Journal of Computers Vol. 32 No. 6, 2021, pp. 83-97
doi:10.53106/199115992021123206007

A Novel Sub-Iterative Parallel Skeletonization Method

Jun Ma, Xun-Huan Ren*, Tsiviatkou Viktar Yurevich,
Valery Konstantinovich Kanapelka

Department of Inforcommunication Technologies, Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus
{majun, renxunhuan}@bsuir.by

Received 26 March 2021; Revised 14 July 2021; Accepted 15 August 2021

Abstract. Parallel thinning methods are digital skeletonization approaches that apply parallel strategies to accelerate the processing speeds of algorithms. Existing parallel thinning methods fail to produce clean and complete single-pixel-width skeletons, where clean means that a skeleton contains fewer unwanted branches caused by boundary noise and complete means that the skeleton should have the same topology as the original image. To over-come this problem, in this paper, a novel sub-iterative parallel thinning method is proposed based on the Zhang-Suen (ZS) method by altering the original partial conditions and adding several additional deletion templates and one restoration template to each sub-iteration. Three experiments are conducted to evaluate the performance of the proposed method. The simple pattern experiment shows that the skeleton resulting from the proposed method can maintain the complete original topology. The noise experiment shows that the proposed algorithm is insensitive to boundary noise. Thus, it can produce a relatively clean skeleton. The complicated image experiment shows that the proposed method has a higher thinning rate than other approaches and has application potential in natural images.

Keywords: skeleton, skeletonization, robust to boundary noise, sub-iterative parallel thinning

1 Introduction

The skeleton, or medial axis, is a widely used shape descriptor in many application areas, such as finger pattern classification systems and circuit board inspection systems. A skeleton contains the shape and topology information of the original object. The method that extracts the skeleton from a binary image is called skeletonization, whose foundation was first defined by Blum [1] through an analogy with grassfires. Skeletonization methods may be divided into three major categories [2] based on their principles and underlying object representations: continuous geometric approaches, the continuous curve propagation approach, and digital approaches, as shown in Fig. 1.

Continuous geometric approaches generate skeletons by concentrating on the properties of Blum's medial symmetry axis. The method based on the Voronoi diagram [3-4] is one popular approach in this category. How-ever, these methods require many vertices to generate a proper polygonal approximation of a shape. Therefore, Voronoi skeletonization tends to yield many unwanted skeletal branches [3] that contribute little information about the overall shape.

The continuous curve propagation approach [5] is modeled using partial differential equations, in which cer-tain singularities occur; these singularities are referred to as shocks. The flaw of this method is that the resulting skeleton may not be topologically connected [6].

Digital skeletonization approaches are referred to as thinning algorithms in some research. Such a method removes the boundary pixels from a digital grid under specific geometric and topologic rules, which is the most primitive yet most popular method. Thinning algorithms can be further divided into iterative algorithms and non-iterative algorithms [7]. Generally, an iterative thinning algorithm can produce a continuous skeleton by repetitively scanning the original image and deleting the boundary

