń, C. Zieliński. Warszawa, Oficyna Wydawnicza Politechniki Warszawskiej 2016, 224–234 (in Polish).

[4] S.A. Green *et al.*: Human-Robot Collaboration: A literature review and augmented reality approach in design. *Int. J. of Advanced Robotic Systems*, **5**(1), (2008), 1–18.

[5] Y. Jiang, C. Zhang and M.A. Saxena: *Learning to place new objects*, 2012, ICRA Conference, 3088–3095.

[6] H. Kjellsrom, J. Romero and D. Kragic: Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding*, **115** (2011), 81–90.

[7] H.S. Koppula, R. Guthra and A. Saxena: Learning human activities and object affordances from RGB-D videos. *International Journal of Robotics Research* (2013), 951–970.

[8] M. Kyrarini *et al.*: Robot learning of object manipulation task actions from human demonstrations. *Mechanical Engineering*, **15** (2017), 217–229.

[9] Q. Li, M. Wang and W. Gu: Computer based system for apple surface defect detection, 2002, Computers and electronics in Agriculture 36, 2021, 215–223.

[10] J. Patel, Y.A. Choudhary and G.M. Bone: Fault tolerant robot programming by demonstration of sorting tasks with industrial objects. *2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, 278–283.

[11] Seema, A. Kumar and G.S. Gill: Automatic fruit grading and classification system using computer vision: A review. *2015 Second International Conference on Advances in Computing and Communication Engineering* (2015), 598–603.

[12] S. Wang *et al.*: An application of vision technology on intelligent sorting system by delta robot. *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 1–6.

[13] https://www.kuka.com/en-us/technologies/human-robot-collaboration.