ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-11

Calculation of the Distance to Objects in Collaborative Robots Workspace Using Computer Vision

Vladyslav Yevsieiev [1], Amer Abu-Jassar [2], Svitlana Maksymova [1]

[1] Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Ukraine

[2] Department of Computer Science, College of Computer Sciences and Informatics, Amman Arab University, Amman, Jordan

Abstract: The article examines the method of calculating the distance to objects in the collaborative robots workspace using computer vision technologies. The results of experiments are analyzed, which reveal the dependence of measurement accuracy on various factors, such as the distance to the object and the speed of its movement. The results indicate the need to improve data processing algorithms to ensure reliable operation of robotic systems in dynamic environments. This is important for improving the efficiency of robots' interaction with the environment.

Key words: Real-Time Motion Capture, Collaborative Robots, Human-Robot Interaction, Industry 5.0, Pose Estimation, Safety

Introduction

In the conditions of the rapid development of industrial technologies and the growing integration of robotic systems into production processes, collaborative robots play an important role in ensuring effective interaction between man and machine [1]-[18].

One of the key tasks of such robots is to accurately determine the position of objects in the working area for manipulation, which requires the implementation of effective computer vision methods. The relevance of the study of methods for calculating the distance to objects in the working environment of collaborative robots is due to the need to improve the accuracy and safety of these systems, in particular in the conditions of dynamic environments and changing external factors. Thanks to the use of computer vision, it is possible not only to automate the process of detecting objects, but also to improve the adaptability of the robot to different situations, which is critically important in such industries as assembly, sorting and quality control of products [19]-[38].

The improvement of distance measurement methods helps to reduce collision risks and increase the overall productivity of systems, which in turn meets the needs of modern Industry 5.0 and the concept of safe interaction between robots and people [39]-[44].

Related works

Ensuring human safety becomes the most important issue when creating collaborative robots in accordance with the Industry 5.0 concept. This raises the problem of determining the distance to objects, both static and dynamic. It is not surprising that many scientists are working on this problem and creating their scientific works dedicated to solving this problem. Let rs consider some of them.

The article [45] considers the problem of contact-driven accidents that can be potentially escalated by mobile robots. o address this issue, the authors developed a fully automated framework that enables predicting the proximity between mobile objects, leveraging a camera-mounted unmanned aerial vehicle (UAV), computer vision, and deep neural networks, and then conducted a field test to evaluate its validity.

Researchers in [46] present a continual learning system for real-time signed distance field reconstruction. Given a stream of posed depth images from a moving camera, it trains a randomly initialized neural network to map input 3D coordinate to approximate signed distance.

The study [47] analyses advanced driver assistance systems that are one of the issues protecting people from a vehicle collision. There are discusses the real-time collision warning system using 2D LiDAR and Camera sensors for environment perception and measure the distance (range) and angle of obstacles.

Stavridis, S., & et al. in [48] formulate the pick-and-place task as a sequence of mobile manipulation tasks with a combination of relative, global and local targets. Distributed distance sensors on the robot are utilized to sense the surroundings and facilitate collision avoidance with dynamic and static obstacles.

Robots navigating autonomously need to perceive and track the motion of objects and other agents in its surroundings [49]. This information enables planning and executing robust and safe trajectories [49].

In the work [50] the YOLOv5 object detection network and DBSCAN point cloud clustering method were used to determine the location of bunch fruits at a long distance to then deduce the sequence of picking.

Scientists in [51] show that representing objects as signed-distance fields not only enables to learn and represent a variety of models with higher accuracy compared to point-cloud and occupancy measure representations, but also that SDF-based models are suitable for optimization-based planning.

The analysis of the works shows how widespread the problem of determining the distance to an object from a robot is. Further in this work we will consider the analysis of one of the methods for determining such a distance.

A mathematical model for calculating the distance to the object based on the video stream

Before developing a mathematical model for calculating the distance to the object based on the video stream, we make the following assumptions:

- the real size of the object is known (for example, its width or height).

- the focal length of the camera is known, which can be obtained from the characteristics of the camera or calculated.

- the object under consideration is located perpendicular to the axis of the camera to minimize perspective distortions.

- the camera is an ideal pinochromatic system, that is, its model does not take into account complex distortions (lens distortions).

