ISSN: **2992-9148** **2024, 2**(2), 115–124

Object Recognition and Tracking Method in the Mobile Robot's Workspace in Real Time

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]Faculty of Information Technology, Department of Computer Science, Ajloun National University, Ajloun, Jordan

Abstract:

This article presents an object recognition and tracking method in the mobile robot's workspace in real time. The main approach is to use a color mask, for which a mathematical description of the algorithm is proposed. The method is implemented in the Python programming language using the PyCharm development environment. During the research, experiments were carried out, on the basis of which important performance indicators were obtained. The processing time indicator, which measures the processing time of each frame of a video stream, demonstrated high efficiency, ranging from 0.0010 to 0.0020 seconds. Detection speed, which determines the speed of object detection in FPS, also presented good results, ranging from 501.47 to 1037.42 FPS.

Key words: Industry 5.0, Computer Vision Systems, Mobile Robots, Work zone

Introduction

In this era of Industry 5.0, where industrial processes are becoming increasingly integrated and automated, the role of mobile robots in increasing the efficiency and flexibility of production lines is becoming critical [1]-[9]. The implementation of methods for recognizing and tracking objects in the mobile robot's workspace in real time becomes a key component of the control and monitoring system [10]-[17]. Using modern computer vision technologies such as Computer Vision Systems, these mobile robots acquire the ability to perceive the environment, perform object recognition and make decisions based on the information received [18]-[20]. At the same time, here you can use various methods and approaches that help to understand this research problem [21]-[31].

Of particular relevance in Industry 5.0 concepts is the need to ensure the safety and efficiency of people in the work areas of mobile robots. Real time is becoming a critical factor in ensuring mobile robots can quickly respond to changing conditions in the production environment.

The implementation of methods for recognizing and identifying objects in the robot's work area not only improves production processes, but also helps reduce risks and increase the accuracy of tasks performed by mobile robots in real time. Therefore, research in the field of object recognition and tracking in the work areas of mobile robots is of great importance for the further development of robotics, strengthening the position of Industry 5.0 and improving the overall productivity in various industries.

ISSN: **2992-9148 2024, 2**(2), 115–124

Related works

There are plenty of researches devoted to solving the problem of object recognition and tracking. Let us consider only small part of such recent works.

There is noted that three-dimensional (3D) object recognition is widely used in automated driving, medical image analysis, virtual/augmented reality, artificial intelligence robots, and other areas [32]. In [32] there is presented a comprehensive review and classification of the latest developments in the deep learning methods for multi-view 3D object recognition.

It is also noted in [33] that with the rapid development of deep learning, many deep learning-based approaches have made great achievements in object detection tasks. Authors insist that deep learning is a data-driven approach. Data directly impact the performance of object detectors to some extent.

Researchers assess the tendency of state-of-the-art object recognition models to depend on signals from image backgrounds in [34]. Theirr analysis of backgrounds brings us closer to understanding which correlations machine learning models use, and how they determine models' out of distribution performance.

Scientists [35] consider object recognition as a key research area in the field of image processing and computer vision, which recognizes the object in an image and provides a proper label. There are compared three popular feature descriptor algorithms: Scale Invariant Feature Transform, Speeded Up Robust Feature and Oriented Fast and Rotated BRIEF for experimental work of an object recognition system. They are determined individually and with different combinations of these three methodologies.

The paper [36] proposes the indoor scene semantic segmentation model. It not only has good performance and high efficiency, but also can segment the contours of different scale objects clearly and adapt to the indoor uneven lighting environment.

Luo, W. and co-authors [37] contribute the first comprehensive and most recent review on multiple object tracking. They inspect the recent advances in various aspects and propose some interesting directions for future research.

Article [38] proposes an multiple object tracking system that allows target detection and appearance embedding to be learned in a shared model.

Researchers [39] note that an important area of computer vision is real-time object tracking, which is now widely used in intelligent transportation and smart industry technologies. This article focuses on the correlation filter-based object tracking algorithms.

Thus the field of object recognition and tracking is very popular among scientists and extremely relevant. Further in our article we propose our object recognition and tracking method in the mobile robot’s workspace in real time.

