



**О. Ю. Шамуратов, Н. Б. Шаховська**

*Національний університет "Львівська політехніка", м. Львів, Україна*

## АЛГОРИТМИ КОНТУРНОГО АНАЛІЗУ ЗОБРАЖЕНЬ

Порівняно основні властивості нелінійних методів контурного аналізу за вимогами у використанні обмеженості обчислювальних ресурсів апаратних засобів. Для цього було проаналізовано загальні поняття про нелінійні алгоритми пошуку контурів на зображеннях та було обрано 4 методи для проведення подальшого аналізу. З них 3 методи з ковзною маскою 3×3, а саме: оператор Прюїтта; оператор Собеля; оператор Кірша та метод з маскою 5×5 – оператор Лапласа. Усі методи протестовано на однаковому зображенні роздільною здатністю 6016×4000 пікселів. Під час порівняння методів особливу увагу приділено використанню обчислювальних ресурсів апаратних засобів, тобто оперативної пам'яті та завантаженості процесора, а також швидкості виконання алгоритму. Оператор Кірша є найбільш ресурсно-витратним методом серед розглянутих, але при цьому його перевага в дуже чутливій масці, що виділяє цей метод для використання навіть за найнижчої загальної яскравості зображення. Оптимальним методом щодо обчислювальної потужності та знаходження контурів є оператор Собеля, це зумовлено використанням маски з коефіцієнтами тільки для середніх значень. Окремо можна виділити оператор Лапласа. Цей метод виконано швидше і має він меншу обчислювальну вартість. При цьому дає не набагато гірший за інші методи результат. Цей метод добре використовувати, якщо обчислювальні потужності є не високими.

**Ключові слова:** методи контурного аналізу; оператор Прюїтта; оператор Собеля; оператор Кірша; оператор Лапласа.

**Вступ.** Контурний аналіз дає змогу описувати, зберігати порівнювати і знаходити об'єкти, що перебувають у формі зовнішніх контурів – контурів. Передбачають, що контур містить необхідну інформацію про форму об'єкта. Внутрішні точки об'єкта до уваги не беруть. Це обмежує область застосування алгоритмів контурного аналізу, але розгляд тільки контурів дає змогу перейти від двомірного простору образу до простору контурів і, отже, зменшити обчислювальну й алгоритмічну складності. Контурний аналіз допомагає ефективно вирішувати головні завдання розпізнавання шаблонів – перенесення, поворот і масштабування зображення об'єкта. Методи контурного аналізу інваріантні щодо таких перетворень.

Методи контурного аналізу актуальні й на сучасному етапі у сфері розпізнавання об'єктів. Наприклад у програмах розпізнавання тексту, такі програми зараз значно поширені навіть у повсякденному житті. Окрім розпізнавання тексту, контурний аналіз використовують у системах розпізнавання об'єктів потокового відео. Такі системи на сучасному етапі використовують у сфері безпеки, для автопілотів автомобілів тощо. Під час використання методів контурного аналізу можна підраховувати кількість потрібних об'єктів на зображенні, такі системи використовуються на складі підприємств для проведення аудитів товарів.

Методи контурного аналізу часто використовують в Інтернеті 4.0. для моделювання та візуалізації етапів ви-

робничого процесу. Окрім розпізнавання тексту на зображенні для оцінювання відповідності об'єкта певним вимогам, існує можливість розпізнавання символів за допомогою Web-камери.

У цій роботі розглянемо основні нелінійні методи контурного аналізу, порівняємо їхні основні властивості та ресурсно-витрати на виконання.

**Аналіз літературних джерел.** Усі методи контурного аналізу базуються на одній із властивостей яскравості – розривності. Для пошуку цієї розривності використовують так звану ковзну маску (Bastian & Schiele, 2003; Hao et al., 2017), що є квадратною матрицею коефіцієнтів, відповідну до групи пікселів вхідного зображення. Цю матрицю використовують для виконання просторової фільтрації зображення (Yang et al., 2016; Peleshko et al., 2016).

