

## Automated recognition and sorting of agricultural objects using multi-agent approach

Kateryna Ovchar, Andrii Borodin, Ivan Burlachenko<sup>[0000-0001-5088-6709]</sup>  
and Yaroslav Krainyk<sup>[0000-0002-7924-3878]</sup>

Petro Mohyla Black Sea National University, 68, Desantnykiv Str., Mykolaiv, 54000, Ukraine  
 {kvchar80, electron.t5164, ivan.burlachenko2010}@gmail.com,  
 yaroslav.krainyk@chmnu.edu.ua

**Abstract.** In the following paper, we propose an alternative approach to automatic recognition and sorting of the agricultural objects. The objects are sorted, based on their color. The sorting algorithm works with pictures, captured by a web camera, pre-processes them, determines the mass centers of the recognized objects, and calculates distances to the points which were found. Afterward, the recognized objects are sorted by a hand-manipulator with four degrees of freedom. The manipulator is powered by Arduino Uno Rev 3 and servo motors of two types. The sorting algorithm has been implemented in Python, with the use of the OpenCV library. For image pre-processing, the spatial anti-aliasing, filtering, and morphological opening and closing have been used. Color maps have been created to determine the mass centers of the recognized objects. To calculate the rotation angle of the manipulator's elbow based on the shoulder position, linear regression model is trained. The practical value of the research and development results is the possibility of their use in the harvesting of the agricultural objects with further extend to manufacture with conveyor sorting tapes.

**Keywords:** automatic harvest sorting, robotic arm, OpenCV, Arduino, servo drives, multi-agent system.

### 1 Introduction

As one of the key branches of the industry of Ukraine, agricultural industry aims to deliver the high-quality goods with lower self-costs, while also reducing the time spent on their production. Replacing the manual labor with the machine labor is an effective way of solving this problem. A human laborer cannot perform repetitive tasks for a long time and stay effective for long.

Since harvesting is one of the primary parts of all agricultural processes, automating this task has been a researchers' priority since the 1960s. However, automated harvesting has several complications to avoid. Accurate recognition of objects and avoiding their damage during the gathering process are the main problems for every researcher aiming to automate this procedure.

One of the first researches devoted to automatic harvest recognition and gathering was performed in 1968 [11]. The authors were comparing the lengths of the waves of light, which were reflected from the fruitage and leaves on the orange trees. When the photographic equipment became more affordable, similar researches started analyzing images of the fruit instead of the objects' surfaces.

Since the problem of the automatic agricultural objects' sorting comprises both object recognition and capturing, it encounters the same complications as the automatic harvesting task does. Nowadays, there are several approaches to the automatic sorting of the agricultural objects – magnetohydrodynamic installations [10], Bayesian classifiers [4], machine vision modules [6]. However, most of these methods don't take into consideration the physical properties of the objects, or are expensive to set up and maintain.

In this work, we propose a comparatively low-complexity and effective approach to the automatic agricultural objects' sorting.

## **2 Analysis of scientific sources and hardware solutions**

In this section, we will consider some of the existing methods of the automatic sorting of the agricultural objects, based on the machine vision.

### **2.1 Analysis of the existing methods of machine vision-based agricultural objects recognition**

Most of the modern researches use images, captured by CCD cameras, for object detection and recognition. Some of them rely on the fuzzy logic algorithms [2]. For object borders detection, the thresholding method was used. Afterwards, the determined objects were sorted by color (97,8% accuracy) and size (88,9% accuracy) by calculating the degree of compliance between the objects.

Bayesian classifiers [9] are also widely used for automatic agricultural objects recognition and sorting. The borders of an object were determined by the method of finding the shortest path on the graph. Then the objects were classified by ripeness, taking into consideration the average values of the R, G and B channels on the image, along with their gradients. For this method, the accuracy of sorting by ripeness is 92,2%.

Clustering methods [7] are another one popular approach to automatic sorting of the agricultural objects. After obtaining the image of an object, its projection on the X axis was made. The points with the lowest color saturation determined the borders of an object. Then the areas of a particular color are determined within each object, and the objects are clustered with a k-means clustering method. The number of clusters is determined by the user.

Neural networks [5] represent the most complicated but the most promising approach to the automatic recognition and sorting of the agricultural objects. Having separated the image dataset in 3 classes, researchers trained a three-layer neural network, which is able to sort objects on images by their brightness (color) and size.