Object recognition and tracking method in real time

The most common method of describing an image consists of three separate color channels representing the colors red (R), green (G), and blue (B). WIC provides support for these three channels in the order of red green blue (RGB) or blue green red (BGR). Let us have an input image from a streaming video from a mobile robot's camera in the BGR color format. We denote by $B(x,y)$, $G(x,y)$ and $R(x,y)$ the pixel values in the images. To implement object recognition and tracking in real time on streaming video, in the first step we will convert to the HSV (Hue, Saturation, Value) color space. The mathematical function to convert from BGR to HSV can be written as follows:

$$(H(x,y),S(x,y),V(x,y)) = f((B(x,y),G(x,y),R(x,y),color\_BGR2HSV), \quad (1)$$

$B(x,y)$, $G(x,y)$, $R(x,y)$ – image pixel values in format BGR;

$H(x,y),S(x,y),V(x,y)$ – image pixel values in format HSV;

f - the function converts BGR image values to HSV, in this study the cv2.cvtColor function for the cv2 library of the Python language will be used.

After this, we will introduce a description of the threshold values of the ranges of hue, saturation and brightness to select the required object in the frame.

$$\begin{cases} H_{low} \le x_i \le H_{high} \\ S_{low} \le x_k \le H_{high} \\ V_{low} \le x_q \le V_{high} \end{cases}, \quad (2)$$

where: x_i, x_k, x_q – the corresponding values of hue, saturation and brightness for the HSV (Hue, Saturation, Value) color space, provided that the values $H_{low}, S_{low}, V_{low} \ge 0$ и $H_{high}, S_{high}, V_{high} \le 255$.

We will perform binarization on the resulting image in the HSV color space, with the condition that the binarization is performed using threshold values for each HSV channel, as described in expression 2. and create an image mask in accordance with the following expression:

$$M(x,y) = \begin{cases} 1, \text{if } H_{low} \le x_i \le H_{high}, S_{low} \le x_k \le S_{high}, V_{low} \le x_q \le V_{high} \\ 0 \end{cases}, \quad (3)$$

where: $M(x,y)$ - mask with coordinates (x,y);

ISSN: **2992-9148 2024, 2**(2), 115–124

x_i, x_k, x_q - the corresponding values of hue, saturation and brightness for the HSV (Hue, Saturation, Value) color space, provided that the values $H_{low}, S_{low}, V_{low} \geq 0$ и $H_{high}, S_{high}, V_{high} \leq 255$.

Based on the resulting mask, it is necessary to determine the sequence in which the contours are located. Let M be a binary mask representing the image after binarization. The pixels in this mask take on the values 0 or 1, where 1 represents the object and 0 represents the background. First, let's define the operation of selecting the contour of an object. Let C be the outline of the object on the mask M, then the selection operation can be represented in the following form:

$$C = \{(x,y) | M(x,y) = 1\}, \quad (4)$$

where: (x, y) – pixel coordinates on the mask,

$M(x, y)$ – pixel value.

The next step, using the selected contours, is to obtain a list of contours, where each contour is represented by a set of points.

$$C_{list} = \{C_1, C_2, ..., C_m\}, \quad (5)$$

where: C_{list} – a list of contours, where each contour is represented by a set of points;

C_m – m-th contour represented as a set of points.

Thus, the sequence of mathematical definitions for finding contours using a mask is as follows:

$$M \rightarrow C \rightarrow C_{list} \rightarrow \{C_1, C_2, ..., C_m\} \rightarrow n_j \quad (6)$$

where: M – binary mask;

C – outline of the object on the mask;

C_{list} – list of contours;

C_m – m-th contour from the list of contours;

n_j – number of points in j-th contour.

ISSN: **2992-9148 2024, 2**(2), 115–124

Software implementation

To check the correctness of the reasoning, we will develop a program in Python in the PyCharm 2022.2.3 (Professional Edition) development environment. Let us give an example of software implementation of the above described mathematical expressions.

```
lower_yellow = np.array([20, 100, 100]).
upper_yellow = np.array([30, 255, 255]).
```

The following code snippet sets thresholds in the HSV color space to highlight yellow in an image. lower_yellow specifies the lower limit of the hue, saturation, and value range, and upper_yellow specifies the upper limit. These values are used in the image binarization process, creating a mask that highlights only yellow objects. Thus, this piece of code is a key part of the color processing algorithm, aimed at highlighting and tracking objects of a given color in real time.

```
mask = cv2.inRange(hsv, lower_yellow, upper_yellow).
```

