КОМПЬЮТЕРНЫЕ НАУКИ, ПРИБОСТРОЕНИЕ И АВТОМАТИЗАЦИЯ
КОМПЬЮТЕРЛІК ҒЫЛЫМДАР, АСПАП ЖАСАУ ЖӘНЕ АВТОМАТТАНДЫРУ
COMPUTER SCIENCE, INSTRUMENTATION AND AUTOMATION

UDC 004.42
IRSTI 50.41.25
https://doi.org/10.53364/24138614_2026_40_1_21

A.K. Kereev[1*], A.O. Zhaylybayeva[1], A.K. Tashimova[1],
L.Y. Kaparova[1], B.G. Umirzakova[2]
[1]K. Zhubanov Aktobe Regional University, Aktobe, Kazakhstan

[*]E-mail: adiljan@mail.ru

ANALYSIS OF CONTOUR DETECTION ALGORITHMS FOR MOBILE ROBOTS

Abstarct. *An important task for autonomous robots is to navigate safely in unfamiliar environments, potentially using computer vision to detect and recognize obstacles. Vision-based control systems have been developed for several years. Some rely on artificial landmarks, while more advanced systems make use of natural landmarks.The latter approach is preferable when a robot must operate in real, unstructured environments.*

In the field of autonomous robot navigation, which includes map building, path planning, and self-localization, this work develops the concept of a simple autonomous agent relying exclusively on visual information. The integrated navigation system reproduces certain functions of natural systems, as it requires minimal prior knowledge, limited onboard computation, and lacks an omnidirectional field of view. Since the goal is to move the robot across the floor while avoiding obstacles and people, the camera is mounted on top of the robot in a fixed forward-facing position.

The article focuses on one of the fundamental tasks of image processing—detecting the boundaries of objects in the observed scene. The aim of the research is to study contour detection algorithms based on preliminary image filtering and to compare the proposed approaches with well-known edge detectors such as Sobel, Canny, and Laplacian of Gaussian. Preliminary filtering is used to suppress image noise and enhance edges.

The scientific novelty of the work lies in the development and experimental evaluation of a contour detection algorithm that incorporates a pre-filtering stage based on contrast enhancement, the Kalman filter, and Monte Carlo methods. This increases the robustness of video processing for mobile robots operating in noisy environments.

The Sobel, Canny, and LoG algorithms were comprehensively analyzed and compared using a set of metrics, including the number of lost pixels, mean squared error, normalized MSE, and the structural similarity index. This approach provided a deeper understanding of their effectiveness under different noise conditions.

Keywords: *computer vision, image processing algorithms, Kalman filter, Monte Carlo methods, image segmentation, Laplacian of Gaussian, Sobel and Canny algorithms, real-time video analysis.*

Introduction.

One of the fundamental problems in the field of image processing is the extraction of object contours and other elements of the observed scene. This can also be used for robot orientation based on natural landmarks [1,2,3]. In general, contour extraction is used to significantly reduce the mount of data in an image while preserving structural properties that can be used for further image processing [4]. Contours of objects and scene elements can also serve as key features when matching heterogeneous images obtained, for example, from sensors of different types [5].

The result of boundary extraction is a set of connected curves that represent the contours of objects. In real images, it is often impossible to extract all points belonging to the contours. Frequently, the contour image produced by an algorithm has shortcomings, such as missing observable boundaries, gaps, or the presence of false boundaries that do not belong to objects or segments in the scene. The task of contour extraction becomes significantly more difficult when the object boundary is blurred or insufficiently smooth, which may be affected by noise and various characteristics of the image acquisition process.

A large number of studies have been devoted to the problem of contour extraction. Most algorithms are based on computing the gradient (the magnitude of intensity change in a specific direction) by convolving the image with a kernel—a gradient operator. Many such operators have been proposed: the cross Roberts operator [6], the Sobel operator [7], the Prewitt operator [8], the discrete Laplacian operator [9], the Kirsch operator [10], the Robinson operator [11], and others. A gradient operator may be represented by one or several kernels for computing gradients in different directions—for example, vertically, horizontally, or diagonally.

Often, the first step of a contour extraction algorithm is image preprocessing in order to improve the final result. For instance, blurring is commonly used to suppress noise in an image. In this work, the quality of contour extraction algorithms will be evaluated on both original and preprocessed images. The following approaches to preliminary image enhancement are considered: contrast enhancement, image correction using the Kalman filter, and the Monte Carlo method.

This work proposes a contour extraction algorithm consisting of two stages. The first stage is image preprocessing aimed at improving the final result. The second stage is the contour extraction itself. It should be noted that preprocessing is optional and can also be used together with well-known edge detection algorithms.

Materials and Methods.

One of the main tasks of computer vision is object recognition in images. To achieve this, classification and detection methods are employed, such as Support Vector Machine (SVM), Cascade Classifier method (see Figure 1 and Deep Learning methods) [12,13].

These techniques are widely used for tasks that involve detecting and identifying objects in static images or video streams. SVM is a supervised machine learning algorithm that is used for classification and regression tasks, often applied in image recognition. Cascade Classifiers are particularly effective for object detection in real-time applications, such as face detection, as they use a series of increasingly complex classifiers. Deep Learning, particularly Convolutional Neural Networks (CNNs), have revolutionized the field by allowing for more complex feature extraction and learning patterns from large amounts of data, leading to improved object detection and recognition.

These methods form the core of many modern computer vision applications and are crucial for tasks related to image processing, real-time video stream analysis, and autonomous systems.