* Corresponding Author

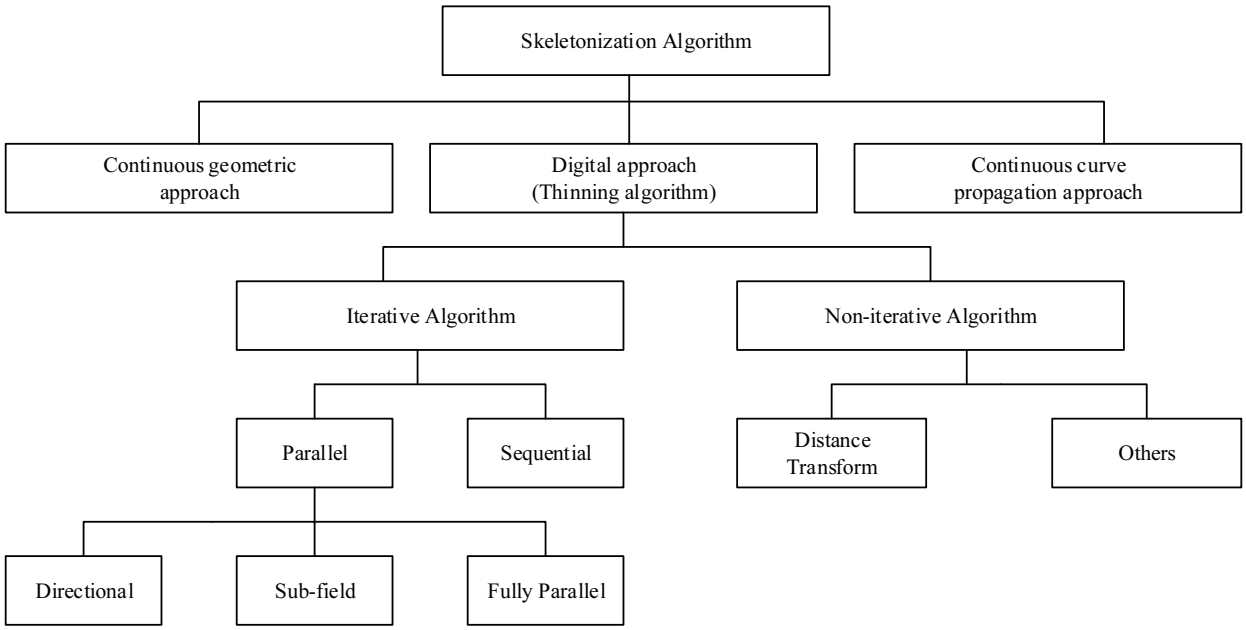


Fig. 1. Classification of skeletonization methods

pixels in each iteration, whereas non-iterative methods, such as distance transform (DT) methods [8], are characterized by computational efficiency since they do not require repetitive image scans. However, an apparent defect of DT methods is that they may be hard to parallelize. Iterative algorithms can be classified into parallel and sequential algorithms in terms of their computational efficiency [9-10]. By adopting different parallelization techniques, parallel algorithms can be partitioned into three groups: directional [7, 11-17], sub-field [18-19], and fully parallel thinning algorithms [20-26]. The directional approach breaks each thinning iteration into several sub-iterations according to a combination of directions (north, west, east, and south). The sub-field approach partitions the input image into several minor subfields according to some criteria, such as the parity of the pixels rather than edge orientation, so that pixels belonging to the same subfield are deleted in parallel. A fully parallel approach uses the same deletion criterion in each iteration.

Recently, a series of skeletonization methods based on deep learning techniques have emerged. These meth-ods tend to train nonlinear models such as U-Net [27] to extract skeletons. However, these models cannot guar-antee robustness for nonrigid objects, and their generalization performance is not satisfactory [28-29].

In general, a good skeletonization algorithm should assume some vital properties, which are given as follows [14, 17]:

1. It should maintain the same topology (connectivity) as that of the original object, i.e., it has the same numbers of components and holes.
2. It should be robust to boundary noise so that fewer unwanted branches occur in the skeleton.
3. The resulting skeleton should be thin and one pixel thick.
4. The resulting skeleton should be located in the middle of the object.
5. It should preserve the geometric features of the object, meaning that the skeleton should have compo-nents corresponding to the various sections of the original object.
6. The resulting skeleton should possess the same shape as the object so that it can be used to recover the original object.
7. It should have high computational speed. For iterative methods, a smaller number of iterations corre-sponds to a higher computational speed.

One challenge faced by most skeletonization methods is that they fail to extract complete and clean single-pixel-thick skeletons, corresponding to the former three requirements mentioned above. According to this prob-lem, to obtain a complete and clean single-pixel-thick skeleton, this paper proposes a novel sub-iterative parallel skeletonization method based on the famous Zhang-Suen (ZS) algorithm.

Our target is to maintain and utilize of the advantages of the ZS algorithm, including its high robustness to boundary noise and high computational speed, and to avoid the existing disadvantages,

including the loss of the original topology caused by excessive erosion and a low thinning rate.

The contribution of this paper is that a new skeletonization approach, which has better performance than ex-isting methods in terms of the thinning rate and topology preservation, is proposed. In addition, it maintains similar or slightly better performance in terms of robustness compared with that of the ZS method.

The remainder of this paper is as follows: Section 2 is a review of the related works. The proposed algorithm is presented in Section 3. The evaluation measures are introduced in Section 4, and the experiment is implemented in Section 5. Section 6 concludes the paper. Finally, future work ideas are presented in Section 7.

2 Related Works

In this section, the ZS algorithm, modified ZS (MZS) algorithm, and improved one-path thinning algorithm (IOPTA) are introduced, where the MZS algorithm and IOPTA are both the latest parallel thinning algorithms. The MZS algorithm is a hybrid approach of the sub-iterative and subfield methods based on the ZS algorithm, which exhibits better performance according to a comprehensive comparison with many other methods. On the other hand, the IOPTA is the latest fully parallel thinning algorithm that is used to extract fingerprints.

2.1 The ZS Algorithm

The ZS algorithm was proposed by Zhang and Suen [12] in 1984; it adopts the 8-neighborhood method with a three-by-three window to check each potential candidate foreground pixel and delete those pixels when they satisfy certain conditions. Here, P_1 denotes the current pixel, and its 8-neighborhood window is shown in Fig. 2.

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

Fig. 2. 8-neighborhood window

In the ZS algorithm, there are two sub-iterations. In each sub-iteration, a foreground pixel P_1 whose value is one, will change to a background pixel whose value is 0 if it holds corresponding deletion conditions presented in Table 1.

