JOURNAL OF UNIVERSAL SCIENCE RESEARCH

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

Capturing Human Movements in Real Time in Collaborative Robots Workspace within Industry 5.0

Vladyslav Yevsieiev [1], Svitlana Maksymova [1], Ahmad Alkhalaileh [2]

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

[2] Senior Developer Electronic Health Solution, Amman, Jordan

Abstract: The article examines the application of technologies for capturing human movements in real time in collaborative robots workspace in the context of Industry 5.0. Mathematical apparatus and software for analyzing movements with high accuracy has been developed, which allows to increase the safety and efficiency of human-robot interaction. The conducted experiments showed the influence of lighting conditions and movements speed on the accuracy of movements capture and visualization, which is critically important for the optimal operation of collaborative robots in production environments.

Key words: Real-time motion capture, collaborative robots, human-robot interaction, Industry 5.0, pose estimation, workplace safety

Introduction

Capturing human movements in real time in the collaborative robot workspace manipulators is of particular importance in the context of the development of Industry 5.0, where the main emphasis is placed on the harmonious interaction of humans and robots [1]-[15]. Industry 5.0 differs from the previous paradigm of Industry 4.0 in that it strives for a closer integration of human creativity and intelligence with automated systems, increasing the efficiency, flexibility and adaptability of production processes. Capturing human movements in the work area allows manipulator robots to respond to operator actions in real time, ensuring safe and efficient collaboration [16]-[20]. This is important for tasks such as joint assembly of complex components, performing delicate operations that require precision and human-machine interaction.

The study of this technology is relevant due to the need to increase the level of automation without losing human control, as well as with the aim of improving working

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

conditions, reducing operator fatigue and ensuring his safety. Different methods and approaches can be used here [21]-[36]. Ensuring safety is a key issue in developing collaborative robotsWithin Industry 5.0, it is important that such technologies contribute to the development of personalized production, where robots are able to adapt to the individual needs and work style of each operator, which makes research in this area extremely relevant.

Related works

With the advent of Industry 5.0, the problem of ensuring safety has become especially acute. Accordingly, human detection in the robot's workspace has come to the fore. And it is natural that more and more scientists are devoting their works to this problem. Let's consider several recent such works.

Let us begin with the work [37]. There is noted that industrial robots especially in human–robot collaboration settings can be hazardous if safety is not addressed properly. Human–robot collaboration systems can be a tremendously complex process; therefore, proper safety mechanisms must be addressed at an early stage of development.

Bonci, A., and co-authors in [38] write that if the robot is not aware of the human position and intention, a shared workspace between robots and humans may decrease productivity and lead to human safety issues. So, their study [38] presents a survey on sensory equipment useful for human detection and action recognition in industrial environments.

Rahmaniar, W., & Hernawan, A. in [39] propose to use SSD MobileNet V2 model that provides the highest accuracy with the fastest computation time compared to other models in our video datasets with several scenarios in order to detect an object such as a human.

The paper [40] researches hand gestures that are quite suitable for space human–robot interaction because of their natural and convenient features. But hand gestures are very complicated and hand sizes are very small in some images. These problems make the robust real-time hand detection and localization very difficult.

Scientists in [41] introduce a new feature to improve human classification in sparse, long-range point clouds. They present a system for online learning of human

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

classifiers by mobile service robots using 3D LiDAR sensors, and its experimental evaluation in a large indoor public space.

The article [42] notes that presence of workers in the shared workspace with robots decreases the productivity, as the robot is not aware about the human position and intention, which leads to concerns about human safety. This issue is addressed in this work by designing a reliable safety monitoring system for collaborative robots (cobots).

Researchers in [43] propose a multilayer feedforward neural network-based approach for human–robot collision detection taking safety standards into consideration.

Liu, C., & Szirányi, T. in [44] consider the problem of real-time UAV human detection and recognition of body and hand rescue gestures. They use body-featuring solutions to establish biometric communications, like yolo3-tiny for human detection.