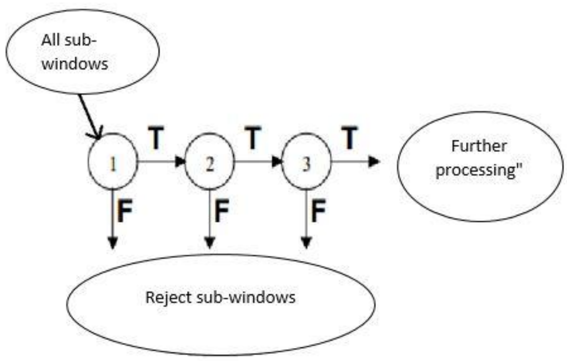


Figure 1 – Schematic of the Detection Cascade

The general form of the detection process is a degenerate decision tree, which we refer to as a "cascade." A positive result from the first classifier triggers the evaluation of the second classifier, which has also been tuned for very high detection performance. A positive result from the second classifier triggers the third classifier, and so on. A negative outcome at any stage immediately leads to the rejection of the sub-window.

An important aspect of computer vision is image segmentation, which involves dividing an image into separate parts to extract more detailed information. Various algorithms are applied for this purpose, such as thresholding methods, region growing methods, and graph-based optimization methods (see Figure 2).

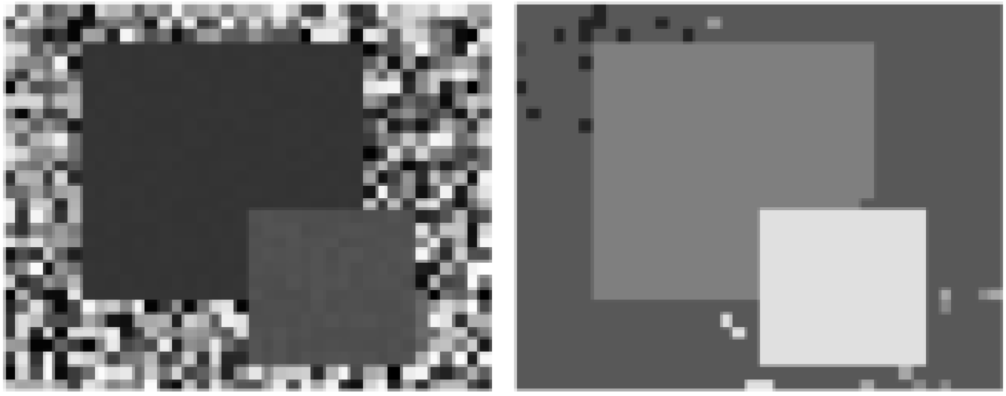


Figure 2 – Synthetic Image (Gray Image 40 × 32) and Segmentation Using Graph-based Optimization Based on Similarity to Neighboring Graphs

Another important task in computer vision is object tracking in video. For this purpose, optimization methods such as the Kalman Filter and Monte Carlo Methods are used [14, 15].

The Kalman Filter is an algorithm used to estimate the state of a system based on incomplete and noisy data. It was developed by Rudolf Kalman in 1960 and has found wide applications in various fields, including automatic control, navigation, and signal processing.

The main idea of the Kalman Filter is that it combines system measurements and a model of its dynamics to obtain the best estimate of the current state of the system. This is done by estimating mathematical expectations and covariance matrices, which represent measures of uncertainty in the measurements and the model.

To formalize the Kalman Filter algorithm, matrix algebra is used. The state of the system is represented as a state vector x, which describes all known parameters of the system. The system's dynamic model is given by the transition matrix F and the control vector u. The system measurement is given by the measurement vector z and the measurement matrix H.

At the first step of the Kalman Filter, the initial system state x0 and the initial covariance matrix P0, representing the uncertainty in the initial estimate, are evaluated. Then, at each subsequent step, the algorithm performs the following steps:

Prediction of the System's State:

$$x_{k-} = F_{k-1}x_{k-1} + Bu_{k-1} \quad (1)$$

where:

- k is the current step number;
- "-" denotes the predicted value;
- F_{k-1} is the transition matrix at the current step;
- Bu_{k-1} is the control vector at the current step

Prediction of the Covariance Matrix:

$$Pk- = F_{k-1}P_{k-1}F_{k-1}T + Q_{k-1} \quad (2)$$

where: Q_{k-1} - is the covariance matrix of the model noise.

Update of the System's State Estimate:

$$K_k = P_k - H_kT(H_kP_kH_kT + R_k) - 1 \quad (3)$$

where: K_k - is the Kalman gain matrix
H_k - is the measurement matrix at the current step;
R_k - is the covariance matrix of the measurement noise at the current step.

Update of the System's State Estimate:

$$x_k = x_k - + K_k(z_k - K_kH_{k-}) \quad (4)$$

where z_k - is the measurement at the current step.

Update of the Covariance Matrix:

$$P_k = (I - K_kH_k)P_{k-} \quad (5)$$

where I is the identity matrix.

These steps are performed for each subsequent step of the Kalman Filter. At each step, the algorithm uses the new measurement of the system and the model of its dynamics to update the estimate of the current state of the system. The key point is the optimal combination of the measurement information and the system's dynamic model to obtain the best estimate of the system's state.

The Kalman Filter has many applications in various fields, including automatic control, navigation, and signal processing. For example, it can be used to control a missile, determining its current position and velocity based on sensor information. It can also be used in mobile devices to determine the user's location based on GPS signals and other sensors.

The Monte Carlo method is a statistical technique used to solve mathematical problems by generating random numbers and estimating statistical characteristics of the obtained results. It was developed in the 1940s as part of the Manhattan Project for modeling nuclear reactions, but today it is widely applied in many fields, including physics, biology, economics, and machine learning.

The main idea behind the Monte Carlo method is that it generates random numbers according to a given distribution and uses them to estimate integrals and mathematical expectations. To do this, the Monte Carlo method uses the law of large numbers, which states that as the number of random samples increases, the accuracy of the estimate improves.