The above code snippet is used to create a binary mask that selects pixels in an image that fall within a given yellow range in the HSV color space. The cv2.inRange function compares the values of each pixel in an image with certain threshold values, setting those that fall within a given range to 1 (white) and those that do not to 0 (black). Thus, this operation creates a mask that selects only yellow objects for further analysis and processing.

```
len(contours) > 0:
    largest_contour = max(contours, key=cv2.contourArea)
    M = cv2.moments(largest_contour)
    if M["m00"] != 0:
        cx = int(M["m10"] / M["m00"])
        cy = int(M["m01"] / M["m00"])
        cv2.drawMarker(frame, (cx, cy), (0, 0, 255), cv2.MARKER_CROSS,
markerSize=40, thickness=3)
```

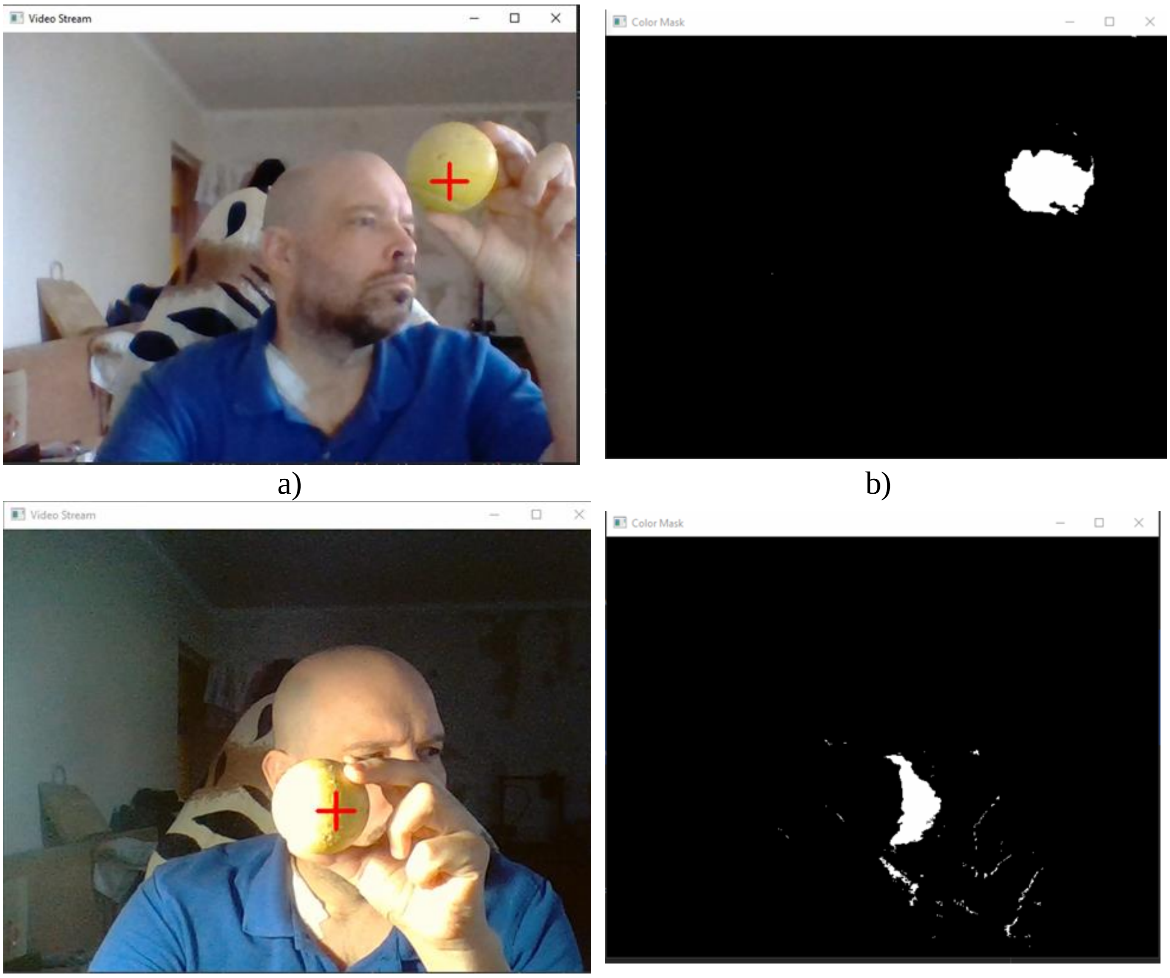
This code snippet performs object tracking based on the contours found in the previous step. If there are edges in the image, the code determines the largest edge that represents the largest object in the frame. Then, using the center of mass of this contour (determined using moments), the coordinates of the center of the object are calculated and a cross is drawn at this point on the video frame, indicating the center of the tracked object.