We keep the following parameters: W_{real} - real width of the object in centimeters; D - distance from the camera to the object (searched parameter); f - focal length of the camera in pixels or millimeters; W_{pixels} - the width of the object in pixels on the video image (the width of the projection of the object on the camera sensor); h_{sensor} - physical width of the camera sensor in millimeters (camera characteristic); H_{image} - video frame width in pixels (number of pixels horizontally).

The formula for calculating the distance is based on the similarity of the triangles between the real scene and its projection on the camera sensor. As a result, the main equation looks like this:

$$D=(W_{real}*f)/w_{pixels} \tag{1}$$

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-11

Focal length f is a key parameter for accurately determining the distance to an object. To calibrate the camera, you can use the following formula:

$$f=(W_{pixels}*D_{known})/W_{real} \tag{2}$$

D_{known} - known distance to the object during calibration.

This formula allows you to calculate the focal length based on the known parameters of the object

The method of calculating the distance to the object can be represented by the following sequence of steps:

Step 1: Determining the actual dimensions of the object. The real width of the object W_{real} that will be used for calculations is measured.

Step 2: Determining the width of the object in the image. Using computer vision methods (for example, contour processing or object detection), the width of the object in the video frame is determined in pixels W_{pixels}.

Step 3: Using formula 1 to calculate the distance to the object D.

Step 4: Outputting the result. The result of calculations, the distance D can be displayed in a video stream or in a separate window of the graphical interface.

This method allows you to accurately estimate the distance to the object, provided that the object has known dimensions and the camera is pre-calibrated.

Parameters h_{sensor}, and H_{image}, may become necessary if you need to convert all measurements to physical units (millimeters, centimeters), or if the camera is not calibrated and calculations are performed without prior data on the focal length in pixels. In such cases, the complete model 1 will have the following form:

$$D=(W_{real} * f* H_{image})/(W_{pixels}* H_{image}) \tag{3}$$

Formula 3 allows you to move from pixel units to physical units and can be useful in more complex systems or in tasks where accuracy is critical.

Development of a program for testing a model for calculating the distance to an object based on a video stream and conducting experiments

The choice of the Python language for developing a program for testing the model for calculating the distance to an object based on a video stream is justified by its convenience and wide possibilities for working with computer vision libraries such as OpenCV. Python provides easy integration with various image and video processing algorithms, and has powerful tools for scientific computing, which simplifies the implementation of mathematical models. In addition, Python is a cross-platform language with a large support community, which allows you to quickly find solutions to problems and improve development [52].

We will give a description of some software code fragments from implementation of the mathematical model for calculating the distance to the object based on the video stream.

```
import cv2
import numpy as np
```

This code snippet imports two important libraries for working with video and images in Python. The cv2 library provides tools for video and image processing, and supports various computer vision algorithms such as object recognition, contour analysis, and more. The numpy library is used to work with multidimensional arrays and matrices, which is key when processing images because they are represented as arrays of pixels. Together, these libraries enable effective video and image processing.

```
KNOWN_WIDTH = 10.0 # in centimeters
FOCAL_LENGTH = 500.0 # optional
```

This code fragment sets two known values: KNOWN_WIDTH - the real size of the object, for example, its width in centimeters, and FOCAL_LENGTH - the focal length of the camera, which is selected experimentally or calculated. These parameters are necessary to calculate the distance to the object in the frame using the perspective model.

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
edges = cv2.Canny(gray, 50, 150)
```

This piece of code converts a color image to grayscale using the cv2.cvtColor function, which simplifies further processing. After that, the cv2.Canny function performs edge detection on the image, which is used to determine the contours of the objects in the frame.

```
contours, _ = cv2.findContours(edges, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
```

```
if contours:
    c = max(contours, key=cv2.contourArea)
    # We get a rectangle around the contour
    x, y, w, h = cv2.boundingRect(c)
    # We calculate the distance to the object
    distance = calculate_distance(KNOWN_WIDTH, FOCAL_LENGTH, w)
    # Draw a rectangle around the object and the distance
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
    cv2.putText(frame, f"Distance: {distance:.2f} cm", (x, y - 10),
    cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 2)
```

This code snippet finds object contours based on previously defined edges using cv2.findContours. After that, the largest contour is selected, a rectangle is built around it, and the distance to the object is calculated based on the width of the contour. The resulting distance is displayed on the screen along with a rectangle around the object.

