ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

THE SOBEL ALGORITHM IMPLEMENTATION FOR DETECTION AN OBJECT CONTOUR IN THE MOBILE ROBOT'S WORKSPACE IN REAL TIME

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:

This article is devoted to the Sobel algorithm implementation for detection an object contour in the mobile robot's workspace in real time. Mathematical models of the algorithm functioning were examined in detail, and the developed program in Python in the PyCharm environment was subjected to a series of experiments. The experimental results indicate the outstanding performance of the algorithm, reaching 30.00 frames per second when processing a video stream. However, the contouring speed of 1.19 frames per second indicates potential performance challenges when working in dark conditions using artificial side lighting. The research results presented in the article highlight the effectiveness of the algorithm in real-time conditions and highlight the importance of taking into account lighting features when using it in mobile robots.

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

Introduction

In the context of the rapid development of technological progress and the transition to Industry 5.0, the implementation of innovative solutions in the field of mobile robotics is becoming extremely relevant. In this context, the field of Computer Vision Systems becomes particularly important [1]-[13], where modern image processing technologies and computer vision algorithms play a key role in increasing the efficiency and autonomy of mobile robots [14]-[28]. Here, other approaches can also be used to find new solutions [29]-[32].

One of the fundamental aspects in the development of mobile robots is the ability to accurately perceive the environment and make informed decisions based on the information received. In this context, implementing the Sobel algorithm to extract the objects contours in the mobile robot's workspace in real time becomes an important step in achieving accurate navigation and obstacle detection [33],[34].

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

The mobile robots' workspace is an environment full of dynamic changes, requiring continuous monitoring and adaptive response. The real-time implementation of the Sobel algorithm allows mobile robots to effectively detect the contours of objects, which is a fundamental element for effective navigation and collision avoidance.

Such research is justified by the need to develop technologically advanced mobile robotics systems capable of functioning in difficult work environments. At the same time, the use of the Sobel algorithm in Industry 5.0 becomes an important tool for ensuring high accuracy and responsiveness of mobile robots in real time

Related works

Among many studies devoted to the extraction of object contours, many researchers prefer to use the Sobel algorithm. Let us look at just a small part of them.

Lynn, N. D., and co-authors in [33] presented results of comparison Canny and Sobel algorithms. After using both algorithms to detect edges of real-time images, the result showed that the Canny algorithm produced thick edges compared to the Sobel algorithm. More so, Canny uses a double threshold for edge revelation and applies the Gaussian filter which removes of any noise from an image, unlike Sobel which is not resistant to noise.

Authors in [34] note that Sobel edge detection is widely used in computer vision and image processing but its processing time becomes a serious problem in real-time environments, especially when an image is very large. They propose their own algorithm that reduces the number of arithmetic operations and data loads, so that processing speed is increased and energy consumption reduced.

There is proposed an improved edge detection algorithm based on Sobel operator in [35]. There is noted that raditional Sobel edge detection operator has several disadvantages, such as low accuracy of image edge location and rough edge extracted.

In Sobel edge detection algorithm hardware implementation is chosen because it can work with less decline in noise at high level. But where the edge detected image cannot obtained completely and the time and space complexity are also high [36]. In order to rectify these drawbacks, the proposed system has the addition of extra masks and also changes the coefficients of each masks in image filter.

AS, R. A., & Gopalan, S. in their work [37] Sobel algorithm with 8-directional template is implemented for improving the detection of edges in brain tumor MRI images.

Reserchers in [38] write that one popular algorithm for edge detection is the Sobel. Many researchers have focused on accelerating the Sobel filtering, but to the

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

best of theirr knowledge they are the first to propose a 5×5 convolution kernel implementation using OpenCL. In this work [38], they implement the Sobel filter, one of the most effective and popular edge detection algorithms in image processing, in the OpenCL programming language.

In the study [39], authors propose a hybrid scheme that considers both weighted guided image filtering and the Sobel mask for accurate edge detection. The weighted guided image filtering enhances edges, while the Sobel mask is used for edge detection.