Thus, we see that the problem of human detection and understanding of his gestures is extremely relevant. Further in our article we will present a study of the dependence of detection quality on lighting, as well as gesture recognition on the speed of their execution.

Capturing human movements in real time mathematical model.

The mathematical model of capturing human movements with the coordinates calculation in the frame in real time based on computer vision systems can be described as a sequence of stages.

The first stage, receiving a video stream from the camera installed on the gripping device of the robot. A video stream can be represented as the following discrete sequence of frames

$$V(t) = \{F_1, F_2, \ldots, F_n\},\ F_i = f(t_i) \qquad (1)$$

$V(t)$ – video stream;

F_i – a single frame received at a point in time t_i;

$f(t_i)$ – a function that describes a frame at time.

The next stage is frame conversion. Each frame F_i is first converted to a coordinate system suitable for further processing (for example, RGB or grayscale):

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

$$F_i^{RGB}=T(F_i) \quad (2)$$

$T(F_i)$ – the frame conversion operator to the desired color space, which represents each pixel of the frame as a vector $P(x,y)$ in the color space.

After that, it is necessary to determine the key points (Landmarks Detection). The model of the human body can be described through a set of key points (landmarks), which represent joints or other characteristic points. Let the computer vision system determine the N key points, for each of which the coordinates in the frame space are calculated:

$$L_i=\{(x_i,y_i)\},\ i=1,2,\ldots,N \quad (3)$$

L_i – point i coordinates;

(x_i,y_i) – coordinates of this point on the frame plane (in pixels).

To generalize the position of the body in relative coordinates, the position of key points is normalized:

$$x_i'=x_i/W,\ y_i'=y_i/H \quad (4)$$

W and H – width and height of the frame, respectively;

x_i' and y_i' – normalized point coordinates.

Distances between key points, which can be calculated using the Euclidean metric, are important for motion analysis:

$$d_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \quad (5)$$

d_{ij} – distance between points i and j.

In order to smooth the movement, which can be "noisy" due to small deviations or errors in the frames, filtering is used. For example, smoothing using an exponential moving average:

$$x_i''=ax_i(t)+(1-a)x_i(t-1) \quad (6)$$

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

a – коефіцієнт згладжування;

x_i'' – the smoothed value of the coordinate at the moment of time t;

$x_i(t)$ – coordinate at a moment in time t.

Human movement can also be described through the rate of change in the position of key points in time.

Point movement speed:

$$v_i(t)= (x_i(t)-x_i(t-1))/\Delta t \quad (7)$$

Δt – time interval between frames.

Acceleration is calculated as a change in velocity:

$$a_i(t)= (v_i(t)-v_i(t-1))/\Delta t \quad (8)$$

After all the calculations, the coordinates of each point in the frame space can be represented as:

$$C=\{(x_1',y_1'), (x_2',y_2'),..., (x_N',y_N')\} \quad (9)$$

C – a set of coordinates of all key points.

The presented model covers the process of capturing human movements in real time, starting with the processing of the video stream and ending with the body key points coordinates calculation. This model can be used to control manipulator robots, responding to human movements and ensuring effective cooperation in the workspace.

Development of a program for testing the human motion capture model in real time and conducting experiments

The choice of Python as the language for developing a real-time human motion capture model testing program is due to its versatility and ease of working with computer vision libraries such as OpenCV and MediaPipe [45]. Python supports easy integration with machine learning algorithms, allowing rapid prototyping and testing of

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

models. High code readability and a large amount of documentation simplify the process of developing and debugging programs. As an integrated development environment (IDE), PyCharm provides convenient tools for writing code, debugging, and working with libraries, making it an ideal choice for Python projects. PyCharm supports virtual environments, making it easier to manage dependencies and simplify project setup to work with real-time models. The combination of Python and PyCharm makes it possible to quickly develop, test and scale high-accuracy human motion capture applications.