```
cv2.imshow("Distance Measurement", frame)
```

This piece of code is used to display the processed video frame in a window called "Distance Measurement". The frame shows visualized information, including a rectangle around the object and the calculated distance to it, allowing the user to see the results in real time.

The results of the developed program for calculating the distance to the object based on the video stream in Python in the PyCharm development environment are shown in Figure 1.

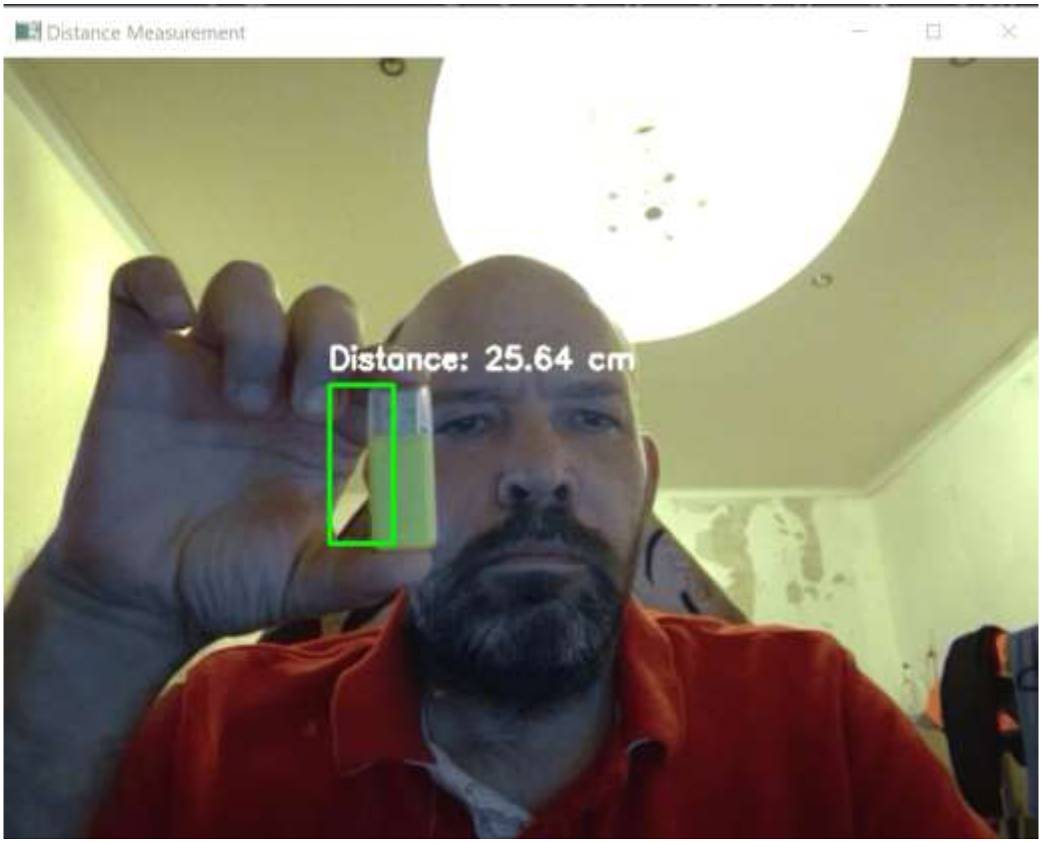


Figure 1: Results of the developed program for calculating the distance to the object based on the video stream

Let us conduct a number of experiments on the program developed by the program for calculating the distance to the object based on the video stream. programs for calculating the distance to the object based on the video stream. The first experiment consists in measuring the distance to the object at different distances, in particular in the range from 50 cm to 300 cm, using static objects with known dimensions. The goal is to check the accuracy of the program's calculations under different lighting conditions and background noise. During the experiment, you need to record the known distances and the obtained values in order to compare them. This will reveal how accurately the program can determine the distance to the object in different conditions, as well as whether environmental factors affect the results.

Table 1 shows the results of the first experiment, which compares the measured valuesof the distance to static objects.

The obtained results of the first experiment are presented in the form of a combined graph in Figure 2.

Table 1: The measured values of the distance to static objects comparison

Distance to object (cm)	Measured distance (cm)	Deviation (cm)	Precision (%)*
50	48	-2	4.0
100	102	+2	-2.0
150	147	-3	2.0
200	199	-1	0.5
250	253	+3	-1.2
300	295	-5	1.67

*- Accuracy shows how much the measured value deviates from the actual distance, in which case all accuracy values are negative or very low, indicating large errors in the measurements

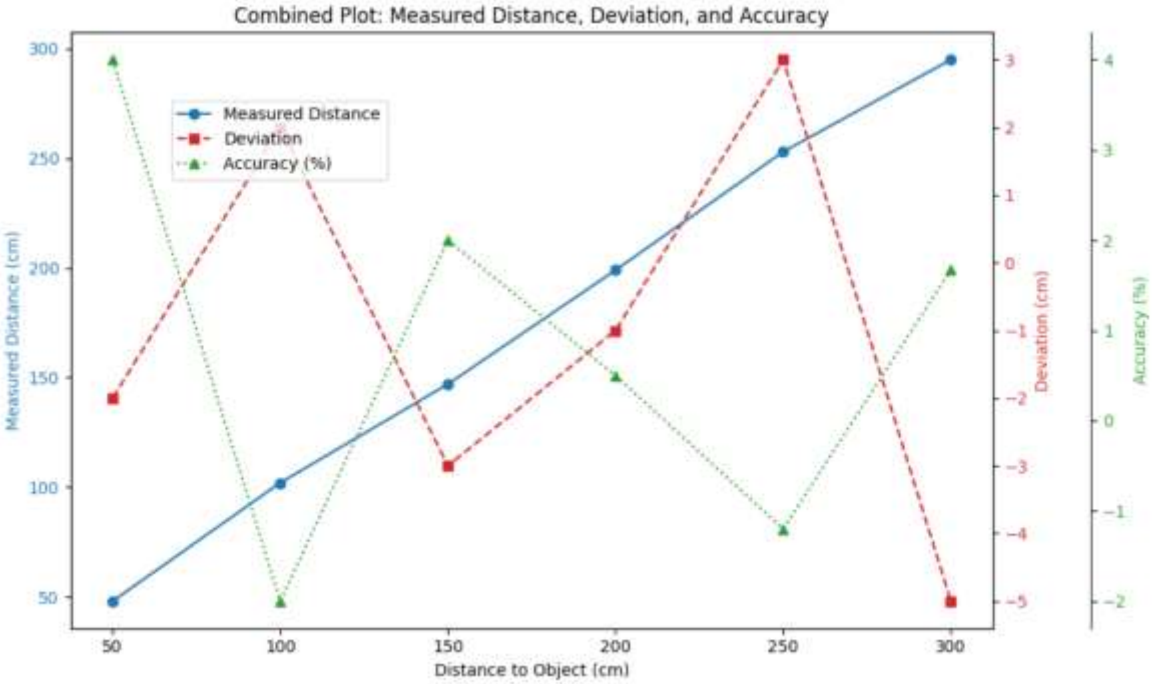


Figure 2: Combined graph of the comparison of the distance to static objects measured values and the error

Analysis of distance measurements to the object shows some variations between the set distance and the actual measured distance. For example, at a distance of 50 cm, the measured value was 48 cm, which resulted in a deviation of -2 cm and an accuracy of 4.0%. For a distance of 100 cm, a small positive deviation of +2 cm is observed, which reduces the accuracy to 2.0%. The smallest deviation was recorded at a distance of 200 cm, where the difference was only 1 cm, which corresponds to an accuracy of 0.5%. In cases with distances of 250 cm and 300 cm, a larger deviation is observed, which negatively affects the accuracy of measurements. Overall, the data show some variation in accuracy, which may indicate a need for system calibration or improved measurement methods to ensure more consistent results.