To formalize the Monte Carlo method, mathematical statistics and probability theory are used. Sample means and variances are used to estimate statistical characteristics. For example, to estimate the integral of a function $f(x)$ over the interval $[a, b]$, the following formula can be used:

$$I = (b - a) * E[f(x)] \quad (6)$$

where $E[f(x)]$ is the expected value of the function f(x), which can be estimated using the sample mean:

$$E[f(x)] = 1/N * sum(f(xi)) \quad (7)$$

where N is the number of random samples, and x_i is the i-th random sample.

To estimate the variance of the integral estimate, the following formula can be used:

$$Var(I) = \frac{(b-a)^2}{N} * (E[f(x)^2] - E[f(x)]^2) \tag{8}$$

where $E[f(x)_2]$ - is the expected value of the square of the function $f(x)$, which can be estimated using the sample variance:

$$E[f(x)^2] = \frac{1}{N} * \sum (f(x_i)^2) \tag{9}$$

where N is the number of random samples, and x_i is the i-th random sample.

Real-time video stream analysis algorithms have wide applications in various fields, including computer vision, machine learning, robotics, and others. One of the key tasks in video stream analysis is to separate objects from the background and determine their movement and trajectory. To solve this task, various image and video processing algorithms are used, such as background subtraction algorithms, motion detection algorithms, and object tracking algorithms.

Background subtraction algorithms allow the static background to be separated from the moving objects in the video stream. Various methods are used for this, such as Gaussian model methods, median filtering methods, and sample-based learning methods. The most popular method is the Gaussian model method, which assumes that the intensity of background pixels follows a Gaussian distribution and can be described using a mathematical model.

The Gaussian model method is an image segmentation technique that uses a Gaussian distribution to model the intensity of background pixels. This method is widely used in various areas of computer vision, such as video surveillance, image processing, and pattern recognition.

The main assumption of the method is that the background pixel values at any point in the image can be described by a Gaussian distribution. Therefore, the background model can be described as:

$$P(x) = \frac{1}{\sqrt{2\pi\sigma}} exp(- \frac{(x-\mu)^2}{2\sigma^2} \tag{10}$$

where $P(x)$ - is the pixel intensity;
μ - is the mean intensity of the background;
σ^2 - is the variance of the background intensity.

For each pixel in the image, the background model can be applied to calculate the probability that this pixel is part of the background. If the probability is higher than a threshold, the pixel is considered part of the background; otherwise, it is considered part of the foreground.

One of the main advantages of the Gaussian model method is its simplicity and high processing speed. However, this method can have issues in cases of lighting changes, shadows, camera movement, and other scene anomalies. To address these issues, more complex methods, such as the Gaussian mixture model or techniques based on object movement modeling, can be used.

Motion detection algorithms allow for determining the movement of objects in a video stream. They are based on analyzing changes in pixel brightness and texture across consecutive frames of the video stream. To detect motion, optical flow and differential image methods are commonly used. Optical flow represents a vector field that describes the motion of objects in the video stream, while differential image methods are based on analyzing the difference in brightness between consecutive frames.

Differential image methods [16] are a class of computer vision methods that use the brightness difference between consecutive frames in the video stream to determine the movement of objects in the image. The main idea is that the change in brightness between two consecutive frames can be used to determine the motion of objects in the image.

One of the differential image methods is the optical flow method. It is a vector field that describes the movement of objects in the image. To calculate optical flow, the following formulas are used:

$$\frac{\partial I}{\partial x}u+\frac{\partial I}{\partial y}v+\frac{\partial I}{\partial t}u=0 \tag{11}$$

where I is the image brightness;
u and v are the components of optical flow;
$\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are the partial derivatives of image brightness with respect to the coordinates x, y, and time t.

This formula describes the condition that the brightness of an object in the image remains constant over time. From this, the components of the optical flow u and v can be expressed:

$$u=-\frac{\frac{\partial I}{\partial x}\frac{\partial I}{\partial t}}{\left(\frac{\partial I}{\partial x}\right)^2+\left(\frac{\partial I}{\partial y}\right)^2} \tag{12}$$

$$v=-\frac{\frac{\partial I}{\partial y}\frac{\partial I}{\partial t}}{\left(\frac{\partial I}{\partial x}\right)^2+\left(\frac{\partial I}{\partial y}\right)^2} \tag{13}$$

This formula describes the condition that the brightness of an object in the image remains constant over time. From this, the components of the optical flow \(u\) and \(v\) can be expressed.

Thus, optical flow can be calculated by computing the partial derivatives of image brightness and solving the system of equations.

One of the advantages of the optical flow method is its high accuracy in determining the movement of objects in an image. However, this method has some limitations, such as changes in lighting, fast movements of objects, and other factors that may lead to inaccuracies in the optical flow estimation.

Object tracking algorithms allow for tracking the trajectories of moving objects in a video stream. Various methods are used for this, such as model-based methods, camera calibration-based methods, and graph-based methods. The most popular method is the model-based approach, which assumes that the object's trajectory can be described using a mathematical model, such as the Kalman filter. This method allows for tracking the movement of an object even if it is partially occluded by other objects or if there is a change in lighting.

To implement real-time video stream analysis algorithms, computational limitations must be taken into account. Video stream processing requires significant computational resources, so optimized algorithms and specialized hardware solutions, such as graphical processing units (GPUs), must be used.

