# Hybrid quadratic diagonal algorithm for thinning contour lines

*Narzullo* Mamatov[1*], *Malika* Jalelova[1], *Vohid* Fayziev[1], *Abdurashid* Samijonov[1], and *Boymirzo* Samijonov[2]

[1]Tashkent Institute of Irrigation and Agricultural Mechanization Engineers, National Research University, Tashkent, Uzbekistan
[2]Sejong University, South Korea, Seoul, Korea

**Abstract.** One of the main issues of image analysis is the separation of contour lines. Nowadays, many effective methods for dividing contour lines have been developed. In solving some practical problems, the results obtained by contour separation methods will not be enough, that is, operations such as thinning, filling, and smoothing of contour lines are required. In this case, the development of an efficient contour thinning algorithm used for accurate separation of the shape of the object is an urgent issue. Contour thinning algorithms can reduce the amount of data to be processed and increase processing speed. Based on the literature analysis, the Zhang-Suen algorithm can be recognized as the most efficient among the contour thinning algorithms due to the efficiency and speed of preserving the shape of the objects in the images. However, this algorithm fails to thin some contour lines. Therefore, in this work, an improved quadratic diagonal algorithm based on the strengths of the Zhang-Suen algorithm is proposed. Also, the proposed algorithm is compared with the existing algorithms regarding error and time criteria in contour detection. By conducting experimental studies, the Hybrid Quadratic Diagonal Algorithm showed the smallest error compared to the algorithms obtained for testing in the experiment.

## 1 Introduction

In recent years, the task of object shape recognition in images has been recognized as a crucial step in the development of many modern applications. Examples include medical, security, and applications including biometric recognition, disability support systems, and navigation for wayfinding. Although there are many methods and algorithms for object recognition in images [1-3], each has its advantages and disadvantages. Most existing methods that solve the recognition problem are effective only for certain types of objects, for example, a person's face, simple geometric shapes, handwritten or printed characters, and road signs [4]. However, there is not always an effective method or algorithm for object shape recognition for images under different conditions. Nevertheless, the contour-based description can be considered a more successful model of the object's shape in the image.

* Corresponding author: dilnoz134@rambler.ru

Contour lines contain information about the shape, structure, and composition of an object in an image, making them an important tool for image analysis and recognition. Therefore, researchers in the field of image processing have paid great attention to the development of contour detection methods and algorithms. However, as a result of applying existing contour detection algorithms to images, the lines describing the shape of the object are often more than one pixel thick. Thick contour lines make it difficult to separate the shape of the object from the image and cause an increase in image processing time. In addition, the thickness of the lines should be one pixel to highlight special points as a symbol during recognition [5]. In modern applications, to ensure high accuracy, speed, and reliability in object recognition in images, the problem of thinning contour lines is considered an urgent task.

Thinning the contour lines allows to collection of the most important information useful for further analysis and recognition process. It is applied to a binary image and produces another binary image as an output image [6]. Algorithms for thinning contour lines are widely used in various fields. In particular, it is used for the accurate separation of tumors and other pathologies in medical images for diagnosis, object detection and control in robotics, face recognition in automatic video surveillance applications, as well as for document image analysis, fingerprint identification, biometric authentication using retinal images, signature authentication, etc. used in many applications [7]. In such applications, since it is not enough to represent the contours of the object in the image, the contour lines are thinned to ensure that it is clear and reliable.

Some of the advantages of thinning contour lines in images are:

- outline reduces line width to one pixel;
- creates a thin line image of the fingerprint image;
- provides the connection between object pixels;
- enables faster processing of data on the computer and reduction of processing time;
- can save computer memory, because it generates only important data, and therefore the image size is smaller;

Although various contour thinning algorithms have been developed to achieve the mentioned advantages, they have some disadvantages. Some contour thinning algorithms have disadvantages such as not creating a single-pixel line or losing the contour line at all, spending a lot of time for processing [8-10]. To solve these problems, it is necessary to improve or develop existing contour thinning algorithms. A new or improved algorithm is required to be efficient in processing images of various natures while preserving the object details and contour structure in the image and to work quickly with large amounts of data. Therefore, a new contour thinning algorithm is proposed in this research paper, which is compared with other contour thinning algorithms and tested in computational experiments. Contour thinning algorithms used in computational experiments were selected based on the following literature review and verified in experiments.