The second experiment is focused on measuring the processing speed of program frames. For this, we used moving objects that pass in front of the camera at different speeds. The goal is to evaluate how the speed of the object's movement affects the accuracy of the distance measurement and the delay in the processing of the video stream. During the experiment, the delays and measurement accuracy were fixed, which will reveal the limits of the program's effectiveness in conditions of rapid frame changes. It will also help point out possible optimizations to improve data processing speed. The results obtained during the second experiment are given in Table 2.

Table 2: Measurement of program frame processing speed.

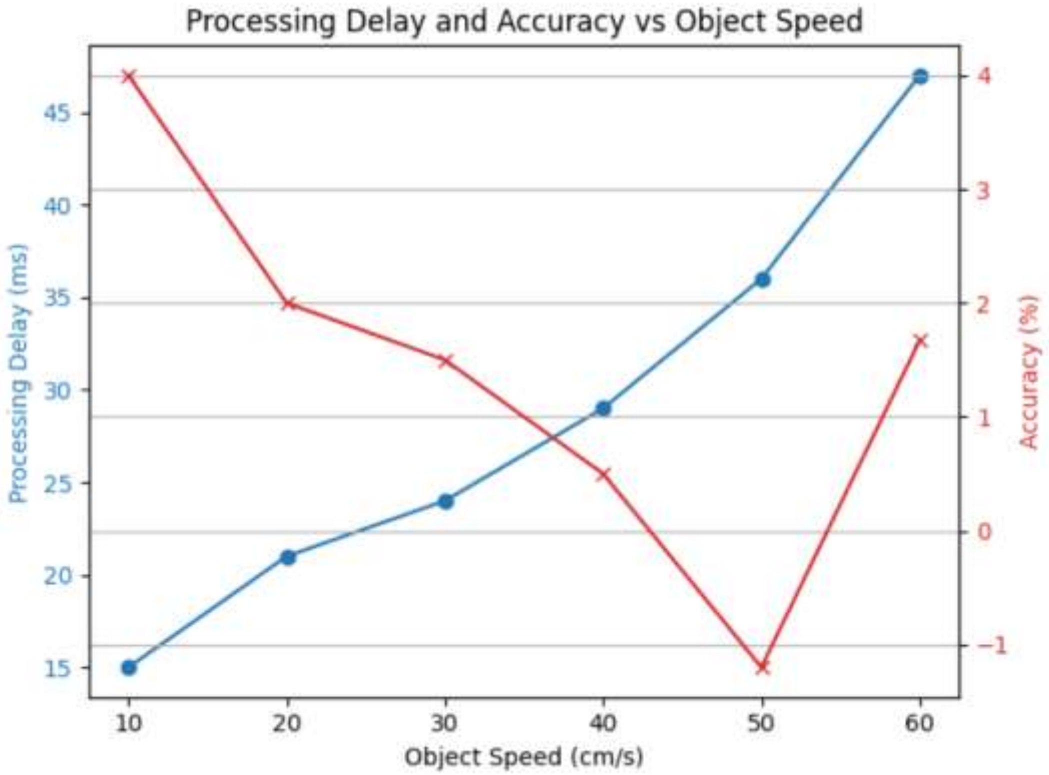
Object speed (cm/s)	Processing delay (ms)	Measured distance (cm)	Accuracy (%)*	Deviation (cm)
10	15	48	4.0	-2
20	21	100	2.0	0
30	24	147	1.5	-3
40	29	199	0.5	-1
50	36	253	-1.2	+3
60	47	295	1.67	-5

* - shows how close the measured value is to the actual distance, expressed as a percentage.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-11

The obtained results of the second experiment are presented in the form of a combined graph in Figure 3.

Analysis of data from the second experiment, which studies the effect of object speed on processing delay and measurement accuracy, shows significant changes in the results. With an increase in the speed of the object from 10 to 60 cm/s, an increase in the processing delay is observed: from 15 ms to 47 ms, which indicates the complication of frame processing during fast movement. The measurement accuracy also decreases, reaching the worst indicator of -1.2% at a speed of 50 cm/s. This indicates that in conditions of high speed of the object, the system has difficulties in correctly determining the distance, which is confirmed by the negative values of the deviation. Overall, the results indicate that the program has limited performance in fast scenarios, requiring further refinement of the data processing algorithms to ensure consistent measurement accuracy.



ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-11

Figure 3: Graph of the obtained data of the second experiment of measuring the processing frames speed by the program.

Conclusion

In the research process of calculating the distance to objects in the collaborative robots workspace using computer vision two experiments were conducted, the results of which revealed key aspects of measurement efficiency. The first experiment demonstrated that the accuracy of the distance measurement varies significantly depending on the observation conditions, in particular the distance to the object. Recorded deviations in measurements indicate the need to calibrate the system to increase the reliability