Considering these features, the neural network assigned each object to one of the 3 classes: low-quality fruit, medium-quality fruit, and high-quality fruit. The classification accuracy was up to 80% [5].

All the methods listed above are effective but quite significant amount of time and resource consuming. In this work, we propose an alternative approach to recognition and sorting of the agricultural objects.

## 2.2 Analysis of the existing hand-manipulator models

There are several criteria for choosing the hand-manipulator prototype:

- a convenient gripper, that would not damage the captured objects;
- minimum 4 degrees of freedom;
- Arduino-compatibility (for reducing the cost of the project).

Wireless Robot Arm [1] is an Arduino-compatible model, but is powered by its own custom secured microcontroller. The manipulator has four degrees of freedom, and its gripper is accustomed to the harvesting tasks. But the parts of the robot arm are made of metal, which is quite an expensive option, if compared to 3D-printing or acryl.

Niryo One from Niryo is an analog of an industrial robot, constructed for learning purposes. It has six degrees of freedom and is made of 3D-printed details. Niryo One is an open-source model, and has its own IDE with a drag-and-drop editor, which makes a model available for all users. The manipulator can be powered by both Arduino and Raspberry PI.

The model also has several grippers which can be mounted into it. But the huge variety of details makes Niryo One complicated to construct and maintain.

Dexter by Haddington Dynamics [3] is a 3D-printed manipulator with 7 degrees of freedom. Powered by the HD Hi-Q CPU, it allows to run the computer vision, neural network training, and signal processing algorithms right on the manipulator. The processing unit also allows to run distributed applications on it. Needless to say, that such a powerful setting is hard to reproduce.

The uArm by UFACTORY is powered by Arduino MEGA 2650 and has four degrees of freedom. Each of the joints can rotate on 180°. The gripper has a round form and is accustomed for capturing round objects. The model is an open-source project, so the manufacturer provides the drawings of the details, which can be carved from acryl to reduce the budget of the project.

Having compared the above-mentioned manipulator models, we decided to perform the reverse engineering of uArm by UFACTORY. This hand-manipulator can lift objects up to 0.5 kg in weight, has a 320 mm working area, and is Arduino-compatible by default.

## 3 Results

In this work, the hand-manipulator, powered by Arduino Uno Rev3 and two types of TowerPro servomotors, is used. The object recognition algorithm implemented in

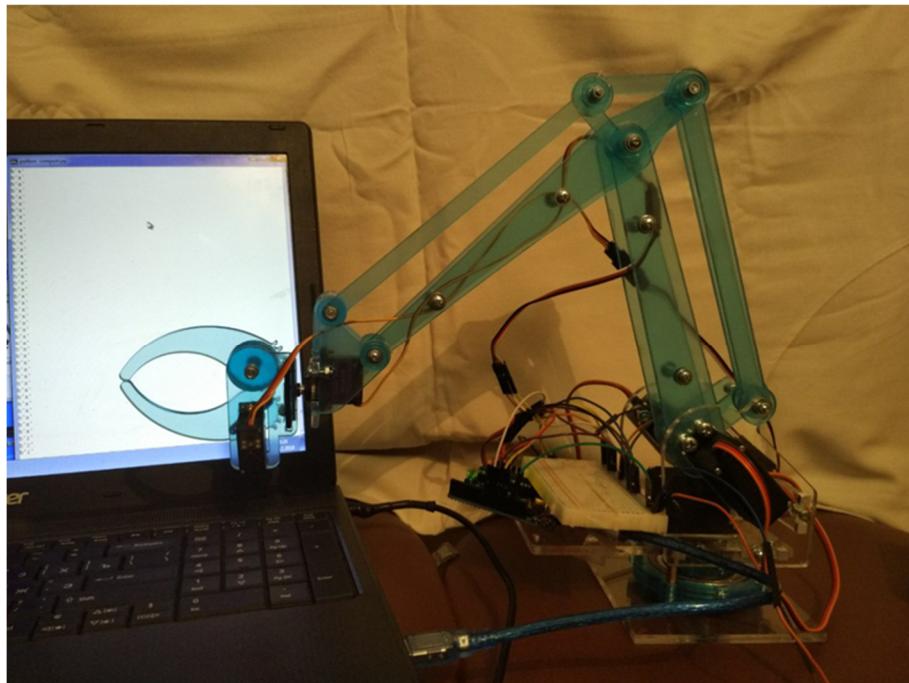
Python with the use of OpenCV library, determines the rotation angle for the manipulator's shoulder and elbow. The manipulator rotates on the appropriate angle, captures the object, and puts it either on the left or on the right, based on its color.