One of the main metrics for evaluating the quality of video stream analysis algorithms is the

accuracy of trajectory detection. Metrics such as the Mean Object Tracking Accuracy (MOTA) and Mean Object Tracking Precision (MOTP) are often used to assess accuracy. These metrics are used to evaluate the performance of computer vision algorithms in object tracking tasks.
MOTA represents the mean trajectory error and is defined as:

$$MOTA = 1 - \frac{\left(\Sigma(i)FN_i + \Sigma(i)FP_i + \Sigma(i)ID_{sw_i}\right)}{\Sigma(i)GT_i} \tag{14}$$

where:

FN_i - is a false negative result (missed detection);
FP_i - is a false positive result (false alarm);
ID_{sw_i-} - is an erroneous identification;
GT_i - is the ground truth.

MOTP represents the mean position error and is defined as:

$$MOTP = \frac{\Sigma(i)d_i}{\Sigma(i)c_i} \tag{15}$$

where:
d_i is the distance between the true and predicted position,
c_ii is the number of correspondences.

Both metrics are important for evaluating the performance of object tracking algorithms. MOTA measures the overall tracking accuracy, considering false alarms, misses, and identification errors. MOTP measures the accuracy in determining the positions of objects, which can be crucial in various applications.

However, it should be noted that these metrics can be sensitive to the choice of threshold values and may not account for certain aspects of the object tracking task. Therefore, when evaluating the performance of object tracking algorithms, it is important to use multiple metrics, perform error analysis, and consider the specifics of the task at hand.

Results and Discussion.

The Sobel, Canny, and Laplace-Gaussian (LoG) algorithms are popular image and video stream processing algorithms used for edge detection in images and videos. Let's take a closer look at each of these algorithms and perform a comparative analysis of their efficiency using Python code examples.

The Sobel algorithm is used to detect object boundaries in an image. It applies two operators: one for calculating the gradient in the horizontal direction, and another for calculating the gradient in the vertical direction. These operators are then used to determine the strength and direction of the gradient at each pixel. The result is an image where the object boundaries are highlighted in white, and the other pixels are black.

The Canny algorithm is also used for edge detection. It begins by blurring the image with a Gaussian filter, which removes noise and smooths the image. Then, gradients in the horizontal and vertical directions are calculated, and the strength and direction of the gradient are computed for each pixel. Next, a thresholding operation is performed, where pixels stronger than a defined threshold are marked as edges. Non-maximum suppression is also applied to find local maxima along the edges, and the rest of the pixels are suppressed to zero. Finally, edge narrowing is performed to produce a more precise image of the object boundaries.

The Laplace-Gaussian (LoG) algorithm is used to detect edges in an image. It starts by blurring the image with a Gaussian filter, then computes the Laplacian of the image, which is the second derivative in both the x and y directions. Points on the image where the Laplacian reaches a local maximum indicate the presence of an edge. The result of the algorithm is an image where the object edges are highlighted in white, and other pixels are black.

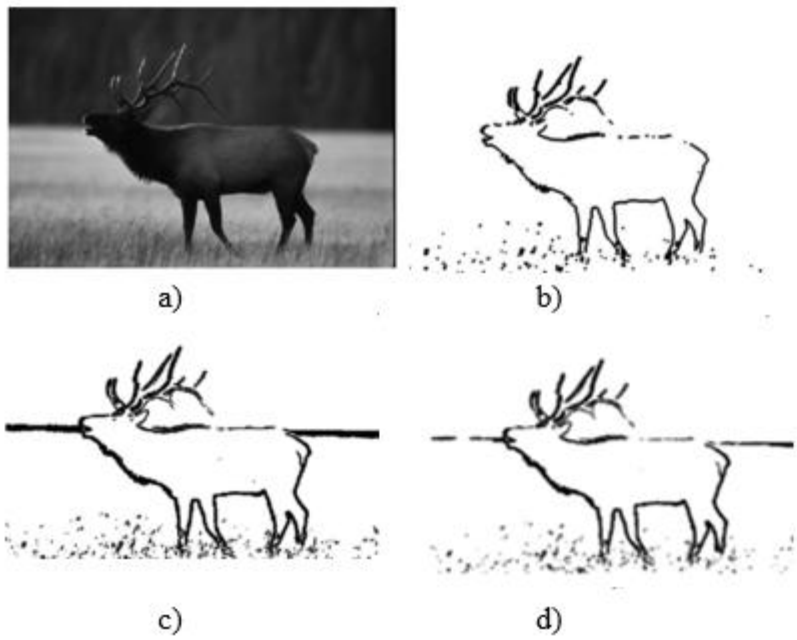


Figure 3 – Results of contour detection: a) original image; b) Sobel operator; c) Canny algorithm; d) Laplacian of Gaussian algorithm.

The following characteristics were selected as criteria for evaluating the quality of object contouring in images:

- Assessment of the number of lost pixels in the contour
- Robustness of contouring methods to noise
- Mean Squared Error (MSE) of contour detection
- Structural Similarity Index (SSIM)

The assessment of the number of lost pixels in the contour represents a comparison between the number of pixels belonging to the reference contoured image and the number of contour pixels in images processed by the software system using the presented contouring methods.

Robustness of contouring methods to noise is understood as the number of lost contour pixels when noise is added to the original image.

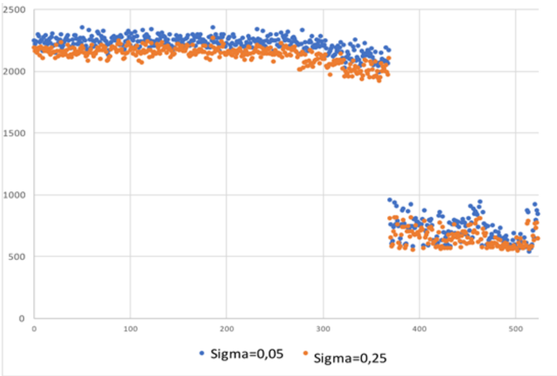
For testing purposes, Gaussian noise with different coefficients α was applied (simulating weak and strong noise levels).