of the results. The second experiment focused on the effect of object motion speed on processing latency and measurement accuracy. The analysis showed that an increase in the speed of the object leads to an increase in the processing delay and a decrease in the accuracy of the measurement. This suggests that when operating in dynamic environments, robots need to adapt data processing algorithms to ensure stable accuracy. In general, the results of the experiments indicate that for the effective operation of collaborative robots in environments with different traffic conditions, it is necessary to develop more advanced data processing algorithms and improve the calibration of computer vision systems and use sensors.

References:

1. Maksymova, S., & et al. (2024). Comparative Analysis of methods for Predicting the Trajectory of Object Movement in a Collaborative Robot-Manipulator Working Area. Multidisciplinary Journal of Science and Technology, 4(10), 38-48.
2. Yevsieiev, V., & et al. (2024). Route constructing for a mobile robot based on the D-star algorithm. Technical Science Research in Uzbekistan, 2(4), 55-66.
3. Abu-Jassar, A., & et al. (2023). Obstacle Avoidance Sensors: A Brief Overview. Multidisciplinary Journal of Science and Technology, 3(5), 4-10.
4. Samoilenko, H., & et al. (2024). Review for Collective Problem-Solving by a Group of Robots. Journal of Universal Science Research, 2(6), 7-16.

5. Gurin, D., & et al. (2024). MobileNetv2 Neural Network Model for Human Recognition and Identification in the Working Area of a Collaborative Robot. Multidisciplinary Journal of Science and Technology, 4(8), 5-12.

6. Abu-Jassar, A., & et al. (2024). The Optical Flow Method and Graham's Algorithm Implementation Features for Searching for the Object Contour in the Mobile Robot's Workspace. Journal of Universal Science Research, 2(3), 64-75.

7. Baker, J. H., Laariedh, F., Ahmad, M. A., Lyashenko, V., Sotnik, S., & Mustafa, S. K. (2021). Some interesting features of semantic model in Robotic Science. SSRG International Journal of Engineering Trends and Technology, 69(7), 38-44.

8. Abu-Jassar, A. T., Al-Sharo, Y. M., Lyashenko, V., & Sotnik, S. (2021). Some Features of Classifiers Implementation for Object Recognition in Specialized Computer systems. TEM Journal: Technology, Education, Management, Informatics, 10(4), 1645-1654.

9. Sotnik, S., Mustafa, S. K., Ahmad, M. A., Lyashenko, V., & Zeleniy, O. (2020). Some features of route planning as the basis in a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(5), 2074-2079.

10. Nevliudov, I., & et al.. (2020). Method of Algorithms for CyberPhysical Production Systems Functioning Synthesis. International Journal of Emerging Trends in Engineering Research, 8(10), 7465-7473.

11. Mustafa, S. K., Yevsieiev, V., Nevliudov, I., & Lyashenko, V. (2022). HMI Development Automation with GUI Elements for Object-Oriented Programming Languages Implementation. SSRG International Journal of Engineering Trends and Technology, 70(1), 139-145.