Просторова фільтрація виконується звичайним переміщенням маски по зображенню та обрахуванням у кожній точці градієнта рівня яскравості. Однотонні простори зображення будуть мати низький градієнт рівня яскравості, у вихідному зображенні такі простори зображення темнішають, а у випадку високого рівня градієнта на вихідному зображенні ці простори будуть мати яскравіші лінії. Перед обрахунком градієнта потрібно вирахувати відгук  $R$  фільтрації в точці  $(x, y)$  (Duda, & Hart, 1973):

$$R = w(-1, -1)f(x-1, y-1) + w(-1, 0)f(x-1, y) + \dots + w(0, 0)f(x, y) + \dots + w(1, 0)f(x+1, y) + w(1, 1)f(x+1, y+1),$$

### Інформація про авторів:

**Шамуратов Олексій Юрійович**, аспірант, кафедра систем штучного інтелекту. Email: jonburned@gmail.com

**Шаховська Наталія Богданівна**, д-р техн. наук, професор, завідувач кафедри систем штучного інтелекту.

Email: nataliya.b.shakhovska@lpnu.ua; <https://orcid.org/0000-0002-6875-8534>

**Цитування за ДСТУ:** Шамуратов О. Ю., Шаховська Н. Б. Алгоритми контурного аналізу зображень. Науковий вісник НЛТУ України. 2019, т. 29, № 6. С. 123–127.

**Цитування APA:** Shamuratov, O. Yu., & Shakhovska, N. B. (2019). Algorithms of image contour analysis. *Scientific Bulletin of UNFU*, 29(6), 123–127. <https://doi.org/10.15421/40290624>

тобто це сума добутку коефіцієнтів маски на значення пікселів, до яких застосовується маска. Для визначення градієнта яскравості використовуються дискретні аналоги похідних першого і другого порядку (Canny, 1983). Перша похідна  $f(x)$  визначається як різниця значень сусідніх пікселів, а друга похідна – як різниця значень сусідніх значень першої похідної (Canny, 1983)

$$\frac{\delta f}{\delta x} = f(x+1) - f(x), \quad \frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x).$$

Піксель зображення звичайно має дві координати, а отже, у випадку двох змінних  $(x, y)$  потрібно просто обчислити часткові похідні за двома просторовими осями. Обчислення першої похідної зображення виконують на дискретних наближеннях двовимірного градієнта. Градієнт зображення – це вектор (Dipalee & Choubey, 2015)

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix}.$$

У методах контурного аналізу важливим є модуль вектора градієнта функції:

$$|\nabla f| = \sqrt{G_x^2 + G_y^2}.$$

Напрямок вектора градієнта теж є основною характеристикою під час знаходження контурів:

$$a(x, y) = \arctg\left(\frac{G_y}{G_x}\right).$$

Напрямок контура в точці  $(x, y)$  буде перпендикулярний напрямку вектора градієнта (David & Hildreth, 1980).

Метою роботи є аналіз алгоритмів визначення контурів зображень та порівняння їх ефективності.

### Викладення основного матеріалу

**1. Оператор Прюїтта** використовує маску  $3 \times 3$  для просторової фільтрації.

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

У цих функціях різниця між сумами по верхній і нижній строками області  $3 \times 3$  є наближеними значеннями похідної по осі  $x$ , а різниця між сумами по першому та останньому стовпцям – похідна по осі  $y$ . Реалізуються ці функції за допомогою маски фільтрації по осях  $x$  та  $y$  (Prewitt, 1970)

$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}; \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}.$$

Виконаємо оператор Прюїтта на тестовому зображенні роздільної здатності  $6016 \times 4000$  px (рис. 1 і 2).

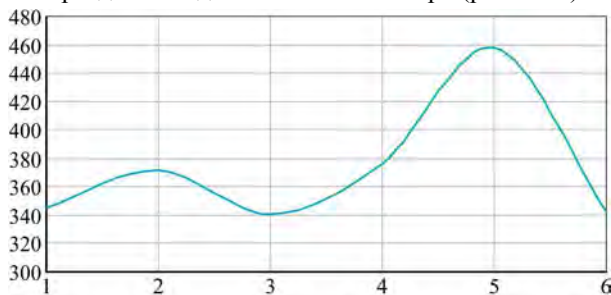


Рис. 1. Графік завантаженості оперативної пам'яті (МБ) під час виконання оператора Прюїтта

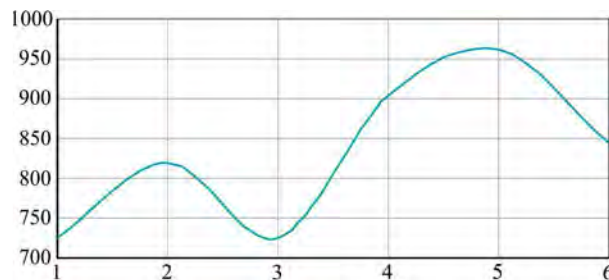


Рис. 2. Графік завантаженості процесора (Гц) під час виконання оператора Прюїтта

Результати роботи оператора Прюїтта (рис. 3).