## 2 Related work

Contour thinning algorithms are algorithms that require a binary image as input. They can be divided into two groups iterative and non-iterative algorithms [7]. Iterative algorithms thin the contour line based on the inspection of each pixel in the binary image. Non-iterative algorithms work by calculating distances without checking pixels one by one. Although such algorithms are efficient in terms of computational time, they may not meet the demand in terms of quality. That is why most researchers prefer to work with iterative algorithms. For example, Zhang-Suen, Guo-Hall, and Hilditch algorithms can be cited as examples. A comparison of four well-known contour thinning algorithms was made in [7], where it was proved that the Zhang-Suen algorithm gives the minimum computation time. The Zhang-

Suen and Guo-Hall algorithms were studied in [11], and it was found that the Zhang-Suen algorithm performed better.

It is mentioned in several literature that the existing algorithms of contour thinning have some shortcomings. In particular, the Zhang-Suen algorithm works on a pixel with eight neighbors, but it does not work in the analysis of the edge pixels of the image [12]. It was reported in [13] that the K3M contour thinning algorithm gave different thinning results when rotating the image to different angles. Also, it was pointed out in [8] that although the morphological algorithm is fast in terms of contour thinning, it is directly dependent on the result of image preprocessing algorithms [14-16].

The fact that the traditional Zhang-Suen algorithm does not check conditions for diagonal elements is also recognized as a drawback of this algorithm. As a solution to this problem, a new improved algorithm known as "bidiagonal", which includes the verification of diagonal pixels in the conditions of the Zhang-Suen algorithm, was proposed by the authors of [9], and it was proved that this algorithm is superior to the Zhang-Suen algorithm in terms of pixel comparison.

As a result of the literature analysis, it can be said that the Zhang-Suen algorithm is a widely used algorithm among the contour thinning algorithms, and it has been taken as a basis for the development of other new algorithms. However, since contour thinning algorithms have some limitations, the task of improving them remains relevant. Below are detailed descriptions of the existing algorithms used in the work and the proposed algorithm.

# 3 Methodology

## 3.1 Morphological operator-based algorithm

This algorithm processes the image based on operations such as dilation, erosion, opening and closing:

Dilation. Suppose that $A$ - is a given image, and $B$ - is a set belonging to $Z^2$ as a structuring element. The extension of set $A$ with set $B$ is denoted by $A \oplus B$ and is defined as:

$$A \oplus B = \{a + b : a \in A, b \in B\} = \bigcup_{b \in B} A_b \tag{1}$$

Erosion is defined as:

$$A \Theta B = \{a : B_a \subseteq A\} = \bigcap_{b \in B} A_b \tag{2}$$

where erosion refers to the intersection of sets and dilation refers to their union.

The opening is defined as:

$$A_B = (A \Theta B) \oplus B \overset{\Delta}{=} A \circ B \tag{3}$$

Unfolding tries to smooth the contours of the object in the image.

The closing is defined as:

$$A^B = (A \oplus B) \Theta B \overset{\Delta}{=} A \bullet B \tag{4}$$

Closing also tries to smooth the contour lines, but unlike opening, it usually eliminates small holes and fills gaps in the contour.

## 3.2 Zhang-suen algorithm

This algorithm is one of the most common algorithms for thinning contour lines. It replaces the brightness of the pixels with the background color according to certain conditions to generate a thin contour line by analyzing the brightness of the pixels around each pixel. As a result, it allows to reduce the number of pixels needed to process the object in the image while preserving its shape and topology information.

The Zhang-Suen algorithm consists of the following two steps, based on which the pixels forming the thick contour line are deleted. That is when moving through the image with a $3\times3$ mask at each step, it is checked that each pixel belongs to the boundary of a certain connected area, and if the conditions of the check are fulfilled, the pixel is deleted:

Stage 1.

1. $2 \leq sum \leq 6$ (Algorithm_1);
2. $B = 1$ (Algorithm_2);
3. $p_2 * p_4 * p_6 = 0$;
4. $p_4 * p_6 * p_8 = 0$.

Applying step 1 to a binary image removes all pixels to the right and bottom of the boundary. Step 2 of the Zhang-Suen algorithm is used to remove the pixels on the top and left border:

Stage 2.

1. $2 \leq sum \leq 6$ (Algorithm_1);
2. $B = 1$ (Algorithm_2);
3. $p_2 * p_4 * p_8 = 0$;
4. $p_2 * p_6 * p_8 = 0$.

Algorithm_1 in the first and second steps is a non-zero resolution checking algorithm.

**Algorithm_1**

```
sum = 0
if t [i - 1] [j]:     sum += 1
if t [i - 1] [j + 1]:     sum += 1
if t [i - 1] [j - 1]:     sum += 1
if t [i] [j + 1]:     sum += 1
if t [i] [j - 1]:     sum += 1
if t [i + 1] [j]:     sum += 1
if t [i + 1] [j + 1]:     sum += 1
if t [i + 1] [j - 1]:     sum += 1
```

The task of Algorithm_2 is to check the number of transitions.