### 3.1 Hand-manipulator construction and controlling

The constructed manipulator has the following characteristics:

- height – 300 mm;
- working area – 140-300 mm;
- carrying capacity – 200 g;
- input current – 6 A;
- board – Arduino Uno Rev 3;
- connection method – MicroUSB.

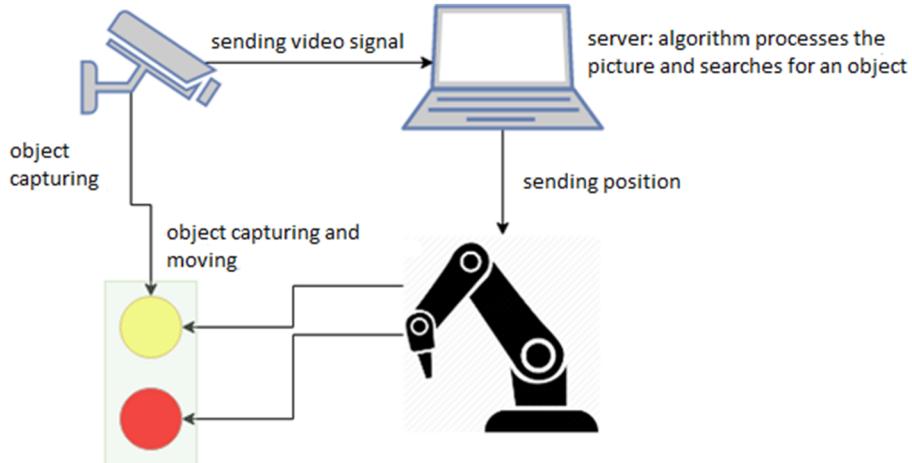
Fig. 1 shows the constructed hand-manipulator.



**Fig. 1.** The constructed hand-manipulator.

All the movable joints of the manipulator have bearings on them, which makes the moves faster and more coordinated. More powerful servomotors (TowerPro MG995) are located in the base, shoulder and elbow of the manipulator, which makes the construction more stable.

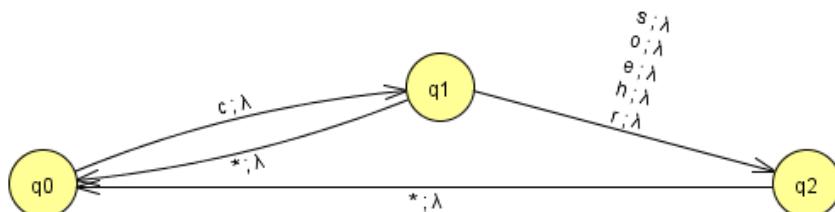
The function of Arduino Uno Rev 3 lies in getting the value of the rotation angle for the manipulator's shoulder through the COM port, and sending the respective commands to the servomotors. The Arduino is connected to a computer where the sorting algorithm runs, and is powered by it. For servomotors, the external source of power is used (5 V, 5 A). Fig. 2 shows the connection scheme of the overall complex.



**Fig. 2.** Connection scheme of the sorting complex.

Five servomotors of two types run the hand-manipulator. Three TowerPro MG995 servomotors are placed in the base, elbow, and shoulder of the manipulator, with the torque of  $10 \text{ kg} \cdot \text{cm}$  with the 6 V power supply. The rotation speed with such a power supply is  $0,16\text{s}/60^\circ$ . The gripper opening-closing and rotating are performed by the TowerPro MG90S servomotors. Their torque is  $2.2 \text{ kg} \cdot \text{cm}$  with the 6V power supply, and the rotation speed is  $0,08\text{s}/60^\circ$ . Both models of servomotors work with any type of the Arduino output (the PWM mode is not obligatory).

The algorithm that controls Arduino can be described by a finite Mealy machine, shown on Fig. 3.