We will give a description of some fragments of the software code from implementation of capturing human movements mathematical model in real time.

```
frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
results = pose.process(frame_rgb)
frame_pose = frame.copy()
```

This piece of code is necessary to correctly capture human poses from a video stream using MediaPipe. First, the frame read by OpenCV in BGR format is converted to RGB, since MediaPipe works with this format. The frame is then processed for pose analysis, where MediaPipe identifies key points on the human body. After that, a copy of the frame is created, on which the processing results will be superimposed, for example, a human skeleton, without changing the original image.if results.pose_landmarks:

```
    # Output of coordinates of all points (e.g. point #0 — nose)
    for idx, landmark in enumerate(results.pose_landmarks.landmark):
      h, w, _ = frame.shape
      cx, cy = int(landmark.x * w), int(landmark.y * h)
```

This piece of code is used to check if MediaPipe was able to find body landmarks (pose_landmarks) in the frame, and if so, it outputs the coordinates of all those points. It loops through each skeleton point, getting its normalized (x,y) coordinates and converting them to pixel coordinates based on the frame dimensions (height and width). This allows you to determine the exact location of each key point in the image, for example, the coordinates of the nose or other body parts.

```
    # Output the coordinates in the frame
    cv2.putText(frame_pose, f'ID {idx}: ({cx}, {cy})', (cx, cy),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 0, 0), 2)
```

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

This piece of code is used to output text information on a video frame that shows the index and coordinates of each key point on the body (for example, the coordinates of the nose or other body parts). The text with the index of the point and its coordinates is applied directly to the image in the corresponding position using the font and color (in this case blue). This helps visualize the location and ID of each point on the human body in the frame.

Drawing a skeleton on the frame

mp_drawing.draw_landmarks(frame_pose, results.pose_landmarks, mp_pose.POSE_CONNECTIONS)

This piece of code is used to render a human skeleton on a frame by drawing key body points and their connections using the MediaPipe library. Using the processing results (pose_landmarks), the draw_landmarks function draws lines and points representing the joints and connections between them on the image. This allows you to graphically display a person's pose in real time, showing their skeletal structure directly on the video frame.

Output of two windows: original video and with poses

cv2.imshow('Real-time Video', frame)

cv2.imshow('Pose Estimation', frame_pose)

This code fragment is responsible for the simultaneous display of two windows: one with the original video stream and the other with a video on which visualization of the human skeletal structure is superimposed. This allows the user to see both the raw video and real-time human pose estimation results. In this way, the user can compare the original video material with the video, on which the key points and the connections between them are marked.

The results of the developed program for capturing human movements in real time in the Python language in the PyCharm development environment are shown in Figure 1.

Let us conduct a series of experiments to evaluate the parameters of the developed program for the implementation of the human motion capture model in real time:

- lighting effects research [46], i.e. experiments in different lighting conditions to evaluate how it affects the accuracy of motion capture and visualization of key points;

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

- analysis of speed and accuracy, study of the delay between human movements and their reflection in the system, and also evaluate the accuracy of detecting coordinates for fast or complex movements.

a) b)

a) Real-time Video Window; b) Pose Estimation Window

Figure 1: Developed program for capturing human movements in real time results

Table 1 shows the results of experiments on the study of the influence of lighting on the capturing human movements in real time accuracy.

Table 1: Results of experiments on the study of the lighting influence on the capturing human movements in real time accuracy of

Lighting conditions	Brightness (lux)	Detection accuracy (%)	Movements visualization	Notes
Bright light	800	95	Clear, no artifacts	Optimal conditions
Moderate light	500	87	Good, possible artifacts	Some points may be omitted

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

Weak light	200	73	Blurry, unclear points	Significant loss of accuracy
Low light	100	61	Very blurry	High probability of errors
Strobe light	-	51	Unpredictable	Deterioration of visualization
Multicolored light	-	62	Inconstant	Effect of colors on accuracy

The obtained results of the experiment on the study of the effect of lighting on the accuracy of capturing human movements in real time are presented in Figure 2.