**Algorithm_2**

```
p2 = t [i] [j - 1]
p3 = t [i + 1] [j - 1]
p4 = t [i + 1] [j]
p5 = t [i + 1] [j + 1]
p6 = t [i] [j + 1]
p7 = t [i - 1] [j + 1]
p8 = t [i - 1] [j]
p9 = t [i - 1] [j - 1]
B = int ((not p2 and p3)) + int ((not p3 and p4)) + int((not p4 and p5)) + \
    int ((not p5 and p6)) + int ((not p6 and p7)) + int((not p7 and p8)) + \
    int ((not p8 and p9)) + int ((not p9 and p2))
```

The Zhang-Suen algorithm examines only three out of eight neighboring pixels to delete the pixel $p_1$ (Figure 1).

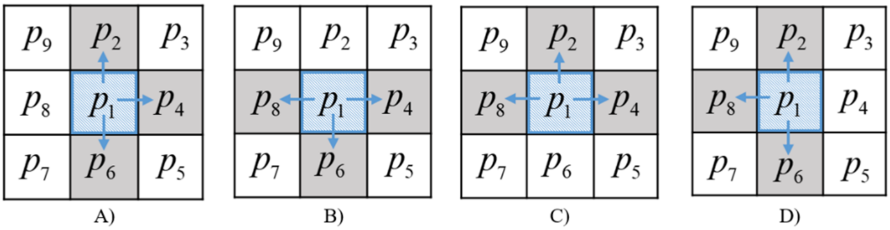


**Fig. 1.** Checked cases in the Zhang-Suen algorithm: A) $p_2p_4p_6$, B) $p_4p_6p_8$, C) $p_2p_4p_8$, D) $p_2p_6p_8$.

### 3.3 Semicircular algorithm

Based on the idea of the Zhang-Suen algorithm, a half-circle algorithm was developed based on checking the conditions for deleting a pixel, not only three of the eight neighboring pixels but five neighbors of the half-circle (Figure 2). To implement this algorithm, the following conditions are checked:

Stage 1.

1. $2 \leq sum \leq 7$;
2. $B = 1$ ;
3. $p_2 * p_9 * p_8 * p_7 * p_6 = 0$;
4. $p_8 * p_7 * p_6 * p_5 * p_4 = 0$.

Stage 2.

1. $2 \leq sum \leq 7$;
2. $B = 1$;
3. $p_6 * p_5 * p_4 * p_3 * p_2 = 0$;
4. $p_4 * p_3 * p_2 * p_9 * p_8 = 0$.

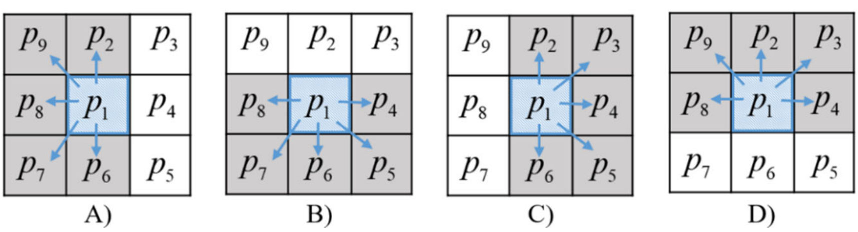


**Fig. 2.** Checked conditions in the semicircular algorithm: A) $p_2p_9p_8p_7p_6$, B) $p_8p_7p_6p_5p_4$, C) $p_6p_5p_4p_3p_2$, D) $p_4p_3p_2p_9p_8$.

### 3.4 Bidiagonal algorithm

A detailed description of this algorithm is given in [3], where the conditions of the Zhang-Suen algorithm include the checking of diagonal pixels (Figure 3).

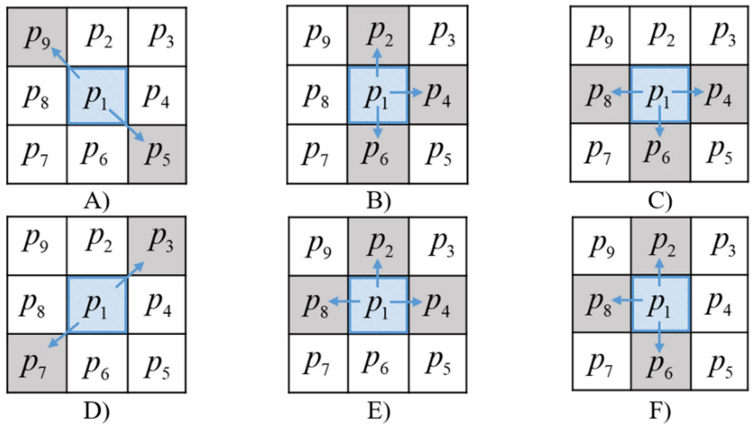


**Fig. 3.** Cases to be checked in the bidiagonal algorithm: A) $p_9p_5$, B) $p_2p_4p_6$, C) $p_4p_6p_8$, D) $p_3p_7$, E) $p_2p_4p_8$, F) $p_2p_6p_8$.