**Fig. 3.** The finite Mealy machine for controlling Arduino board, that controls the hand-manipulator. The states are:  $q_0$  – waiting for the command,  $q_1$  – reading the command,  $q_2$  – processing the command. The input signals are:  $c$  – read command,  $r$  – rotate the base,  $h$  – open/close the gripper,  $e$  – rotate the elbow,  $o$  – rotate the gripper,  $s$  – rotate the shoulder.

In case the server sends an inappropriate signal to Arduino, it just keeps waiting for the next command to be read.

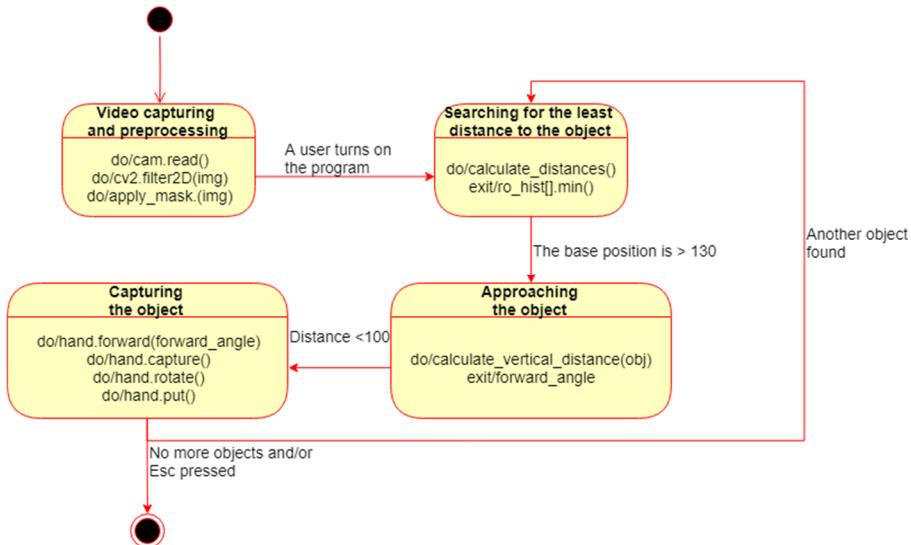
### 3.2 Object Detection Algorithm

The object detection algorithm was described on Python with the use of the OpenCV library [8] – an open library for computer vision and machine learning. The algorithm consists of the 3 key parts – image preprocessing, object detection, and object capturing and moving.

The algorithm is meant to sort agricultural objects of two colors – yellow and red. For more convenient recognition of the manipulating hand itself, the green mark was put on the gripper.

First, the algorithm looks for all red objects and moves them to the left. Then, it starts looking for yellow objects and moves them to the right. When all the objects are sorted, the algorithm stops working.

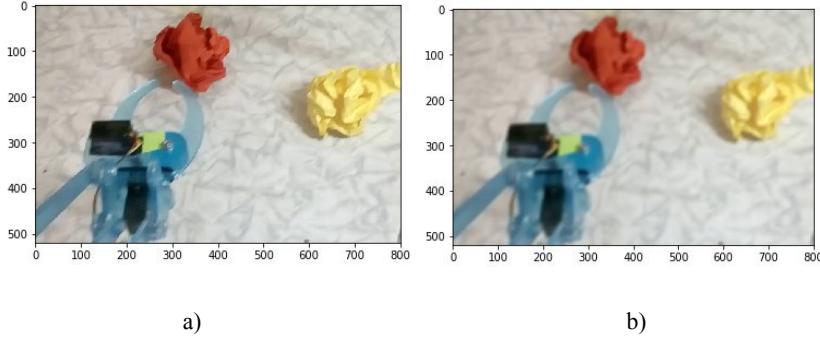
Fig. 4 shows the statechart diagram for the algorithm.



**Fig. 4.** The statechart diagram for the object recognition and sorting algorithm

To increase the algorithm performance, the image preprocessing starts even before a user starts working with a program. The image preprocessing aims to reduce noise on the image and consists of spatial anti-aliasing, filtering (color maps creation), and morphological opening/closing.

For the spatial anti-aliasing, the mean filtering method is used. The 9x9 filter was used, with the value of each cell being equal to 1/81. Since the input image had the BGR representation, the filter was put on the blue, green and red channels separately. Fig. 5 shows an image before and after the mean filtering.