Scientists [40] try to reduce Sobel algorithm disadvantage that is the detection effect is not ideal when the image contains noise. In order to solve this problem, they propose an optimized scheme for edge detection. In this scheme, the weighted nuclear norm minimization image denoising algorithm is combined with the Sobel edge detection algorithm.

Research [41] introduces a quantum improved Sobel edge detection algorithm with non-maximum suppression and double threshold techniques for novel enhanced quantum representation method.

Joshi, R., & et al. [42] proposed a novel CMOS VLSI bit-sliced near-memory computing architecture for rapid Sobel edge detection for IoT edge devices to address this issue. The proposed architecture is compact, modular, scalable, and capable of processing a single image in a constant amount of time, irrespective of image resolution.

So we see that the Sobel algorithm has a series of advantages and disadvantages. And many scientists are exploring the possibilities of detecting the edges of objects using this algorithm. Next, we will consider the application of this algorithm to detect the edges of an object in the mobile robot’s workspace.

The Sobel Algorithm Implementation for Detection an Object Contour in Real Time

The Sobel algorithm is an operator for detecting edges in an image. It is used to distnguish intensity changes in an image, which often correspond to the boundaries of objects. The Sobel operator is based on convolving an image with two Sobel kernels (masks), one for horizontal edge selection and the other for vertical edge extraction. The use of these kernels allows one to calculate intensity derivatives along both axes, and then combines the results, usually using a gradient approximation method.

Let us describe the Sobel algorithm for detection the object contour in real time in the form of mathematical expressions.

Let $I(x, y)$ – is the pixel intensity in coordinates (x, y).

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

Then the Horizontal (G_x) and vertical (G_y) gradient of the image can be expressed by the following formulas:

$$G_x = \sum_{i=-1}^{1}\sum_{j=-1}^{1} I(x+i, y+j)\cdot S_x(i,j), \qquad (1)$$

$$G_y = \sum_{i=-1}^{1}\sum_{j=-1}^{1} I(x+i, y+j)\cdot S_y(i,j), \qquad (2)$$

G_x, G_y – are the horizontal and vertical gradient of an image, which is calculated using the Sobel operator;

$\sum_{i=-1}^{1}$ – horizontal summation operation on a 3x3 matrix around each pixel;

$\sum_{j=-1}^{1}$ – vertical summation operation on a 3x3 matrix around each pixel;

$I(x+i, y+j)$ – is the intensity of the pixel in the image in coordinates $(x+i, y+j)$;

$S_x(i,j)$ – element of the Sobel kernel matrix horizontally at position (i,j);

$S_y(i,j)$ – element of the Sobel kernel matrix vertically in position (i,j).

The next step is to calculate the gradients G_x, G_y horizontally and vertically, respectively

$$S_x = \begin{bmatrix} -1 0 1 \\ -2 0 2 \\ -1 0 1 \end{bmatrix}; \quad S_y = \begin{bmatrix} -1 -1 -1 \\ 0 \;\; 0 \;\; 0 \\ 1 \;\; 2 \;\; 1 \end{bmatrix}. \qquad (3)$$

Based on the results, the calculation of gradients G_x, G_y horizontally and vertically allows for a "gradient calculation," which is the amount of change in the intensity (brightness) of the image at each point. In the Sobel algorithm, the gradient is calculated horizontally G_x and vertically G_y and the magnitude of the gradient (G) is defined as the Euclidean norm of these two components:

$$G = \sqrt{G_x^2 + G_y^2}. \qquad (4)$$

This means that for each point in the image, the gradient magnitude is the length of the gradient vector at that point. The magnitude of the gradient shows how quickly the intensity of the image changes at a given point and in what direction.

The higher the gradient value, the more pronounced the boundaries of objects are. Gradient magnitude can be used to detect edges, boundaries, and textures in an image, making it useful in image processing operations such as edge extraction.

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

It remains to calculate the gradient inclination angle (θ) – this is the angle that indicates the direction of the greatest change in intensity (brightness) in the image at each point. In the context of the Sobel algorithm, the slope of the gradient is calculated using the Horizontal G_x and vertical components G_y of the gradient, and is calculated by the formula:

$$\theta = arctg(\frac{G_y}{G_x}). \quad (5)$$

The slope of the gradient (θ) is measured relative to the horizontal axis and indicates the direction in which the intensity changes. For example, if the angle is 0 degrees, this means that the greatest change in intensity occurs along the horizontal axis. The 90 degree angle indicates the vertical direction of the intensity change.