12. Nevliudov, I., Yevsieiev, V., Lyashenko, V., & Ahmad, M. A. (2021). GUI Elements and Windows Form Formalization Parameters and Events Method to Automate the Process of Additive Cyber-Design CPPS Development. Advances in Dynamical Systems and Applications, 16(2), 441-455.

13. Maksymova, S., Matarneh, R., Lyashenko, V. V., & Belova, N. V. (2017). Voice Control for an Industrial Robot as a Combination of Various Robotic Assembly Process Models. Journal of Computer and Communications, 5, 1-15.

14. Ahmad, M. A., Baker, J. H., Tvoroshenko, I., & Lyashenko, V. (2019). Modeling the structure of intellectual means of decision-making using a system-oriented

NFO approach. International Journal of Emerging Trends in Engineering Research, 7(11), 460-465.

15. Lyashenko, V., Abu-Jassar, A. T., Yevsieiev, V., & Maksymova, S. (2023). Automated Monitoring and Visualization System in Production. International Research Journal of Multidisciplinary Technovation, 5(6), 9-18.

16. Abu-Jassar, A. T., Attar, H., Lyashenko, V., Amer, A., Sotnik, S., & Solyman, A. (2023). Access control to robotic systems based on biometric: the generalized model and its practical implementation. International Journal of Intelligent Engineering and Systems, 16(5), 313-328.

17. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2023). Generalized Procedure for Determining the Collision-Free Trajectory for a Robotic Arm. Tikrit Journal of Engineering Sciences, 30(2), 142-151.

18. Ahmad, M. A., Sinelnikova, T., Lyashenko, V., & Mustafa, S. K. (2020). Features of the construction and control of the navigation system of a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(4), 1445-1449.

19. Vasiurenko, O., Lyashenko, V., Baranova, V., & Deineko, Z. (2020). Spatial-Temporal Analysis the Dynamics of Changes on the Foreign Exchange Market: an Empirical Estimates from Ukraine. Journal of Asian Multicultural Research for Economy and Management Study, 1(2), 1-6.

20. Mustafa, S. K., Ayaz, A. M., Baranova, V., Deineko, Z., Lyashenko, V., & Oyouni, A. A. A. (2020). Using wavelet analysis to assess the impact of COVID-19 on changes in the price of basic energy resources. International Journal of Emerging Trends in Engineering Research, 8(7), 2907-2912.

21. Lyashenko, V. V., Lyubchenko, V. A., Ahmad, M. A., Khan, A., & Kobylin, O. A. (2016). The Methodology of Image Processing in the Study of the Properties of Fiber as a Reinforcing Agent in Polymer Compositions. International Journal of Advanced Research in Computer Science, 7(1), 15-18.

22. Lyubchenko, V., & et al.. (2016). Digital image processing techniques for detection and diagnosis of fish diseases. International Journal of Advanced Research in Computer Science and Software Engineering, 6(7), 79-83.