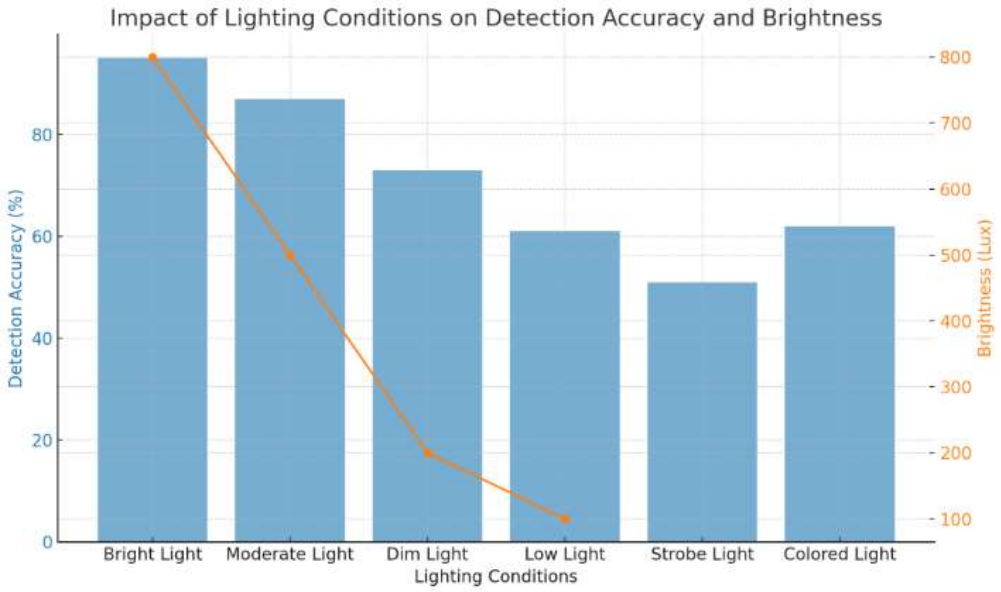

Figure 2: Combined graph of the effect of illumination on the accuracy of human motion capture in real time

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

The obtained results of the study of the lighting influence on the movement capture accuracy demonstrate a clear relationship between the intensity of light and the quality of detecting the positions of key points. In bright conditions (800 lux), the system shows high accuracy (95%) with clear visualization, which confirms optimal conditions for work. In moderate light (500 lux), the accuracy drops to 87% and artifacts appear, indicating that some points are lost. In weak and low light (200-100 lux), the accuracy drops sharply to 73% and 61%, respectively, which is accompanied by blurring and an increased probability of errors. Stroboscopic and multi-colored lighting also significantly degrade accuracy (51% and 62%) due to rendering instability and the effect of colors on the algorithm.

Table 2 shows the results of experiments on the speed and accuracy of the real-time motion capture program, with an emphasis on the delay between human movements and their reflection in the system, as well as on the accuracy of detecting coordinates in complex or fast movements.

The obtained results of the experiment on the study of the speed and accuracy of the real-time motion capture program, with an emphasis on the delay between human movements and their reflection in the system, as well as on the accuracy of detecting coordinates in complex or fast movements, are presented in Figure 3.

The results of the study of the speed and accuracy of the real-time motion capture program show a clear correlation between the complexity and speed of movements and the delay of their display in the system.

Table 2: Results of experiments on the speed and accuracy of the real-time motion capture program.

Movement type	Average latency (ms)	Accuracy of coordinate detection (%)	Notes on movement complexity
Slow, simple movements	47	98	-
Moderate, medium difficulty movements	58	93	Slight delay, accuracy decreases slightly

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

Fast, medium difficulty movements	94	83	Noticeable delay, possible inaccuracies
Very fast movements	128	69	Significant delay, accuracy noticeably deteriorates
Complex, unpredictable movements	164	53	High latency, low accuracy