Gradient slope values are an important parameter used in various image processing algorithms, such as edge detection, object recognition and other computer vision tasks.

Thus, the Sobel algorithm extracts object boundaries using convolution operations with horizontal and vertical Sobel kernels. The result is two images G_x and G_y which can be combined to produce the final gradient image G.

This gradient image is widely used for edge detection in Computer Vision Systems, including real-time image processing on mobile robots.

Software implementation and experiments

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

```
# Open video stream from camera (usually 0 for built-in camera)
cap = cv2.VideoCapture(0)
```

This piece of code is responsible for opening a video stream from a camera using the OpenCV library. Thus, after executing this piece of code, the cap variable becomes associated with the video stream from the camera, and makes it possible to use it for subsequent frame capture, processing and display.

```
# Convert an image to grayscale

gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

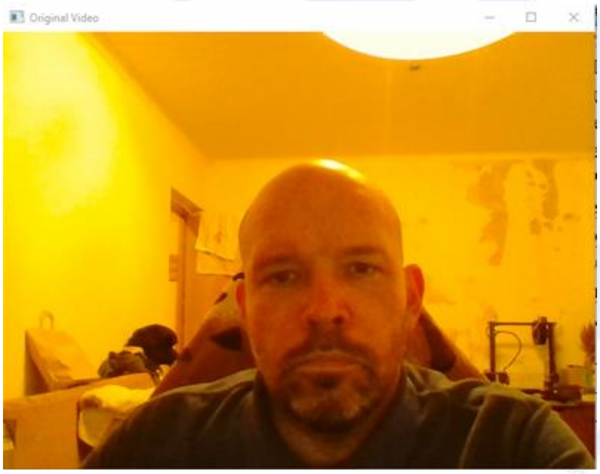
This piece of code converts a color image in BGR (Blue, Green, Red) format to a grayscale image. The process is carried out using the cvtColor (color conversion) function from the OpenCV library.

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

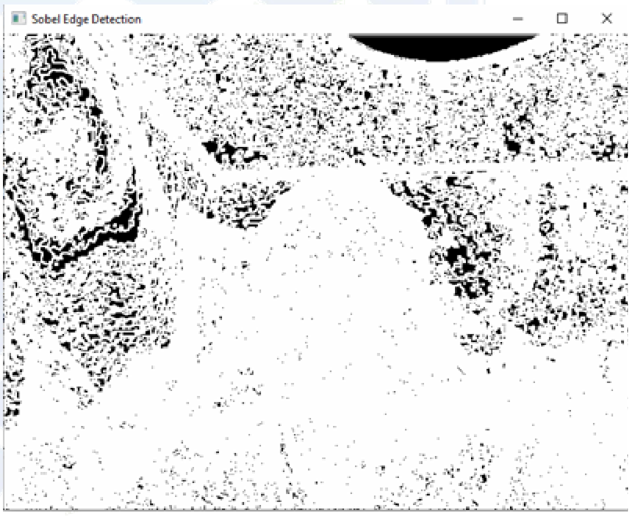
```
# Using the Sobel operator