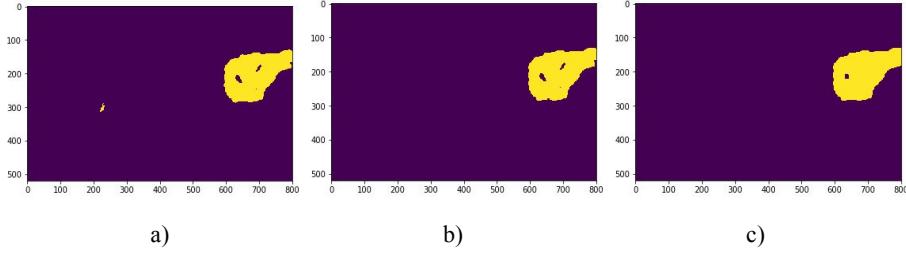
**Fig. 5.** The image captured by a web camera: a) before mean filtering b) after mean filtering

After applying the mean filter, the image was converted to the HSV-format. The model is a non-linear representation of the RGB-model.

The next stage of an algorithm is creating the color maps for yellow, red, and green colors by highlighting all the pixels of a particular color. The pixel's belonging to the following of HSV-values determined its inclusion on a particular color map:

- green – from (45, 50, 50) to (75, 255, 255);
- yellow – from (20, 100, 100) to (30, 255, 255);
- red – from (0, 70, 50) to (10, 255, 255) and from (170, 70, 50) to (180, 255, 255).

To every color map, morphological opening and closing were applied. Fig. 6 shows the outcome of these operations.



**Fig. 6.** a) initial map b) morphological opening c) morphological closing

On the resulting color maps, the mass centers of the objects are determined. For each map, both horizontal and vertical projections of the colored areas are made. Afterward, the means along each axis are calculated. The expected value of the center coordinate is calculated by the formula

$$E = \int_{-\infty}^{\infty} x \cdot f(x) | \{f(x) > q\} dx, \quad (1)$$

where  $q$  – 90% quantile.

To determine the mass centers of the objects more precisely, the exponential moving average method is used. The method is determined by the formula

$$c = c_{t-1} \cdot (1 - \alpha) + x_c \cdot \alpha, \quad (2)$$

where  $c$  — anti-aliased mass center coordinate,

$c_{t-1}$  — previous value of an anti-aliased mass center coordinate,

$x_c$  — not anti-aliased mass center coordinate,

$\alpha$  — anti-aliasing coefficient.

For the coordinates of the gripper, the moving average formula is used

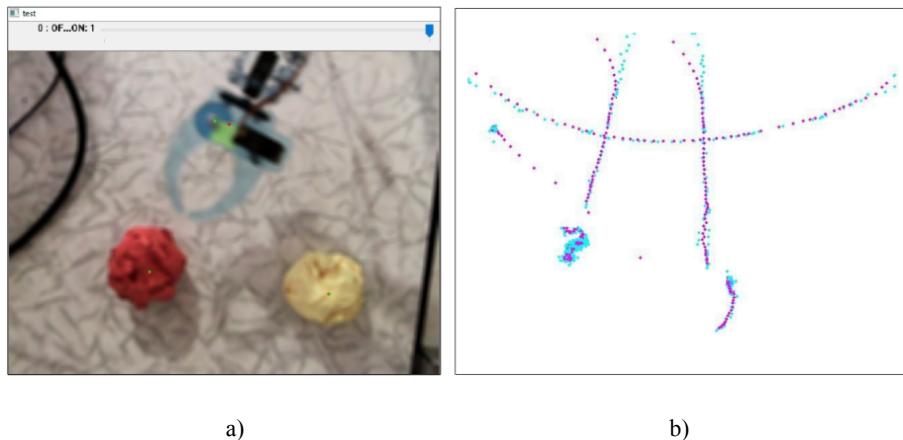
$$v = \frac{1}{n} \left( \sum_{0}^{n-1} v_{t-1} \right), \quad (3)$$

where  $v$  — anti-aliased mass center coordinate,

$n$  — anti-aliasing interval,

$v_{t-1}$  — the previous value of the anti-aliased mass center coordinate.

In the program interface, the mass centers before anti-aliasing are shown in red, while the mass centers after anti-aliasing are showed in green (see Fig. 7, a).