23. Lyashenko, V. V., Matarneh, R., Kobylin, O., & Putyatin, Y. P. (2016). Contour Detection and Allocation for Cytological Images Using Wavelet Analysis Methodology. International Journal, 4(1), 85-94.

24. Kuzemin, O., & Lyashenko, V. Microsituation Concept in GMES Decision Support Systems/A. Kuzemin, V. Lyashenko. Intelligent Data Processing in Global Monitoring for Environment and Security (pp. 217–238).–2011.–P, 217-238.

25. Lyashenko, V., Laariedh, F., Ayaz, A. M., & Sotnik, S. (2021). Recognition of Voice Commands Based on Neural Network. TEM Journal: Technology, Education, Management, Informatics, 10(2), 583-591.

26. Orobinskyi, P., Deineko, Z., & Lyashenko, V. (2020). Comparative Characteristics of Filtration Methods in the Processing of Medical Images. American Journal of Engineering Research, 9(4), 20-25.

27. Sotnik S., & et al.. (2022). Key Directions for Development of Modern Expert Systems. International Journal of Engineering and Information Systems (IJEAIS), 6(5), 4-10.

28. Sotnik S., & Lyashenko V. (2022). Prospects for Introduction of Robotics in Service. International Journal of Academic Engineering Research (IJAER), 6(5), 4-9.

29. Lyashenko, V., & Sotnik, S. (2022). Overview of Innovative Walking Robots. International Journal of Academic Engineering Research (IJAER), 6(4), 3-7.

30. Baranova, V., Orlenko, O., Vitiuk, A., Yakimenko-Tereschenko, N., & Lyashenko, V. (2020). Information system for decision support in the field of tourism based on the use of spatio-temporal data analysis. International Journal of Advanced Trends in Computer Science and Engineering, 9(4), 6356-6361.

31. Sotnik, S., & et al.. (2022). Modern Industrial Robotics Industry. International Journal of Academic Engineering Research, 6(1),. 37-46.

32. Lyashenko, V., Kobylin, O., & Baranchykov, Y. (2018, October). Ideology of Image Processing in Infocommunication Systems. In 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T) (pp. 47-50). IEEE.

33. Lyashenko, V., & Sotnik, S. (2020). Analysis of Basic Principles for Sensor System Design Process Mobile Robots. Journal La Multiapp, 1(4), 1-6.

34. Lyashenko V, Matarneh R, Sotnik S. Modeling of Machine Design with Numerical Control in UG NX 7.5 System. The International Journal of Engineering and Science (IJES), 2018, 7(7), pp. 28-37.

35. Matarneh, R., Sotnik, S., & Lyashenko, V. (2018). Search of the molding form connector plane on the approximation basis by the many-sided surface with use of

the convex sets theory. International Journal of Mechanical and Production Engineering Research and Development, 8(1), 977-988.

36. Babker, A. M., Suliman, R. S., Elshaikh, R. H., Boboyorov, S., & Lyashenko, V. (2024). Sequence of Simple Digital Technologies for Detection of Platelets in Medical Images. Biomedical and Pharmacology Journal, 17(1), 141-152.

37. Yevstratov, M., Lyubchenko, V., Amer, A. J., & Lyashenko, V. (2024). Color correction of the input image as an element of improving the quality of its visualization. Technical science research in Uzbekistan, 2(4), 79-88.