### 3.5 Semicircular diagonal algorithm

This algorithm also includes checking pixels located diagonally to the conditions of the semicircular algorithm check above. Therefore, the cases in Figure 4 are taken to verify the first and second stages of the so-called "semicircular diagonal" and semicircular algorithm.

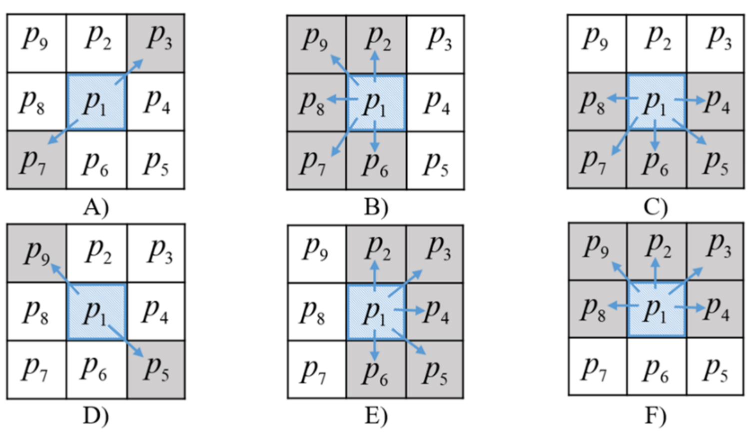


**Fig. 4.** Cases to be checked in the semicircular diagonal algorithm: A) $p_3p_7$, B)$p_2p_9p_8p_7p_6$, C) $p_8p_7p_6p_5p_4$, D) $p_9p_5$, E) $p_6p_5p_4p_3p_2$, F) $p_4p_3p_2p_9p_8$.

### 3.6 Hybrid quadratic diagonal algorithm (HQDA)

Contour thinning can be improved by combining existing algorithms; on this basis, an HQDA is proposed in this paper. This algorithm is an algorithm that can cover all the conditions of the Zhang-Suen, semicircular, bidiagonal, and semicircular diagonal algorithms listed above. Because it is checked in the following two steps: diagonal, semicircular, northwest corner, and southeast corner pixels, that is:

Stage 1.

1. $2 \leq sum \leq 7$;
2. $B = 1$;

3. $p_9 * p_5 = 0;$
4. $p_2 * p_4 * p_6 = 0;$
5. $p_4 * p_6 * p_8 = 0;$
6. $p_2 * p_9 * p_8 * p_7 * p_6 = 0;$
7. $p_6 * p_5 * p_4 * p_3 * p_2 = 0.$

In the first stage, the following can be indicated as the cases to be checked:

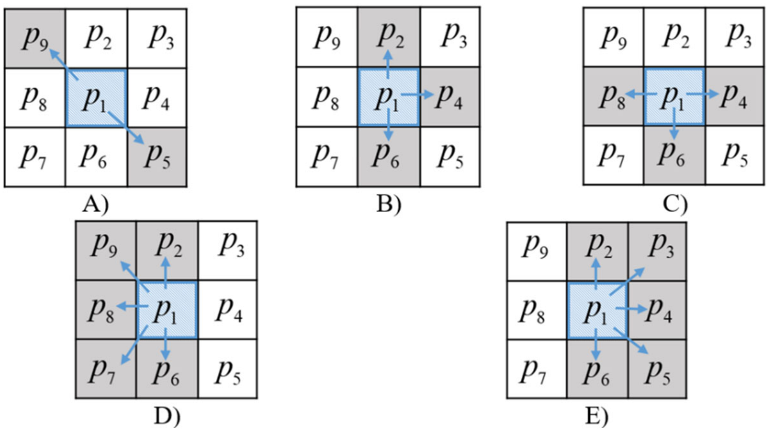

**Fig. 5.** Cases to be checked at the first step in the HQDA: A) $p_9p_5$, B) $p_2p_4p_6$, C) $p_4p_6p_8$ , D) $p_2p_9p_8p_7p_6$, E) $p_6p_5p_4p_3p_2$.