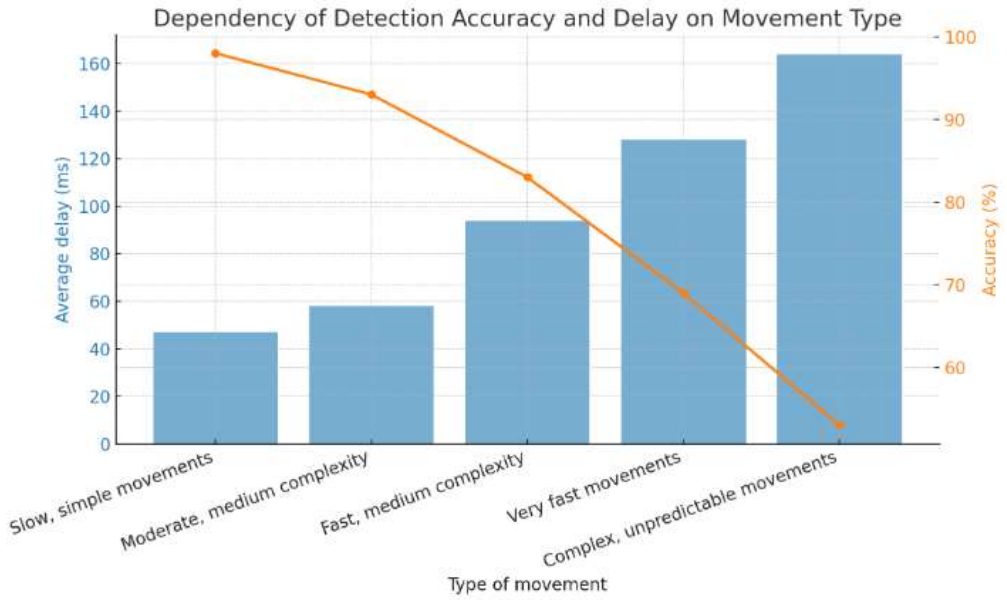

Figure 3: Combined graph of the study of the speed and accuracy of the real-time motion capture program

For slow and simple movements, the average delay is 47 ms with a high accuracy of coordinate detection (98%), which indicates an almost instantaneous response of the system. With moderate movements, the delay increases to 58ms, and the accuracy drops to 93%, indicating minor errors. Fast movements cause a noticeable increase in latency to 94 ms and a drop in accuracy of up to 83%. For very fast and complex movements, the delay reaches 128-164 ms, while the accuracy drops to 69% and 53%, respectively, which indicates significant errors in coordinate processing under such conditions.

Conclusion

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

As part of the study of models, methods and software for capturing human movements in real time in the collaborative robot workspace within Industry 5.0, the importance and effectiveness of using computer vision systems for detecting and analyzing human movements is considered. Modern technologies allow integrating solutions on based on computer vision to improve human-robot collaboration. In the context of Industry 5.0, special attention is paid to the safety and accuracy of the interaction, which is made possible by the real-time analysis of the movements of the workers human with high accuracy thanks to MediaPipe and Python in the PyCharm environment.

Based on experiments, it was determined that lighting and movement speed significantly affect the quality of capture. In bright conditions, accuracy reached 95%, while in low light, accuracy dropped to 61%, and under strobe light, accuracy deteriorated further. Experiments with different types of movements have also shown that slow and simple movements provide maximum accuracy and minimum latency. However, with very fast and complex movements, the delay reached 164 ms, which negatively affects the accuracy of coordinate detection.

The developed software demonstrated its effectiveness in capturing human movements, but showed some limitations in difficult environments. The accuracy of the system depends on the quality of the lighting and the complexity of the movements, which can be critical when working with collaborative robots. In general, the integration of such technology into Industry 5.0 helps to increase the safety, accuracy and efficiency of production processes, which is confirmed by the conducted experiments and the analysis of the obtained results.

References:

1. Samoilenko, H., & et al. (2024). Review for Collective Problem-Solving by a Group of Robots. Journal of Universal Science Research, 2(6), 7-16.
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.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

4. 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.

5. 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.

6. 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.

7. 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.

8. Matarneh, R., Maksymova, S., Deineko, Z., & Lyashenko, V. (2017). Building robot voice control training methodology using artificial neural net. International Journal of Civil Engineering and Technology, 8(10), 523-532.

9. 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.

10. 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.

11. 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.

12. 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.