Рис. 3. Оригінальне зображення (зліва). Результат виконання оператора Прюїтта (справа)

**2. Оператор Собеля** також використовує маску  $3 \times 3$  для просторової фільтрації.

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}; \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}.$$

Однак його особливістю є використання вагового коефіцієнта для значень середніх елементів (Sobel, & Feldman, 1973):

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

Для перевірки роботи оператора Собеля використаємо те саме тестове зображення (рис. 4 і 5).

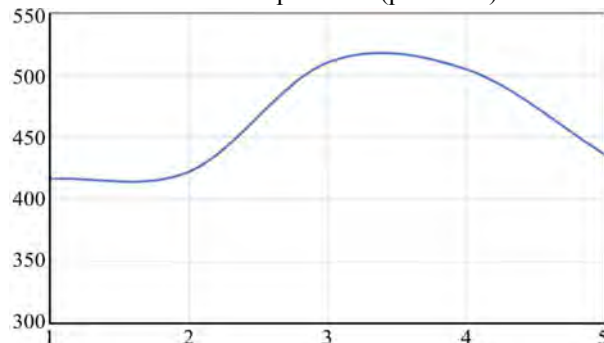


Рис. 4. Графік завантаженості оперативної пам'яті (МБ) під час виконання оператора Собеля

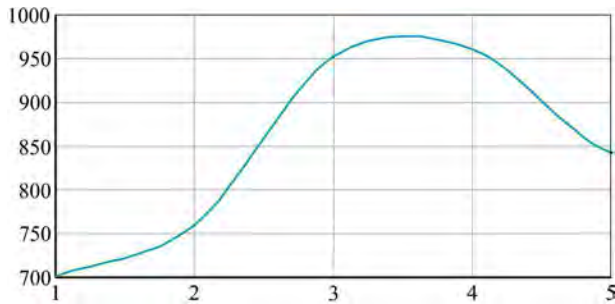


Рис. 5. Графік завантаженості процесора (Гц) під час виконання оператора Собеля

Результати роботи оператора Собеля (рис. 6).



Рис. 6. Оригінальне зображення (зліва). Результат виконання оператора Собеля (справа)

Як видно з результатів, оператор Собеля більш чутливий до перепадів яскравості на зображенні.

**3. Оператор Кірша** використовує коефіцієнти для всіх значень, окрім середнього. Маска оператор Кірша (Kirsch, 1971) (рис. 7 і 8).

$$G_x = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}; G_y = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & 3 & -3 \end{bmatrix}$$

Як видно з результатів виконання оператора Кірша, вагові коефіцієнти дають дуже сильну чутливість до перепадів яскравості.

Результати роботи оператора Кірша наведено на рис. 9.

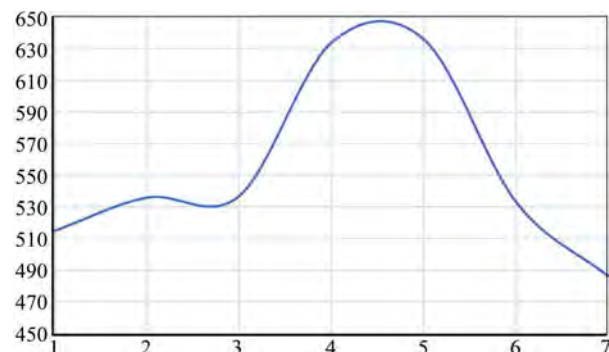


Рис. 7. Графік завантаженості оперативної пам'яті (МБ) під час виконання оператора Кірша