Experimental studies and analysis of the results obtained

The following hardware was used for research: CPU Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz, RAM 16 Gb, GPU NVideo GeForce GTX 1660Ti (Ram

ISSN: **2992-9148 2024, 2**(2), 115–124

8Gb), Web-camera HD WebCam, OS Windows 10 Pro (Version 22H2). The program for object recognition and tracking in real time was developed in the PyCharm 2022.2.3 (Professional Edition) environment in Python. The tracking object is a circular object with the following color space codes: HSB(48,39,82), RGB(208,192,126), CMYK(18,21,56,3), in Web#d0c07e. The obtained results of object recognition and tracking in real time in the mobile robot's workspace are presented in Figure 1.



a) b)

c) d)

a), c) – streaming video frame with object; b),d) – object mask

Figure 1: Object recognition and tracking results in real-time in the mobile robot's workspace

So, we have conducted several experiments.

During experiments on object recognition and tracking in real time, the following indicators were obtained:

– processing time this indicator measures the time spent processing each frame from the video stream. It allows to evaluate how quickly the algorithm can analyze and process input data.

During the experiment, it fluctuated within the range from 0.0010 to 0.0020 seconds.

– detection speed this parameter measures the number of detected objects per second (FPS - frames per second). High detection speed is important for real-time system responsiveness, especially when dealing with fast-moving objects or dynamic scenes.

During the experiment, it fluctuated between 501.47 and 1037.42 FPS. Basically, 501.47 FPS was obtained with a sharp movement of the object within the frame; with slow movement, the FPS tended to be 1037.42.

Conclusion

As a result of the research carried out on the implementation of a object recognition and tracking method in the mobile robot's workspace in real time using a color mask, significant conclusions were obtained. A mathematical description of the operation of the algorithm, based on obtaining a color mask, allows to effectively select and track objects in a video stream.

Important performance indicators are processing time and detection speed. The experimental results showed that the frame processing time varied from 0.0010 to 0.0020 seconds, which indicates the high efficiency of the algorithm. Object detection speed also showed impressive results, ranging from 501.47 to 1037.42 FPS. It is especially worth noting the high sensitivity of the algorithm when moving objects slowly, where the maximum detection speed was achieved.

Thus, the proposed implementation not only effectively implements object recognition and tracking, but also demonstrates high real-time performance, making it promising for application in mobile robots and other automation scenarios.

References:

1. Borysov, H., & et al. (2023). Parameters for Mobile Robot Kinematic Model Development Determination. Multidisciplinary Journal of Science and Technology, 3(4), 85-91.
2. Basiuk, V., & et al. (2023). Mobile Robot Position Determining Using Odometry Method. Multidisciplinary Journal of Science and Technology, 3(3), 227-234.
3. Nevliudov, I., & et al. (2023). Mobile Robot Navigation System Based on Ultrasonic Sensors. In 2023 IEEE XXVIII International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED), IEEE, 1, 247-251.
4. Al-Sharo Y., & et al. (2023). A Robo-hand prototype design gripping device within the framework of sustainable development. Indian Journal of Engineering, 2023, 20, e37ije1673.
5. Nevliudov, I., & et al. (2023). A Small-Sized Robot Prototype Development Using 3D Printing. In XXXI International Conference CAD In Machinery Design Implementation and Educational Issues,12