Stage 2.

1. $2 \leq sum \leq 7;$
2. $B = 1;$
3. $p_3 * p_7 = 0;$
4. $p_2 * p_4 * p_8 = 0;$
5. $p_2 * p_6 * p_8 = 0$
6. $p_8 * p_7 * p_6 * p_5 * p_4 = 0;$
7. $p_4 * p_3 * p_2 * p_9 * p_8 = 0$

In the second stage, the following can be indicated as checked cases:

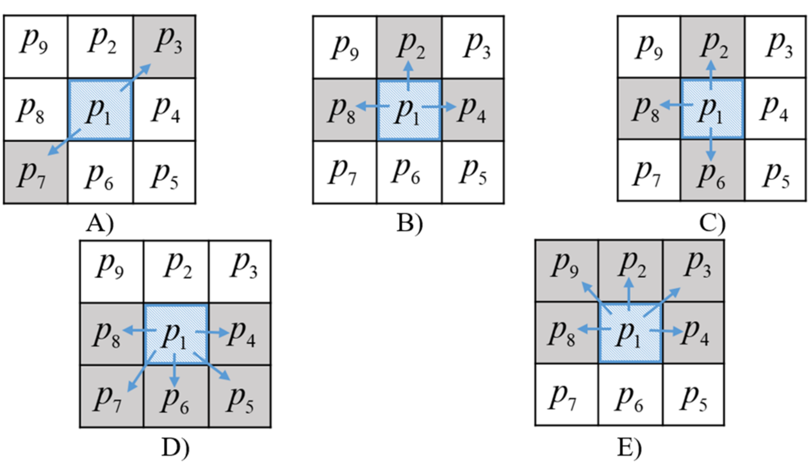

**Fig. 6.** Cases checked at the second stage in the HQDA: A) $p_3p_7$, B) $p_2p_4p_8$, C) $p_2p_6p_8$, D) $p_8p_7p_6p_5p_4$, E) $p_4p_3p_2p_9p_8$.

The general working principle of the HQDA is similar to a square shape and because it also takes into account the pixels on the diagonal, this algorithm is named "Hybrid Quadratic Diagonal Algorithm ".

"Contour detection error" was taken as a criterion to evaluate the presented and proposed contour thinning algorithms. This criterion was introduced by Ma and Staunton, which determines the error in contour detection by comparing the binary image determined by the contour thinning algorithm with a single-pixel reference image:

$$\omega = 1 - \frac{|G \cap E|}{|G|} \tag{5}$$

where, $G-$ is the single-pixel reference image, and $E-$ is the image obtained by applying a contour thinning algorithm.

(5) indicates that the smaller the value of the criterion, the more efficient the contour thinning algorithm is. Based on this, when comparing algorithms, the algorithm that achieves the smallest value according to the $\omega$ criterion is the most effective.

## 4 Results

56 images from the BSDS500 database with one-pixel ground truth images extracted by an expert were taken for the sample. First, a binary image is generated by applying contour extraction algorithms to the original images. In this case, the famous Sobel filter was used for the detailed separation of object contours, which is a good separator of object outer boundary lines. Morphological, Zhang-Suen, semicircular, bidiagonal, semicircular diagonal, and HQDA of contour thinning were applied to the formed thick linear contour images (Figure 7 - Figure10).