The mean squared error was computed pixel-by-pixel between the reference contoured images and the contoured images obtained by the software system using the selected contouring methods.

The Structural Similarity Index compares small image fragments (typically 8×8 pixels). This evaluation method takes into account the “perception of error” by considering structural changes in the information. The SSIM index ranges from −1 to +1 (the value +1 is achieved only when the samples are completely identical):

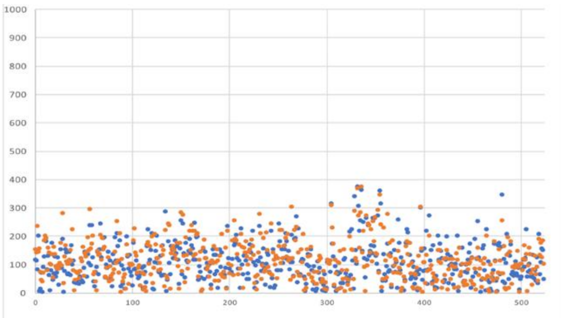
$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_1)} \quad (16)$$

After testing the methods for noise robustness, the following graphs were obtained:



a)

b)



c)

Figure 4 – Number of lost pixels after contouring: a) Laplacian of Gaussian; b) Sobel operator; d) Canny algorithm.

The gaps in the first two graphs are due to the different numbers of pixels in the original images. Images belonging to groups 4 and 5 have a smaller total number of pixels, and accordingly, the relative pixel loss is also lower.

It can also be observed that the number of lost pixels is smaller when using the Canny detector. Noise has little effect on the detection quality, and there is no clear boundary between losses under low noise and high noise conditions.

The average percentage of lost pixels for each method and image group is presented below:

Table 1 – Average Percentage of Lost Pixels

№	CANNY	SOBEL	LAPLAS
1	1.20%	16.49%	14.22%
2	0.49%	16.34%	16.26%
3	0.37%	16.26%	17.02%
4	1.46%	5.07%	8.18%
5	1.62%	5.22%	8.10%

As can be seen from the table, the Canny method shows a lower percentage of lost graphical information when noise is applied. The Sobel method performs better for image groups 3, 4, and 5 compared to the LoG method, which, in turn, handled groups 1 and 2 better than Sobel.

The mean squared error (MSE) for edge detection was calculated using the NumPy package for Python, which is designed for working with multidimensional arrays and matrices, as images are processed as multidimensional arrays.

The graph below presents the mean squared error values for the control group of images, which contains representatives from each class of the selected medical images.

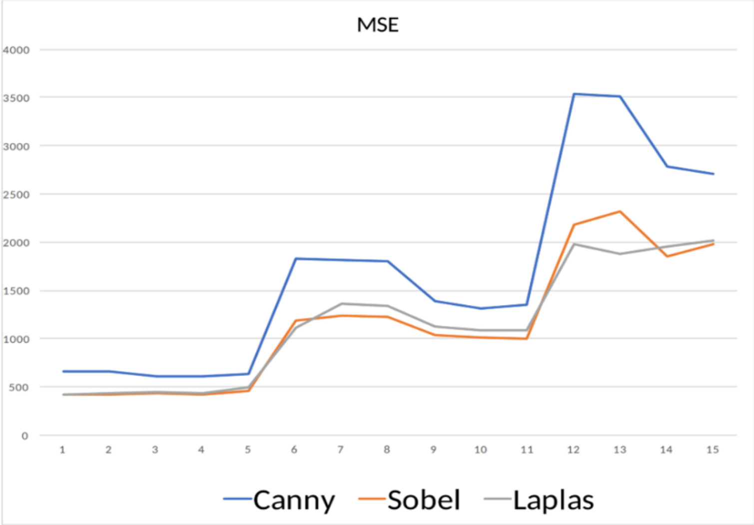


Figure 5 – Mean Squared Error of Edge Detection

As can be seen from the graph, the mean squared error (MSE) of edge detection for the Canny method is higher than for the other methods. This may be due to the double-threshold filtering applied in this method, which results in fewer false edges being detected. Since the difference (error) between the absence of an edge (black pixel) and the presence of an edge (white pixel) is maximal, a smaller total number of edges (including false ones) negatively affects the MSE value.

To account for this characteristic, a normalized mean squared error was used—that is, the error divided by the number of pixels in the edge. The results are shown in the graph below:

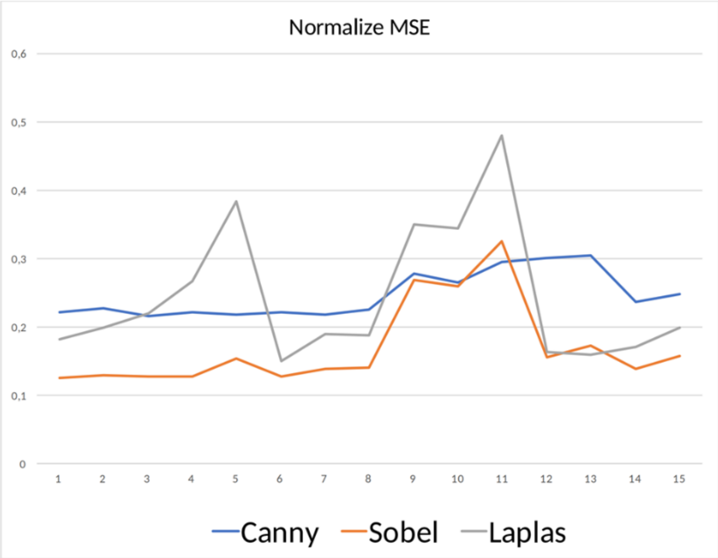


Figure 6 – Normalized Mean Squared Error

As can be seen from the graph, the error curve of the Canny method is smoother, and for all image groups, the error does not exceed 0.3 ± 0.05.