**Fig. 7.** a) Mass centers of the objects and b) moving trajectories of the objects during the scanning and capturing process

To determine the least distances to the objects, the manipulator scans all the area, captured by a web camera (the base rotates from 0° to 130°). On each iteration, the rotation angle of the base changes on 2°. The mass centers are re-calculated on each iteration and the Euclidian distance between the mass centers of the gripper and the object are calculated.

When the scanning process is over, the least Euclidian distance from each object is taken from the array, along with the appropriate base rotation angle. To capture the

object, the manipulator rotates on the respective angle. Fig. 7, b shows the moving trajectories of the objects during the scanning and capturing process.

During the capturing process, it is important to keep the balance between the manipulator's shoulder and elbow to keep the construction stable. For this purpose, the linear regression model was trained. To capture the object successfully, the gripper should stay approximately 2 cm high from the surface. The dataset for the linear regression model consisted of 20 combinations of the shoulder's and elbow's rotation angles, which guaranteed that the gripper stayed on the needed distance from the surface. The formula for determining the elbow position based on the shoulder rotation angle is

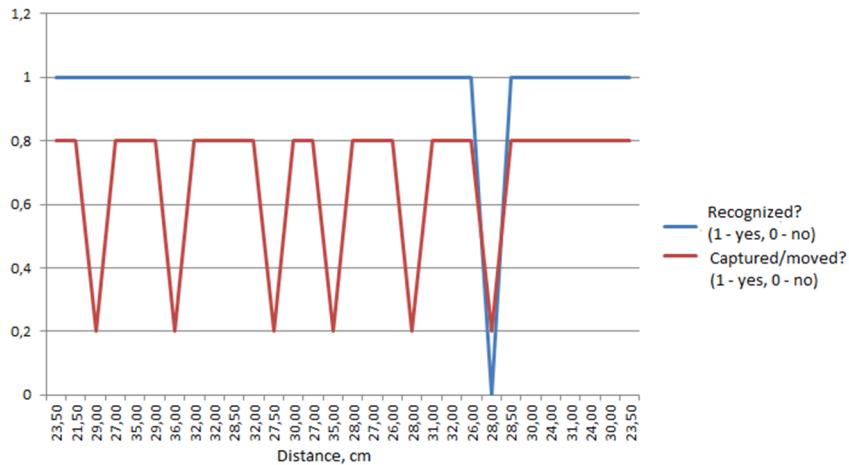
$$p_l = -13 + p_r \cdot 1,54 \quad (4)$$

When the manipulator aims to capture the object, the Euclidian distance between the objects and the gripper is calculated on each iteration. If the distance is bigger than 100, the shoulder position changes on  $1^\circ$ . The elbow position is calculated with the formula (4). When the distance becomes less than 100, the object is captured.

#### 4 Testing

The complex was tested during the test session of 30 experiments, with the objects being placed on distances between 20 cm and 30 cm from the gripper. In each experiment, the yellow and red test objects were interchanged.

Testing was conducted in the same circumstances: on the white background, under the same lighting. The starting position of the hand-manipulator was fixed, along with the position of the camera. The accuracy of the object recognition is 93,4%. The accuracy of the object capturing and moving is 78,3%. Fig. 8 shows the testing results for the red objects.



**Fig. 8.** Testing results for the red objects.

Testing showed that the distance between 26 cm and 30 cm between the manipulator and objects is optimal for successful sorting. The model of investigated multi-agent system (MAS) for efficient sorting based on the autonomous joints rotation is

$$\begin{cases} fp = a \cdot \cos\left(\frac{kb}{n}\right) + \frac{kc}{n}; \\ x_i = \sum_{m=1}^{i=j-1} \frac{1}{n} \cos(fp) + \sum_{m=j}^{i=j+r-1} \frac{1}{n} \cos(fp) + \sum_{m=j+r}^i \frac{1}{n} \cos(fp); \\ y_i = \sum_{m=1}^{i=j-1} \frac{1}{n} \sin(fp) + \sum_{m=j}^{i=j+r-1} \frac{1}{n} \sin(fp) + \sum_{m=j+r}^i \frac{1}{n} \sin(fp); \\ S_{MAS} = M(x_i, y_i, z = const). \end{cases} \quad (5)$$

where  $i$  – the infinitesimal segment along the manipulator,  $a$  – the angle of the tip of the manipulator's base with respect to the horizontal axes,  $b$  – is the number of periods in a length, and  $c$  – the direction of movements for  $S_{MAS}$  as handler function.