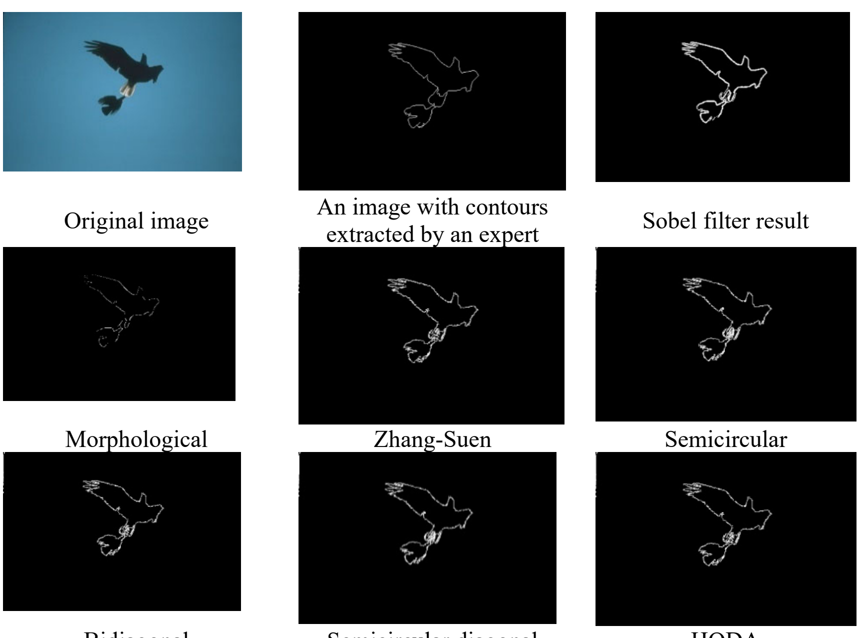


**Fig. 7.** Examples of application of contour thinning algorithms to image base (eagle.jpg).

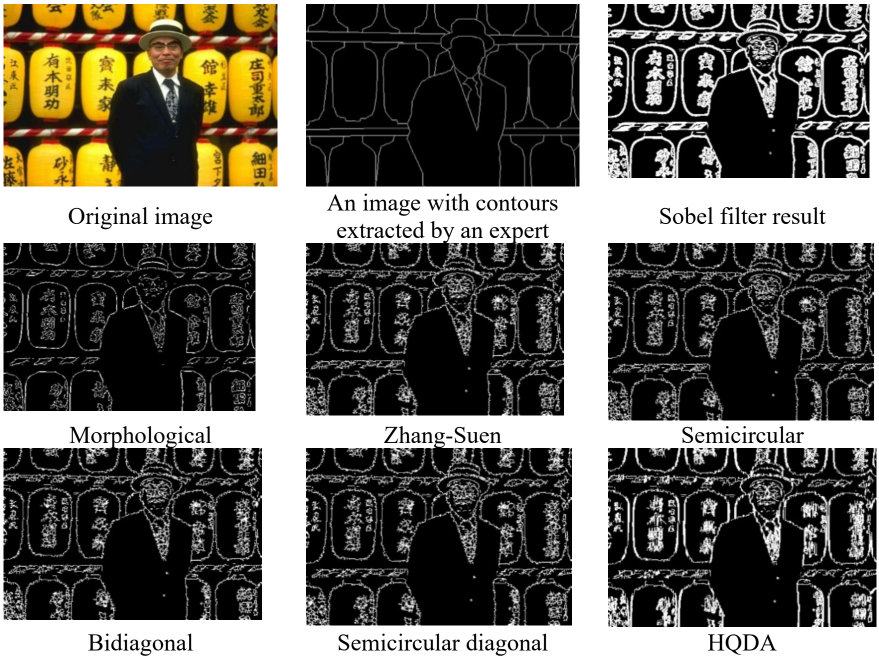


**Fig. 8.** Examples of application of contour thinning algorithms to image base (human.jpg).

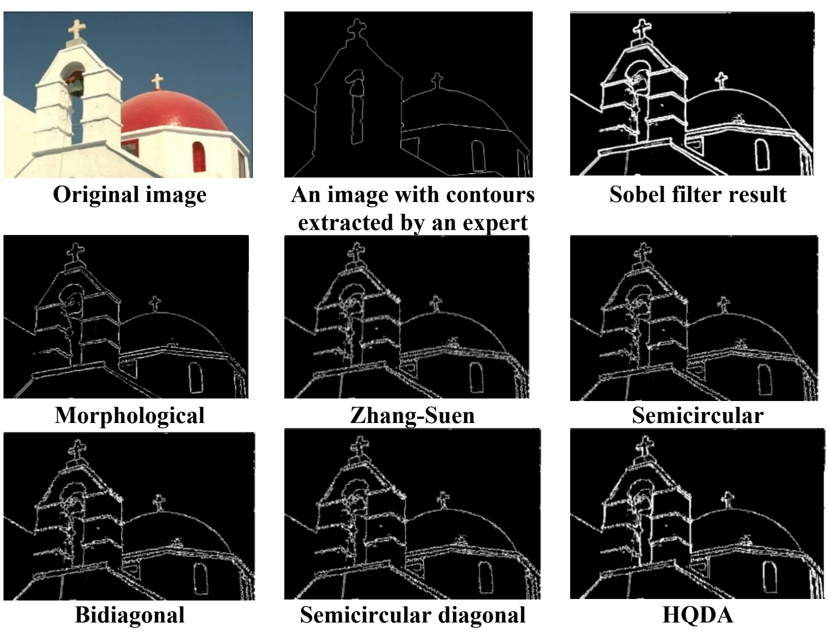


**Fig. 9.** Examples of application of contour thinning algorithms to image base (church.jpg).

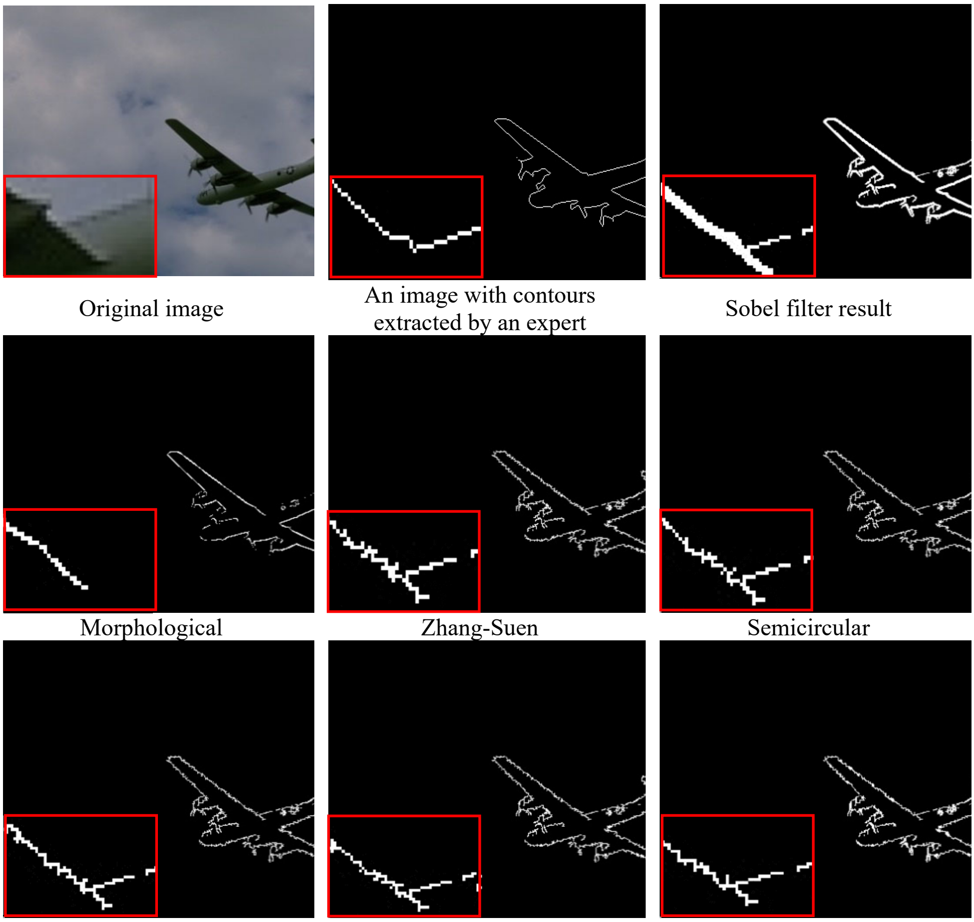


**Fig. 10.** Examples of application of contour thinning algorithms to image base (plane.jpg).

Criterion (5) was used in the mutual comparison of contour thinning algorithms presented above, and the results of evaluation by this criterion and time are presented in the table below.

**Table 1.** Results of comparison of contour thinning algorithms (5) and time criterion.

| **Evaluation indicator** | **Contour thinning algorithms** | | | | | |
|---|---|---|---|---|---|---|
| | **Morphological** | **Zhang-Suen** | **Semicircular** | **Bidiagonal** | **Semicircular diagonal** | **HQDA** |