However, the mean squared error alone is insufficient for evaluating edge detection quality, so the Structural Similarity Index (SSIM) is additionally calculated.

For a more comprehensive assessment of edge detection quality, the Structural Similarity Index (SSIM) is used. The SSIM was computed using the NumPy and Pandas packages in Python.

The Structural Similarity Index indicates how closely images correspond to each other on a scale from –1 to 1.

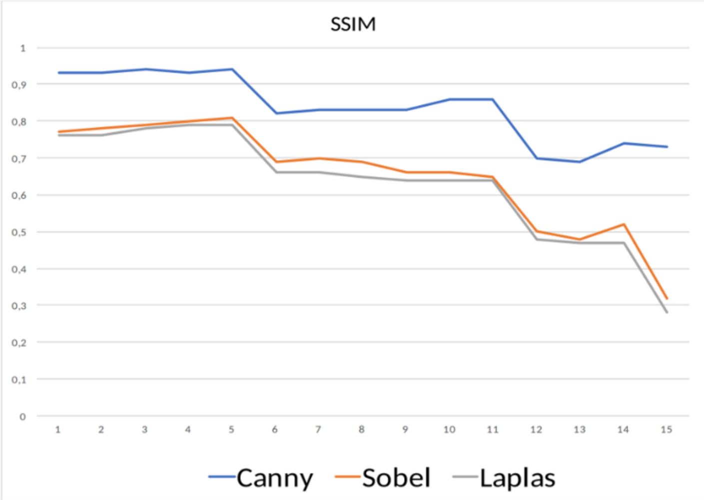


Figure 7 – Structural Similarity Index of the Control Group of Images for the Selected Edge Detection Methods

As can be seen from the graphs, the highest similarity coefficient is observed when using the Canny-based detector. The Sobel and LoG methods perform worst on image group 5.

Conclusions.

In conclusion, real-time video stream analysis algorithms have a wide range of applications in various fields, including video surveillance, automatic transportation control, medical diagnostics, and more. Key areas of application for real-time video stream analysis and processing algorithms in robotics have been identified.

The Sobel algorithm has a simple implementation and high processing speed, making it suitable for real-time video stream processing. However, it is sensitive to noise in the image and may produce false results on images with uneven backgrounds or where objects blend into the background.

The Canny algorithm is more accurate than the Sobel algorithm and provides clearer and more precise object boundaries. However, it is more resource-intensive and slower than Sobel, making it less suitable for real-time applications.

The Laplace-Gaussian algorithm is the most accurate among the three, but it is also the slowest and requires more resources. It can yield the best results on images with a significant amount of noise and complex structures, such as lines, points, and circles.

A quantitative relationship between edge detection quality and the level of noise distortion has been established, enabling the recommendation of the optimal method for computer vision systems in mobile robots. For the first time for this task, a comparison of methods using the SSIM metric was conducted, which made it possible to identify the advantages of the Canny detector and the limitations of the Sobel and LoG algorithms when working with real video streams.

The choice of video stream processing algorithm depends on the specific requirements of the task. If fast execution with minimal resource usage is needed, the Sobel algorithm is the best choice. If the highest quality results are required, the Laplace-Gaussian algorithm should be used. The Canny algorithm can be a good middle ground when more precise processing is needed, but without the high resource demands of the Laplace-Gaussian method.

References

1. Marx, S., Gusenburger, J., Bashir, A. and Müller, R. (2024). "Low-Cost Stereo Vision: A Single-Camera Approach For Precise Robotic Perception," 2024 IEEE 20th International Conference on Automation Science and Engineering (CASE), Bari, Italy, pp. 890-896, doi: 10.1109/CASE59546.2024.10711598.
2. Rao, J, Bian, H, Xu, X, Chen J. (2023). Autonomous Visual Navigation System Based on a Single Camera for Floor-Sweeping Robot. Applied Sciences. 13(3):1562. https://doi.org/10.3390/app13031562
3. Xie Ziyang, Lu Lu, Wang Hanwen, Li Li, Fitts Edward, Xu Xu. (2022). A human-robot collision avoidance method using a single camera. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 66. 2244-2248, doi: 10.1177/1071181322661540.
4. Yang, Yi & Jia, Chengzhen. (2025). Image Denoising and Image Contour Detection. Highlights in Science, Engineering and Technology. 131. 14-23. 10.54097/wd660x73.
5. Z. -F. Zheng et al. (2022). "Optimal OPC model selection with SEM image contours," 2022 International Workshop on Advanced Patterning Solutions (IWAPS), Beijing, China, pp. 1-4, doi: 10.1109/IWAPS57146.2022.9972274.
6. Rosenfeld, A. (2021). "The Max Roberts Operator is a Hueckel-Type Edge Detector," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-3, no. 1, pp. 101-103, Jan. doi: 10.1109/TPAMI.1981.4767056.
7. Song, Y. and Liu, Y. (2024). "Lightweight Image Super-Resolution Reconstruction Based on Edge Detection," 2024 5th International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Shenzhen, China, pp. 292-296, doi: 10.1109/ICBAIE63306.2024.11116927.
8. Verma, A. Dhanda, N. and Yadav, V. (2024). "Enhancing Image Segmentation through an Innovative Hybrid Kernel: A Weighted Fusion of Sobel and Prewitt Operators Using Artificial Neural Networks," 2024 International Conference on Computing, Sciences and Communications (ICCSC), Ghaziabad, India, pp. 1-5, doi: 10.1109/ICCSC62048.2024.10830382.
9. Zhixiang Xu, Qiang Xia and Jianfeng Chen (2021). "The performances of the Laplacian of binomial distribution and the discrete Laplacian of Gaussian edge detection operators," IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century, Vancouver, BC, Canada, pp. 673-678 vol.1, doi: 10.1109/ICSMC.2021.537841.
10. Srisabarimani, K., Kumar, G. D., A. R and Lakshmi, B.D. (2025). "Comparative Study of Brain Tumor Types Interaction with Various Edge Detection Operators," 2025 International Conference on Emerging Systems and Intelligent Computing (ESIC), Bhubaneswar, India, pp. 25-31, doi: 10.1109/ESIC64052.2025.10962756.
11. Ren, H., Sun, J. and Yuan, W. (2020). "Research on Image Edge Detection Method Based on Improved Robinson Operator and Otsu Algorithm," 2020 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China, pp. 1536-1541, doi: 10.1109/ICMA49215.2020.9233791.
12. Srisabarimani, K., Kumar, G. D., A. R and Lakshmi, B. D. (2025). "Comparative Study of Brain Tumor Types Interaction with Various Edge Detection Operators," International Conference on Emerging Systems and Intelligent Computing (ESIC), Bhubaneswar, India, 2025, pp. 25-31, doi: 10.1109/ESIC64052.2025.10962756.
13. Vidya, S. D., Sagar and Prabhakar, C. J. (2022)."A Vison Based Lane Detection Approach Using Vertical Lane Finder Method," 2022 International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Shivamogga, India, pp. 207-211, doi: 10.1109/DISCOVER55800.2022.9974938.
14. Ma, Q. and Wang, M. (2010). "Object tracking in infrared image sequence by Monte-Carlo method," 2010 3rd International Congress on Image and Signal Processing, Yantai, China, pp. 353-357, doi: 10.1109/CISP.2010.5648033.