38. Abu-Jassar, A., Al-Sharo, Y., Boboyorov, S., & Lyashenko, V. (2023, December). Contrast as a Method of Image Processing in Increasing Diagnostic Efficiency When Studying Liver Fatty Tissue Levels. In 2023 2nd International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI) (pp. 1-5). IEEE.

39. Gurin, D., & et al. (2024). Using the Kalman Filter to Represent Probabilistic Models for Determining the Location of a Person in Collaborative Robot Working Area. Multidisciplinary Journal of Science and Technology, 4(8), 66-75.

40. Yevsieiev, V., & et al. (2024). Object Recognition and Tracking Method in the Mobile Robot's Workspace in Real Time. Technical science research in Uzbekistan, 2(2), 115-124.

41. Gurin, D., & et al. (2024). Using Convolutional Neural Networks to Analyze and Detect Key Points of Objects in Image. Multidisciplinary Journal of Science and Technology, 4(9), 5-15.

42. Yevsieiev, V., & et al. (2024). The Canny Algorithm Implementation for Obtaining the Object Contour in a Mobile Robot's Workspace in Real Time. Journal of Universal Science Research, 2(3), 7–19.

43. Gurin, D., & et al. (2024). Effect of Frame Processing Frequency on Object Identification Using MobileNetV2 Neural Network for a Mobile Robot. Multidisciplinary Journal of Science and Technology, 4(8), 36-44.

44. Yevsieiev, V., & et al. (2024). Using Contouring Algorithms to Select Objects in the Robots' Workspace. Technical science research in Uzbekistan, 2(2), 32–42.

45. Kim, D., & et al. (2020). Proximity prediction of mobile objects to prevent contact-driven accidents in co-robotic construction. Journal of Computing in Civil Engineering, 34(4), 04020022.

46. Ortiz, J., & et al. (2022). isdf: Real-time neural signed distance fields for robot perception. arXiv preprint arXiv:2204.02296.

47. Mulyanto, A., & et al. (2020). 2d lidar and camera fusion for object detection and object distance measurement of ADAS using robotic operating system (ROS). JOIV Int. J. Informatics Vis, 4(4), 231-236.

48. Stavridis, S., & et al. (2021). Pick-and-place in dynamic environments with a mobile dual-arm robot equipped with distributed distance sensors. In 2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids) IEEE, 76-82.

49. Shenoi, A., & et al. (2020). Jrmot: A real-time 3d multi-object tracker and a new large-scale dataset. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 10335-10342.

50. Wang, H., & et al. (2022). A study on long-close distance coordination control strategy for litchi picking. Agronomy, 12(7), 1520.

51. Driess, D., & et al. (2022). Learning models as functionals of signed-distance fields for manipulation planning. In Conference on robot learning, PMLR, 245-255.

52. Yevsieiev, V., & et al. (2024). Building a traffic route taking into account obstacles based on the A-star algorithm using the python language. Technical Science Research In Uzbekistan, 2(3), 103-112.