13. 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.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

14. 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.

15. 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.

16. 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.

17. 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.

18. 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.

19. 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.

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

21. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2021). Neural networks as a tool for pattern recognition of fasteners. International Journal of Engineering Trends and Technology, 69(10), 151-160.

22. 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.

23. Deineko, Zh., & et al.. (2021). Color space image as a factor in the choice of its processing technology. Abstracts of I International scientific-practical conference «Problems of modern science and practice» (September 21-24, 2021). Boston, USA, pp. 389-394.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

24. Lyashenko, V., Kobylin, O., & Ahmad, M. A. (2014). General methodology for implementation of image normalization procedure using its wavelet transform. International Journal of Science and Research (IJSR), 3(11), 2870-2877.

25. Lyashenko, V., Matarneh, R., & Kobylin, O. (2016). Contrast modification as a tool to study the structure of blood components. Journal of Environmental Science, Computer Science and Engineering & Technology, 5(3), 150-160.

26. 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.

27. 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.

28. Mousavi, S. M. H., Lyashenko, V., & Prasath, S. (2019). Analysis of a robust edge detection system in different color spaces using color and depth images. Компьютерная оптика, 43(4), 632-646.

29. 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.

30. Orobinskyi, P., Petrenko, D., & Lyashenko, V. (2019, February). Novel approach to computer-aided detection of lung nodules of difficult location with use of multifactorial models and deep neural networks. In 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM) (pp. 1-5). IEEE.

31. Lyashenko, V., Kobylin, O., & Selevko, O. (2020). Wavelet analysis and contrast modification in the study of cell structures images. International Journal of Advanced Trends in Computer Science and Engineering, 9(4), 4701-4706.

32. Matarneh, R., & et al.. (2019). Development of an Information Model for Industrial Robots Actuators. IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE), 16(1), 61-67.

33. Sotnik, S., & et al.. (2022). Analysis of Existing Infliences in Formation of Mobile Robots Trajectory. International Journal of Academic Information Systems Research, 6(1), 13-20.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-10

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

35. Drugarin, C. V. A., Lyashenko, V. V., Mbunwe, M. J., & Ahmad, M. A. (2018). Pre-processing of Images as a Source of Additional Information for Image of the Natural Polymer Composites. Analele Universitatii'Eftimie Murgu', 25(2).

36. Lyubchenko, V., Veretelnyk, K., Kots, P., & Lyashenko, V. (2024). Digital image segmentation procedure as an example of an NP-problem. Multidisciplinary Journal of Science and Technology, 4(4), 170-177.

37. Arents, J., & et al. (2021). Human–robot collaboration trends and safety aspects: A systematic review. Journal of Sensor and Actuator Networks, 10(3), 48.

38. Bonci, A., & et al. (2021). Human-robot perception in industrial environments: A survey. Sensors, 21(5), 1571.

39. Rahmaniar, W., & Hernawan, A. (2021). Real-time human detection using deep learning on embedded platforms: A review. Journal of Robotics and Control (JRC), 2(6), 462-468.

40. Gao, Q., & et al. (2020). Robust real-time hand detection and localization for space human–robot interaction based on deep learning. Neurocomputing, 390, 198-206.

41. Yan, Z., & et al. (2020). Online learning for 3D LiDAR-based human detection: experimental analysis of point cloud clustering and classification methods. Autonomous Robots, 44(2), 147-164.

42. Mohammadi Amin, F., & et al. (2020). A mixed-perception approach for safe human–robot collaboration in industrial automation. Sensors, 20(21), 6347.

43. Sharkawy, A. N., & et al. (2020). Human–robot collisions detection for safe human–robot interaction using one multi-input–output neural network. Soft Computing, 24(9), 6687-6719.

44. Liu, C., & Szirányi, T. (2021). Real-time human detection and gesture recognition for on-board UAV rescue. Sensors, 21(6), 2180.

45. 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.

46. Vizir, Y., & et al. (2024). Lighting Control Module Software Development. Journal of Universal Science Research, 2(2), 29–42.