15. Dong, M., Li, C., Wang, Y., Chen, Z., Wang, Y. and Liu, X. (2025). "Comparative Analysis of the Effect of Common Methods of Trajectory," 2025 37th Chinese Control and Decision Conference (CCDC), Xiamen, China, pp. 3962-3967, doi: 10.1109/CCDC65474.2025.11090776.

16. Song, W. (2021). "Research on the Application of Mathematical Methods in Image Denoising Technology," 2021 2nd International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI), Shenyang, China, pp. 19-22, doi: 10.1109/ICHCI54629.2021.00011.

МОБИЛЬДІ РОБОТТАР ҮШІН КОНТУРЛЫҚ ДЕТЕКЦИЯ АЛГОРИТМДЕРІН ТАЛДАУ

***Аңдатпа**. Автономды роботтардың маңызды міндеттерінің бірі – таныс емес ортада қауіпсіз қозғалу, мүмкіндігінше жасанды көру арқылы кедергілерді анықтау және тану болып табылады. Визуалды басқару жүйелері бірнеше жылдан бері дамытылып келеді. Олардың кейбірі жергілікті жасанды нысандарды қолданса, ал жетілдірілгендері табиғи нысандарға сүйенеді.*

Роботтардың автономды қозғалысы саласында, оған карталар жасау, маршрут жоспарлау және өздігінен орналасу (самолокация) кіреді, жұмыс визуалды ақпаратқа ғана сүйенетін қарапайым автономды агент идеясын дамытады. Интеграцияланған навигациялық жүйе табиғи жүйелердің кейбір функцияларын қайта жасайды, себебі ол аз алдын-ала білімді, борттағы есептеулерді қолданады және барлық бағытты көру мүмкіндігіне ие емес. Мақсат – роботты еденде кедергілер мен адамдардан аулақ жылжыту болғандықтан, камера роботтың үстіне орнатылып, алға қаратып бекітілген.

Мақалада кескіндерді өңдеу саласындағы негізгі міндеттердің бірі қарастырылған, ол бақылау көрінісіндегі объектілердің шеттерін бөліп көрсетуге байланысты. Жұмыстың мақсаты – алдын ала сүзгілеуге негізделген объектілердің контурларын бөліп көрсету алгоритмдерін зерттеу, ұсынылған тәсілдерді белгілі Sobel, Canny және Laplace-Gauss шет детекторларымен салыстыру. Алдын ала сүзгілеу қолдану кескіндегі шуды басуға және шеттерді айқындауға мүмкіндік береді.

Ғылыми жаңалығы – контрастты арттыруға негізделген алдын ала сүзгілеу, Калман фильтрі және Монте-Карло әдістерін қамтитын контурларды бөліп көрсету алгоритмін жасау және тәжірибелік бағалау, бұл мобильді роботтың видеопотокты өңдеуде шуға төзімділігін арттыруға мүмкіндік береді.

Sobel, Canny және LoG алгоритмдері бірқатар метрикалар бойынша кешенді түрде зерттеліп, салыстырылды (жоғалған пиксельдер саны, MSE, нормаланған MSE және құрылымдық ұқсастық индексі SSIM), бұл әртүрлі зақымдану деңгейінде олардың тиімділігін тереңірек түсінуге мүмкіндік берді.