During the experimental tests an idea about multiple hand-manipulators combined as multi-agent system was released. According to the need for several hand-manipulators to be applied to the sorting process, the problem of intersections between them is become important. Every agent  $A_{S_{MAS}}$  should include the minimal behavior's algorithms based on the crossing collisions index  $r$  of multiple segments of the hand-manipulators.

## 5 Conclusions

In the current paper, we present the hardware and software parts of the device for automatic harvest sorting. The hardware part of the complex is the hand-manipulator with four degrees of freedom, controlled by the Arduino Uno Rev 3. The software part of the complex is the algorithm, described on Python with the use of the OpenCV library. The algorithm works with images at 10 FPS and searches for mass centers of the determined objects less than 0,08 s. The accuracy of the object recognition is 93,4%. The accuracy of the object capturing and moving is 78,3%.

The significant advantage of the system is that the hardware part doesn't damage objects during the sorting process. The disadvantages of the tested device are the low speed of the hand-manipulator, and comparatively low accuracy of the objects' capturing and moving. The system can be used on the enterprises with the conveyor tapes, used for harvest sorting.

## References

1. Burlachenko I., Zhuravskaya I., Davydenko Ye., Savinov V.: Vulnerabilities analysis and defense based on MAS method in fast dynamic wireless networks. In: Proceeding of the 4th IEEE International Symposium of Wireless Systems within the IEEE International

- Conf. on Intelligent Data Acquisition and Advanced Computing Systems (IEEE IDAACS-SWS 2018), Lviv, 20-21 Sept. 2018, pp. 98–102. IEEE (2018). doi:10.1109/IDAACS-SWS.2018.8525692
2. Chang W.-H., Chen S., Lin, S.-C., Huang P.-Y., Chen Y.-Y.: Vision based fruit sorting system using measures of fuzziness and degree of matching. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, San Antonio, 2-5 Oct. 1994, vol. 3, pp. 2600–2604. IEEE (1994). doi:10.1109/ICSMC.1994.400263
  3. DEXTER – Haddington Dynamics. <http://hdrobotic.com/dexter> (2019). Accessed 21 Mar 2019
  4. Feng G., Qixin C.: Study on color image processing based intelligent fruit sorting system. In: Fifth World Congress on Intelligent Control and Automation, Hangzhou, 15-19 June 2004, vol. 6, pp. 4802–4805. IEEE (2004). doi:10.1109/WCICA.2004.1343622
  5. Krainyk Y., Razzhyvin A., Bondarenko O., Simakova I.: Internet-of-Things Device Set Configuration for Connection to Wireless Local Area Network. CEUR Workshop Proceedings **2353**, 885–896 (2019)
  6. Kutsevol A., Lega Y., Havrysh O., Bagriy M.: Using machine vision systems for fruit/vegetable sorting automation. Cherkasy State Technological University scientific bulletin 4, 56–62 (2018)
  7. Ohali Y.A.: Computer vision based date fruit grading system: Design and implementation. Journal of King Saud University - Computer and Information Sciences **23**(1), 29–36 (2011). doi:10.1016/j.jksuci.2010.03.003
  8. OpenCV team: OpenCV. <https://opencv.org/> (2019). Accessed 25 Oct 2019
  9. Pla F., Sanchiz J.M., Sánchez J.S.: An integral automation of industrial fruit and vegetable sorting by machine vision. In: 8th International Conference on Emerging Technologies and Factory Automation, Antibes-Juan les Pins, 15-18 Oct. 2001, vol. 2, pp. 541–546. IEEE (2001). doi:10.1109/ETFA.2001.997731
  10. Renu R., Chidanand D.V.: Internal Quality Classification of agricultural produce using Non-destructive Image Processing Technologies (soft X-ray). International Journal of Latest Trends in Engineering and Technology **2**(4), 535–543. [https://www.ijltet.org/journal\\_details.php?id=895&j\\_id=2602](https://www.ijltet.org/journal_details.php?id=895&j_id=2602) (2013). Accessed 25 Oct 2019
  11. Schterz C.E., Brown G.K.: Basic Considerations in Mechanizing Citrus Harvest. Transactions of the ASAE **11**(3), 343–346 (1968). doi:10.13031/2013.39405