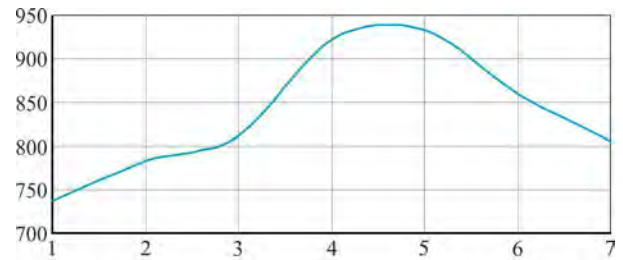


Рис. 8. Графік завантаженості процесора (Гц) під час виконання оператора Кірша



Рис. 9. Оригінальне зображення (зліва). Результат виконання оператора Кірша (справа)

**4. Оператор Лапласа (5×5).** Оператор Лапласа, що використовують для виділення контурів, це розширення векторного оператора Лапласа. Цей оператор має однакову маску для просторової фільтрації як для осі x, так і для осі y.

$$G_{(x,y)} = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

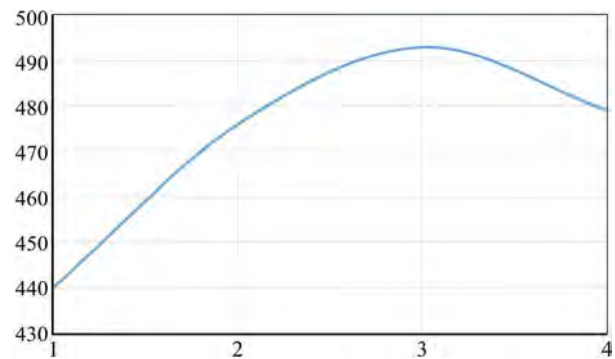


Рис. 10. Графік завантаженості оперативної пам'яті (МБ) під час виконання оператора Лапласа

Таку маску можна інтерпретувати як суму різниць центрального елемента з кожним із 24 інших елементів.

Таким чином, однаково застосовують можливі перепади яскравості у всіх напрямках (Moon, & Spencer, 1953; Kapustiy, Rusyn, & Tajanov, 2005) (рис. 10–12).

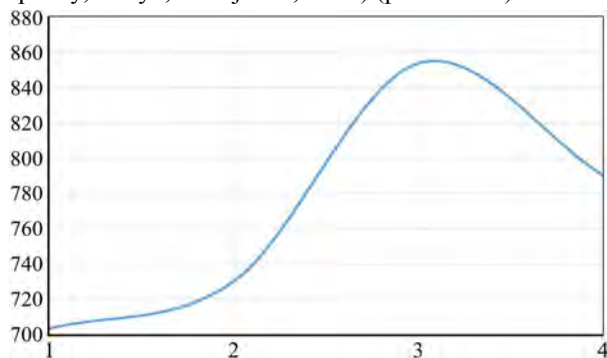


Рис. 11. Графік завантаженості процесора (Гц) під час виконання оператора Лапласа



Рис. 12. Оригінальне зображення (зліва). Результат виконання оператора (справа)

**Висновки.** На основі отриманих результатів під час аналізу нелінійних методів контурного аналізу можна зробити подальші висновки. Оператор Прюїтта найменш ресурсно-витратний, але при цьому більшість контурів на зображенні просто не виділяються, але дуже добре підійде для зображень з високою загальною яскравістю, тому що в такому випадку менша імовірність прояву контуру, якого не існує на вхідному зображенні, це зумовлено менш чутливою маскою. Оператор Кірша є найбільш ресурсно-витратним методом серед розглянутих, але при цьому його перевага в дуже чутливій масці, що виділяє цей метод для використання навіть за найнижчої загальної яскравості зображення. Оп-

тимальним методом щодо ресурсно-витратам та знаходження контурів є оператор Собеля. Це зумовлено використанням маски з коефіцієнтами тільки для середніх значень (Furgala, Mochulsky, & Rusyn, 2018; Kosar, & Shakhovska, 2018). Окремо можна виділити оператора Лапласа. Цей метод виконується швидше і має меншу обчислювальну вартість. При цьому дає не набагато гірший за інші методи результат. Цей метод добре використовувати, якщо обчислювальні потужності є не високими.