***Түйін сөздер**: компьютерлік көру, бейнелерді өңдеу алгоритмдері, Калман сүзгісі, Монте-Карло әдістері, бейнелерді сегментациялау, Собель және Канни алгоритмдері, нақты уақыттағы бейне талдау.*

АНАЛИЗ АЛГОРИТМОВ КОНТУРНОГО ДЕТЕКТИРОВАНИЯ ДЛЯ МОБИЛЬНЫХ РОБОТОВ

***Аннотация.** Важной задачей автономных роботов является безопасное передвижение в незнакомой среде, возможно, с использованием искусственного зрения для обнаружения и распознавания препятствий. Системы визуального управления разрабатываются уже несколько лет. Некоторые из них используют искусственные ориентиры на местности, в то время как более совершенные полагаются на естественные ориентиры.*

В области автономного передвижения роботов, включающей создание карт, планирование маршрута и самолокацию, в работе развивается идея простого автономного агента, опирающегося только на визуальную информацию. Интегрированная навигационная система воспроизводит некоторые функции естественных систем, поскольку использует мало априорных знаний, бортовых вычислений и не имеет всенаправленного обзора. Поскольку цель, состоит в том, чтобы перемещать робота по полу, избегая препятствий и людей, камера находится сверху робота и установлена в фиксированном положении, чтобы смотреть вперед.

В статьи рассмотрена одна из базовых задач в области обработки изображений, связанная с выделением границ объектов наблюдаемой сцены. Целью работы является исследование алгоритмов выделения контуров объектов на изображении на основе предварительной фильтрации, сравнение предлагаемых подходов с известными детекторами границ Собеля, Кэнни и Лапласа-Гаусса. Использование предварительной фильтрации обусловлено возможностью подавить шумовую составляющую изображения, а также подчеркнуть границы.

Научная новизна работы заключается в разработке и экспериментальной оценке алгоритма выделения контуров, включающего этап предварительной фильтрации изображения на основе повышения контраста, фильтра Калмана и методов Монте-Карло, что позволяет повысить устойчивость обработки видеопотока мобильного робота к шумам.

Комплексно исследованы и сравнены алгоритмы Sobel, Canny и LoG по совокупности метрик (количество потерянных пикселей, MSE, нормализованный MSE и индекс структурного сходства SSIM), что обеспечило более глубокое понимание их эффективности в условиях различной зашумлённости.

Ключевые слова: *компьютерное зрение, алгоритмы обработки изображений, фильтр Калмана, методы Монте-Карло, сегментация изображений, OpenCV, алгоритмы Собеля и Канни, анализ видео в реальном времени.*

Авторлар туралы мәлімет

Кереев Адилжан Кутымович	PhD, Қ.Жұбанов атындағы Ақтөбе өңірлік университеті, информатика және ақпараттық технологиялар кафедрасының доценті, Ақтөбе қ., Қазақстан E-mail: .adiljan@mail.ru
Жайлыбаева Айдана Омарқызы	магистр, Қ.Жұбанов атындағы Ақтөбе өңірлік университеті, информатика және ақпараттық технологиялар кафедрасының оқытушысы, Ақтөбе қ., Қазақстан E-mail: aidana_6143@mail.ru
Ташимова Анар Кошимовна	магистр, Қ.Жұбанов атындағы Ақтөбе өңірлік университеті, информатика және ақпараттық технологиялар кафедрасының аға оқытушысы, Ақтөбе қ., Қазақстан E-mail: anar_6868@mail.ru
Капарова Лэззат Есенгалиевна	магистр, Қ.Жұбанов атындағы Ақтөбе өңірлік университеті, информатика және ақпараттық технологиялар кафедрасының оқытушысы, Ақтөбе қ., Қазақстан E-mail: lazzat_09@mail.ru
Умирзакова Бану Габиденовна	магистр, Қ.Жұбанов атындағы Ақтөбе өңірлік университеті, информатика және ақпараттық технологиялар кафедрасының оқытушысы, Ақтөбе қ., Қазақстан E-mail: .banoka_98@mail.ru

Сведение об авторах

Кереев Адилжан Кутымович	PhD, доцент кафедры информатики и информационных технологий, Активобе региональный университет им. К. Жубанова, Активобе, Казахстан E-mail: .adiljan@mail.ru

Жайлыбаева Айдана Омарқызы	магистр, преподаватель кафедры информатики и информационных технологий, Активобе региональный университет им. К. Жубанова, Активобе, Казахстан E-mail: aidana_6143@mail.ru
Ташимова Анар Кошимовна	магистр, старший преподаватель кафедры информатики и информационных технологий, Активобе региональный университет им. К. Жубанова, Активобе, Казахстан, E-mail: anar_6868@mail.ru
Капарова Лэззат Есенгалиевна	магистр, старший преподаватель кафедры информатики и информационных технологий, Активобе региональный университет им. К. Жубанова, Активобе, Казахстан, E-mail: lazzat_09@mail.ru
Умирзакова Бану Габиденовна	магистр, старший преподаватель кафедры информатики и информационных технологий, Активобе региональный университет им. К. Жубанова, Активобе, Казахстан, E-mail: .banoka_98@mail.ru

Information about the authors

Adilzhan Kutymovich Kereev	PhD, Associate Professor, Department of Informatics and Information Technologies, K. Zhubanov Aktobe Regional University, Aktobe, Kazakhstan E-mail: .adiljan@mail.ru
Aidana Omarqyzy Zhaylybayeva	Master’s degree, Lecturer, Department of Informatics and Information Technologies, K. Zhubanov Aktobe Regional University, Aktobe, Kazakhstan E-mail: aidana_6143@mail.ru
Anar Koshimovna Tashimova	Master’s degree, Senior Lecturer, Department of Informatics and Information Technologies, K. Zhubanov Aktobe Regional University, Aktobe, Kazakhstan E-mail: anar_6868@mail.ru
Lezzat Yesengalievna Kaparova	Master’s degree, Senior Lecturer, Department of Informatics and Information Technologies, K. Zhubanov Aktobe Regional University, Aktobe, Kazakhstan E-mail: lazzat_09@mail.ru
Banu Gabidenovna Umirzakova	Master’s degree, Senior Lecturer, Department of Informatics and Information Technologies, K. Zhubanov Aktobe Regional University, Aktobe, Kazakhstan E-mail: .banoka_98@mail.ru