| **eagle.jpg** | | | | | | |
| $\omega$ | 0.564 | 0.335 | 0.407 | 0.319 | 0.346 | <u>**0.284**</u> |
| $t$ (sec) | 0.21 | 3.38 | 2.99 | 2.17 | 4.98 | 5.94 |
| **human.jpg** | | | | | | |
| $\omega$ | 0.616 | 0.467 | 0.502 | 0.472 | 0.504 | <u>**0.214**</u> |
| $t$ (sec) | 0.41 | 30.59 | 31.82 | 49.38 | 49.13 | 68.08 |
| **church.jpg** | | | | | | |
| $\omega$ | 0.548 | 0.446 | 0.461 | 0.426 | 0.451 | <u>**0.147**</u> |
| $t$ (sec) | 0.13 | 11.46 | 14.23 | 19.09 | 16.49 | 44.80 |
| **plane.jpg** | | | | | | |
| $\omega$ | 0.504 | 0.345 | 0.363 | 0.343 | 0.359 | <u>**0.285**</u> |

| | | | | | | |
|---|---|---|---|---|---|---|
| $t$ (sec) | 0.18 | 4.56 | 3.51 | 5.80 | 3.98 | 6.63 |
| **red_bird.jpg** | | | | | | |
| $\omega$ | 0.578 | 0.317 | 0.383 | 0.309 | 0.370 | **0.161** |
| $t$ (sec) | 0.28 | 5.60 | 6.49 | 7.55 | 8.53 | 12.82 |
| **car.jpg** | | | | | | |
| $\omega$ | 0.563 | 0.365 | 0.375 | 0.370 | 0.379 | **0.310** |
| $t$ (sec) | 0.35 | 28.11 | 28.88 | 35.33 | 36.65 | 32.10 |