    sobel_x = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=3)
    sobel_y = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=3)
    edges = cv2.magnitude(sobel_x, sobel_y)
```

This code snippet uses the OpenCV library to apply the Sobel operator to a grayscale image. The Sobel operator calculates the gradient of an image, which allows you to highlight the boundaries of objects in the image. Thus, the edges variable will contain an image where each pixel represents the magnitude of the gradient at that point, which ensures that the edges of objects in the original image are highlighted.

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 8Gb), Web-camera HD WebCam, OS Windows 10 Pro (Version 22H2). A program for implementing the Sobel algorithm for obtaining the contour of an object in real time from a camera was developed in the PyCharm 2022.2.3 (Professional Edition) environment in Python. The results of the program are presented in Figure 1.

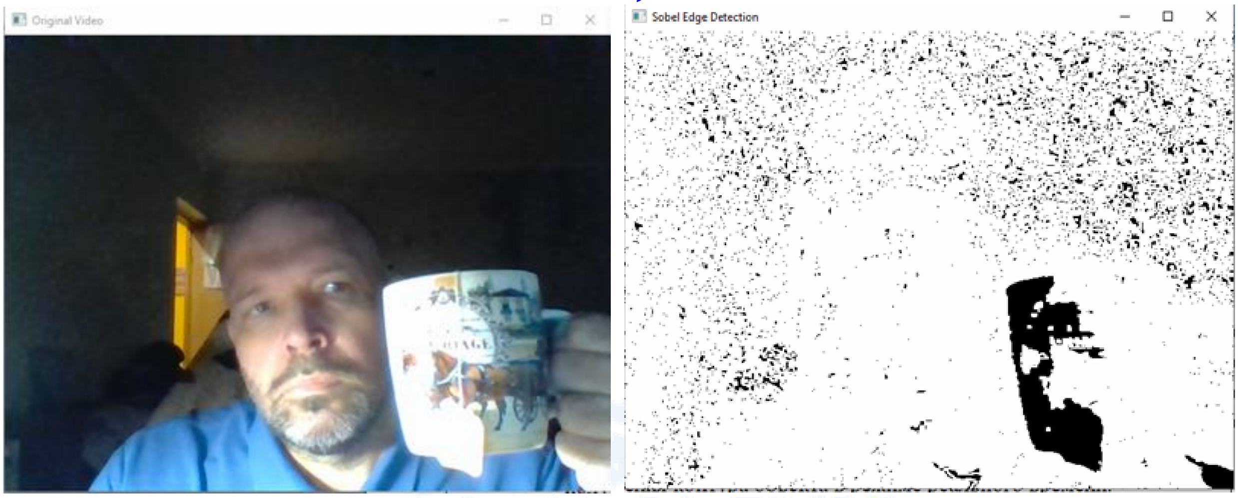


a)



b)

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3



c) d)

a), c) – Original Video; b), d) – Sobel Edge Detection

Figure 1: Results of the Sobel algorithm implementing program for detection the object contour in real time.

The results obtained (Fig. 1.a, b) of the Sobel algorithm indicate its high performance when processing video streams, reaching a speed of an impressive 30.00 frames per second. At the same time, the speed of constructing object contours is 1.19 frames per second, which confirms the effectiveness of the algorithm in real time. At the same time, the results are presented in Figure 1.c, d; they revealed a high speed of processing video frames at the level of 30.00 frames per second, which emphasizes the efficiency of the algorithm in real time. However, the edge rendering speed of 0.77 frames per second indicates a potential performance penalty when performing additional calculations to generate the gradient image. This is due to the fact that the streaming video was obtained in the dark using lateral artificial lightening.

Conclusion

In the study conducted on the Sobel algorithm implementation for detection the objects contour in the mobile robot's workspace in real time, mathematical models of the algorithm functioning were considered. The developed program in Python, implemented in the PyCharm development environment, went through a series of experiments, the results of which indicate the high performance of the algorithm. The processing speed of video frames was 30.00 frames per second, which indicates its efficiency in real time.

Thus, the results of the study not only highlight the effectiveness of the Sobel algorithm in real-time conditions, but also highlight the importance of considering lighting characteristics when using video processing algorithms on mobile robots.

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

Further improvements could include adapting the algorithm to different lighting conditions to improve stability and overall system performance.

References:

1. 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.

2. Orobinskyi, P., & et al.. (2020). Comparative Characteristics of Filtration Methods in the Processing of Medical Images. American Journal of Engineering Research, 9(4), 20-25.

3. 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.

4. 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.

5. 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.

6. Гиренко, А. В., Ляшенко, В. В., Машталир, В. П., & Путятин, Е. П. (1996). Методы корреляционного обнаружения объектов. Харьков: АО “БизнесИнформ, 112.

7. 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.

8. 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.

9. Lyubchenko, V., Matarneh, R., Kobylin, O., & Lyashenko, V. (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.

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3
10. 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.

11. 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.

12. 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.

13. Babker, A., & Lyashenko, V. (2018). Identification of megaloblastic anemia cells through the use of image processing techniques. Int J Clin Biomed Res, 4, 1-5.