### Перелік використаних джерел

- Bastian, L., & Schiele, B. (2003). Analyzing appearance and contour based methods for object categorization. *Computer Society Conference on Computer Vision and Pattern Recognition*. (Vol. 2). Proceedings.
- Canny, J. F. (1983). *Finding edges and lines in images*. Masters thesis, MIT. AI Lab. TR, 720 p.
- David, M., & Hildreth, E. (1980). Theory of edge detection. (Ser. B. Biological Sciences). *Proceedings of the Royal Society of London*, 207(1167), 187–217.
- Dipalee, G., & Choubey, S. (2015). Discrete wavelet transform for image processing. *International Journal of Emerging Technology and Advanced Engineering*, 4(3), 598–602.
- Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York: Wiley, 482 p.
- Furgala, Y., Mochulsky, Y., & Rusyn, B. (2018). Evolution of objects recognition efficiency on maps by various methods. *Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing DSMP*. (pp. 595–598).
- Hao, C., et al. (2017). DCAN: Deep contour-aware networks for object instance segmentation from histology images. *Medical image analysis*, 36, 135–146.
- Kapustiy, B., Rusyn, B. P., & Tajanov, V. A. (2005). A new Approach to Determination of Correct Recognition Probability of the Set Objects. *Upravlyayushchie Sistemy i Mashiny*, 2, 8–12.
- Kirsch, R. (1971). Computer determination of the constituent structure of biological images. *Computers and Biomedical Research*, 4, 315–328.
- Kosar, O., & Shakhovska, N. (2018). An Overview of Denoising Methods for Different Types of Noises Present on Graphic Images. *Conference on Computer Science and Information Technologies*. (pp. 38–47). Springer, Cham.
- Moon, P., & Spencer, D. E. (1953). The Meaning of the Vector Laplacian. *J. Franklin Inst.*, 551–558.
- Peleshko, D., et al. (2016). Design and implementation of visitors queue density analysis and registration method for retail videosurveillance purposes. *First International Conference on Data Stream Mining & Processing*.
- Prewitt, J. M-S. (1970). Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1), 15–19.
- Sobel, I., & Feldman, G. (1973). A 3x3 Isotropic Gradient Operator for Image Processing. *Pattern Classification and Scene Analysis*, 271–272.
- Yang, J., Price, B., Cohen, S., Lee, H., & Yang, M. H. (2016). Object contour detection with a fully convolutional encoder-decoder network. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 193–202).

**O. Yu. Shamuratov, N. B. Shakhovska**

*Lviv Polytechnic National University, Lviv, Ukraine*

### ALGORITHMS OF IMAGE COUTOUR ANALYSIS

The comparison of the basic properties of nonlinear methods of contour analysis is presented in the article. Analysis was performed according to the requirements for the use of limited computing resources of hardware. To do this, the general concepts of nonlinear algorithms for searching contours in images were analyzed and four methods for further analysis were selected. Of these, three methods are with a sliding mask of 3x3 matrix, namely the Prewitt operator, Sobel operator and the Kirsch operator. Another method

with  $5 \times 5$  matrix mask is the Laplace operator. The Prewitt operator uses a  $3 \times 3$  mask for spatial filtering. The Sobel operator also uses a  $3 \times 3$  mask for spatial filtering, however, its feature is using a weighting factor for mean elements. And the Kirsch operator uses coefficients for all values except the mean. The Laplace operator used to select contours is an extension of the Laplace vector operator. This operator has the same mask for spatial filtration, both for the  $x$ -axis and for the  $y$ -axis. Such a mask can be interpreted as the sum of the differences between the central element and each of the twenty-four other elements. Thus, the possible variations of brightness in all directions are applied equally. All methods were tested on the same image with a resolution of  $6016 \times 4000$  pixels. When comparing algorithms, the main attention was paid in case of use methods in minimal hardware computing resources, i.e., the random access memory and central processing unit load, as well as the performance rate of the algorithm. The Kirsch operator is the most resource-cost method among the ones considered, although this algorithm has advantage in a very sensitive mask, which distinguishes this method for use even with images with a very low overall brightness. The optimal method in case of computational cost and number of contours found is the Sobel operator. This is achieved by the use of a mask with coefficients for averages only. Separately, the Laplace operator can be selected. This method is faster and has a lower computing value. At the same time, the results are not much worse than other methods. This algorithm is well used if computing power is not high.

**Keywords:** methods of contour analysis; Prewitt operator; Sobel operator; Kirsch operator; Laplace operator.