From Table 1, the algorithm based on the morphological operator, although fast in terms of time, provided a very poor result in terms of accuracy. It turned out that the proposed HQDA is superior in accuracy to all the contour thinning algorithms presented in the work.

## 5 Conclusion

In this work, the problem of thinning the contour lines, which is important in recognizing objects in the image, was studied and its advantages were listed. Based on the literature analysis, it was found that there are some limitations in applying the existing contour thinning algorithms to images, and these cases highlighted the need to improve the contour thinning algorithms. Also, as a result of literature analysis, it was found that the Zhang-Suen algorithm is more popular than other thinning algorithms in terms of image processing speed and accuracy. Therefore, an improved algorithm based on this algorithm was developed in the work, and this algorithm was named "Hybrid Quadratic Diagonal Algorithm" because its general principle of operation is similar to a square shape and takes into account pixels on the diagonal.

A comparative analysis with existing contour thinning algorithms was performed to evaluate the effectiveness of the proposed quadratic algorithm. In the study, tests were conducted on different images for the experiment and evaluated based on a criterion such as contour detection error. As a result of conducting this research, the following general conclusions were formed:

- although the algorithm based on the morphological operator provides quick results, it showed a very low indicator in terms of accuracy;
- it was found that the semicircular algorithm and the improved semicircular diagonal algorithm based on it are less accurate than the Zhang-Suen algorithm;

The HQDA demonstrated high performance in terms of the contour detection error criterion compared to all the contour thinning algorithms presented in the work.

The implementation of the proposed algorithm is relatively simple and can be used in various image processing systems. This research contributes to some extent to the ongoing work on improving the efficiency of contour thinning algorithms and demonstrates the potential of hybrid approaches in solving image processing problems.

## References

1. J. Fan, T. Zhao, Z. Kuang, Y. Zheng, J. Zhang, J. Yu, J. Peng, IEEE Transactions on Image Processing **26**, 1923 (2017).
2. J. Tang, C. Deng, G.-B. Huang, IEEE Transactions on Neural Networks and Learning Systems **27**, 809 (2015).
3. O. Ronneberger, P. Fischer, T. Brox, Lecture Notes in Computer Science, **9351,** 234-241 (2015).
4. V. Hrytsyk, M. Medykovskyy, M. Nazarkevych, Symmetry **14**, 903 (2022).

5. D. A. Bushenko, and R. H. Sadykhov, Reports of the Belarusian State University of Informatics and Radioelectronics, **7(45),** 81-86 (2009).
6. L. O'Gorman, R. Kasturi, *Document Image Analysis* (Library of Congress, 1994).
7. D. Kocharyan, American Journal of Software Engineering and Applications **2**, 1 (2013).
8. R.P. Prakash, V.P. Binu, 2015 IEEE International Advance Computing Conference (IACC) 1026 (2015).
9. G. Matúš, V. Jozef. Comparison of thinning algorithms for vectorization of engineering drawings **94** (2016).
10. B. Coüasnon, A. Lemaitre, Handbook of Document Image Processing and Recognition (2014).
11. R. Luthra, G. Goyal, International Journal of Computer Applications **121**, 1 (2015).
12. M. Nazarkevych, S. Dmytruk, V. Hrytsyk, O. Vozna, A. Kuza, O. Shevchuk, Y. Voznyi, I. Maslanych, V. Sheketa, International Journal of Sensors Wireless Communications and Control **11**, 542 (2020).
13. M. Tabedzki, K. Saeed, A. Szczepański, International Journal of Applied Mathematics and Computer Science **26**, 439 (2016).
14. X. Qian, J. Wang, S. Guo, Q. Li, Medical Physics **40**, 021911 (2013).
15. L. Jiao, J. Zhao, IEEE Access **7**, 172231 (2019).
16. N.S. Mamatov, M.M. Jalelova, A.N. Samijonov, B.N. Samijonov, Journal of Physics Conference Series **2697**, 012013 (2024).