6. Yevsieiev, V., & et al. (2023). A Small-Scale Manipulation Robot a Laboratory Layout Development. International independent scientific journal, 47, 18-28.
7. Yevsieiev, V., & et al. (2022). A Robotic Prosthetic a Control System and a Structural Diagram Development. Collection of scientific papers «ΛΌГОΣ», Zurich, 113-114.
8. 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.
9. Lyashenko, V., & et al. (2023). Automated Monitoring and Visualization System in Production. Int. Res. J. Multidiscip. Technovation, 5(6), 09-18.
10. 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.
11. Rabotiahov, A., Kobylin, O., Dudar, Z., & Lyashenko, V. (2018, February). Bionic image segmentation of cytology samples method. In 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET) (pp. 665-670). IEEE.
12. 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.
13. Lyashenko, V., Deineko, Z., & Ahmad, A. (2015). Properties of Wavelet Coefficients of Self-Similar Time Series. International Journal of Scientific and Engineering Research, 6, 1492-1499.
14. Гиренко, А. В., Ляшенко, В. В., Машталир, В. П., & Путятин, Е. П. (1996). Методы корреляционного обнаружения объектов. Харьков: АО “БизнесИнформ, 112.
15. 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.
16. Babker, A. M., Altoum, A. E. A., Tvoroshenko, I., & Lyashenko, V. (2019). Information technologies of the processing of the spaces of the states of a complex biophysical object in the intellectual medical system health. International Journal of Advanced Trends in Computer Science and Engineering, 8(6), 3221-3227.
17. Matarneh, R., Tvoroshenko, I., & Lyashenko, V. (2019). Improving Fuzzy Network Models For the Analysis of Dynamic Interacting Processes in the State Space. International Journal of Recent Technology and Engineering, 8(4), 1687-1693.
18. Maksymova, S., & Velet, A. (2022). Development of an Automated System of Terminal Access to Production Equipment Using Computer Vision//

Manufacturing & Mechatronic Systems 2022: Proceedings of VIst International Conference, Kharkiv, 22-23.
19. Shcherbyna, V., & et al. (2023). Mobile Robot for Fires Detection Development. Journal of Universal Science Research, 1(11), 17-27.
20. Nikitin, V., & et al. (2023). Traffic Signs Recognition System Development. Multidisciplinary Journal of Science and Technology, 3(3), 235-242.
21. Lyashenko, V. V., Babker, A. M. A. A., & Kobylin, O. A. (2016). The methodology of wavelet analysis as a tool for cytology preparations image processing. Cukurova Medical Journal, 41(3), 453-463.
22. Lyashenko, V., & et al.. (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.
23. Dadkhah, M., Lyashenko, V. V., Deineko, Z. V., Shamshirband, S., & Jazi, M. D. (2019). Methodology of wavelet analysis in research of dynamics of phishing attacks. International Journal of Advanced Intelligence Paradigms, 12(3-4), 220-238.
24. 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.
25. 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.
26. 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.
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. Lyashenko, V., Zeleniy, O., Mustafa, S. K., & Ahmad, M. A. (2019). An advanced methodology for visualization of changes in the properties of a dye. International Journal of Engineering and Advanced Technology, 9(1), 711-7114.
29. Mousavi, S. M. H., Victorovich, L. V., Ilanloo, A., & Mirinezhad, S. Y. (2022, November). Fatty Liver Level Recognition Using Particle Swarm optimization (PSO) Image Segmentation and Analysis. In 2022 12th International Conference on Computer and Knowledge Engineering (ICCKE) (pp. 237-245). IEEE.
30. Boboyorov Sardor Uchqun o‘g‘li, Lyubchenko Valentin, & Lyashenko Vyacheslav. (2023). Image Processing Techniques as a Tool for the Analysis of Liver Diseases. Journal of Universal Science Research, 1(8), 223–233.

31. 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.
32. Qi, S., & et al. (2021). Review of multi-view 3D object recognition methods based on deep learning. Displays, 69, 102053.
33. Sun, X., & et al. (2022). FAIR1M: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. ISPRS Journal of Photogrammetry and Remote Sensing, 184, 116-130.
34. Xiao, K., & et al. (2020). Noise or signal: The role of image backgrounds in object recognition. arXiv preprint arXiv:2006.09994.
35. Bansal, M., & et al. (2021). 2D object recognition: a comparative analysis of SIFT, SURF and ORB feature descriptors. Multimedia Tools and Applications, 80, 18839-18857.
36. Jiang, D., & et al. (2021). Semantic segmentation for multiscale target based on object recognition using the improved Faster-RCNN model. Future Generation Computer Systems, 123, 94-104.
37. Luo, W., & et al. (2021). Multiple object tracking: A literature review. Artificial intelligence, 293, 103448.
38. Wang, Z., & et al. (2020). Towards real-time multi-object tracking. In European Conference on Computer Vision, Cham: Springer International Publishing, 107-122.
39. Liu, S., & et al. (2021). Overview and methods of correlation filter algorithms in object tracking. Complex & Intelligent Systems, 7, 1895-1917.