14. Abu-Jassar, A., & et al. (2023). Obstacle Avoidance Sensors: A Brief Overview. Multidisciplinary Journal of Science and Technology, 3(5), 4-10.

15. Yevsieiev, V., & et al. (2024). Using Contouring Algorithms to Select Objects in the Robots' Workspace. Technical Science Research In Uzbekistan, 2(2), 32-42.

16. Maksymova, S., & et al. (2022). Development of an Automated System of Terminal Access to Production Equipment Using Computer Vision. In Manufacturing & Mechatronic Systems 2022: Proceedings of VIst International Conference, 22-23.

17. Akopov, M., & et al. (2023). Choosing a Camera for 3D Mapping. Journal of Universal Science Research, 1(11), 28-38.

18. Yevsieiev, V., & et al. (2024). Active Contours Method Implementation for Objects Selection in the Mobile Robot's Workspace. Journal of Universal Science Research, 2(2), 135-145.

19. Nevliudov, I. Sh., & et al. (2023). Conveyor Belt Object Identification: Mathematical, Algorithmic, and Software Support. Appl. Math. Inf. Sci. 17(6), 1073-1088.

20. Maksymova, S., & et al. (2023). Selection of Sensors for Building a 3D Model of the Mobile Robot's Environment. In Manufacturing & Mechatronic Systems 2023: Proceedings of VIIst International Conference (M&MS), Kharkiv, 33-35.

21. 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.

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

22. 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.

23. 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.

24. 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.

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

26. 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.

27. Al-Sharo Y., & et al. (2023). A Robo-hand prototype design gripping device within the framework of sustainable development. Indian Journal of Engineering, 20, e37ije1673.

28. Lyashenko, V., & et al. (2023). Automated Monitoring and Visualization System in Production. Int. Res. J. Multidiscip. Technovation, 5(6), 09-18.

29. Sotnik, S., Matarneh, R., & Lyashenko, V. (2017). System model tooling for injection molding. International Journal of Mechanical Engineering and Technology, 8(9), 378-390.

30. Lyashenko, V., Ahmad, M. A., Sotnik, S., Deineko, Z., & Khan, A. (2018). Defects of communication pipes from plastic in modern civil engineering. International Journal of Mechanical and Production Engineering Research and Development, 8(1), 253-262.

31. 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.

32. 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.

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

33. Lynn, N. D., & et al. (2021). Implementation of Real-Time Edge Detection Using Canny and Sobel Algorithms. In IOP Conference Series: Materials Science and Engineering, IOP Publishing, 1096(1), 012079.

34. Pengo, T., & Chaikan, P. (2021). High performance and energy efficient sobel edge detection. Microprocessors and Microsystems, 87, 104368.

35. Han, L., & et al. (2020). Research on edge detection algorithm based on improved sobel operator. In MATEC Web of Conferences, EDP Sciences, 309, 03031.

36. Ravivarma, G., & et al. (2021). Implementation of Sobel operator based image edge detection on FPGA. Materials Today: Proceedings, 45, 2401-2407.

37. As, R. A., & Gopalan, S. (2022). Comparative Analysis of Eight Direction Sobel Edge Detection Algorithm for Brain Tumor MRI Images. Procedia Computer Science, 201, 487-494.

38. Sanida, T., & et al. (2020). A heterogeneous implementation of the Sobel edge detection filter using OpenCL. In 2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST), IEEE, 1-4.

39. Ranjan, R., & Avasthi, V. (2023). Edge Detection Using Guided Sobel Image Filtering. Wireless Personal Communications, 132(1), 651-677.

40. Tian, R., & et al. (2021). Sobel edge detection based on weighted nuclear norm minimization image denoising. Electronics, 10(6), 655.

41. Chetia, R., & et al. (2021). Quantum image edge detection using improved Sobel mask based on NEQR. Quantum Information Processing, 20, 1-25.

42. Joshi, R., & et al. (2022). Fast Sobel edge detection for iot edge devices. SN Computer Science, 3(4), 302.