Table 1. Deletion Condition used in the ZS algorithm

Deletion Conditions in Sub-iteration 1		Deletion Conditions in Sub-iteration 2	
$2 \le B(P_1) \le 6$	(ZS1.1)	$2 \le B(P_1) \le 6$	(ZS2.1)
$A(P_1) = 1$	(ZS1.2)	$A(P_1) = 1$	(ZS2.2)
$P_2 \times P_4 \times P_6 = 0$	(ZS1.3)	$P_2 \times P_4 \times P_8 = 0$	(ZS2.3)
$P_4 \times P_6 \times P_8 = 0$	(ZS1.4)	$P_2 \times P_6 \times P_8 = 0$	(ZS2.4)

$A(P_1)$ is the number of 01 pairs in the ordered set $\{P_2, P_3, ..., P_8, P_9\}$ that contains the eight neighbors of P_1, and $B(P_1)$ denotes the number of black pixels in the 8-neighborhood of P_1. The iterative process stops when no more pixels are removed.

The merit of the ZS algorithm is that it considers the directions of the deletable pixels during the removal process. Therefore, it exhibits strong robustness to boundary noise. However, one of the main problems faced by the ZS algorithm is that when we use it to process some shapes, such as crossed lines or squares, the topologies cannot be well preserved. Another main problem is that the skeleton resulting from the ZS algorithm is not always one pixel wide, which may increase the difficulty of some applications.

In summary, the ZS algorithm can produce a clean skeleton. However, this skeleton is not one pixel thick everywhere, and some vital topology may be lost.

2.2 The MZS Algorithm

The MZS algorithm was proposed by Lynda Ben Boundaoud, Basek Solaiman and Abdelkamel Tari [17] to overcome the ZS algorithm's defects. In their paper, the MZS algorithm was compared with seven other algorithms [11-12, 14, 21, 30-32], and experiments proved that the MZS algorithm achieved the best performance in terms of the thinning rate and topology preservation. The MZS algorithm inherits some characteristics from the ZS algorithm, such as the fact that it also works with a 3-by-3 window and has two sub-iterations; however, the MZS algorithm adds one additional deletion condition and modifies some existing deletion conditions contained in the ZS algorithm. In addition, to prevent excessive erosion, the MZS algorithm introduces four retaining templates in the second sub-iteration. The deletion conditions and retaining templates are shown in Table 2 and Fig. 3, respectively.

Table 2. Deletion Condition used in the MZS algorithm

Deletion Conditions in Sub-iteration 1		Deletion Conditions in Sub-iteration 2	
$(i+j) \bmod 2 = 0$	(MZS1.1)	$(i+j) \bmod 2 \neq 0$	(MZS2.1)
$2 \le B(P_1) \le 7$	(MZS1.2)	$1 \le B(P_1) \le 7$	(MZS2.2)
$A(P_1) = 1$	(MZS1.3)	$A(P_1) = 1$	(MZS2.3)
$P_2 \times P_4 \times P_6 = 0$	(MZS1.4)	$P_2 \times P_4 \times P_8 = 0$	(MZS2.4)
$P_4 \times P_6 \times P_8 = 0$	(MZS1.5)	$P_2 \times P_6 \times P_8 = 0$	(MZS2.5)

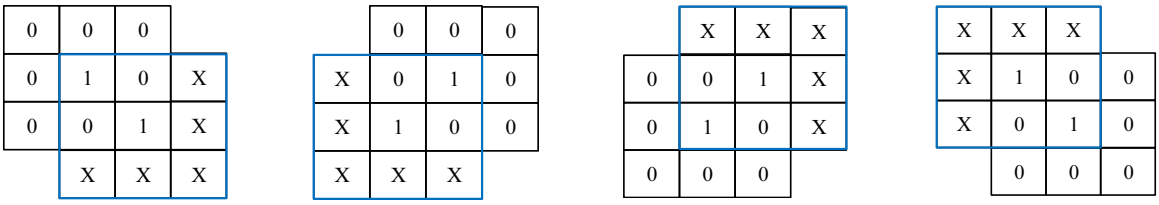


Fig. 3. Retaining templates used in the MZS algorithm

2.3 The IOPTA Algorithm

The IOPTA algorithm [26] was proposed by Rui-Zheng Wang and his coworkers based on the OPTA algorithm [20], in which eight elimination templates and six restoration templates were applied. These templates are shown in Fig. 4 and Fig. 5. According to the author, their algorithm can significantly reduce the occurrence of burrs in the thinning process, and the thinning time and memory space are effectively saved compared with the OPTA algorithm and its improved version [24, 33].

2.4 Summary

Although the MZS algorithm proved that it overcomes the existing problems faced by the ZS algorithm, the authors did not conduct noise experiments in their paper. Their modification may decrease the robustness of the algorithm to boundary noise. In fact, their modification is more sensitive to boundary noise than the ZS algorithm.

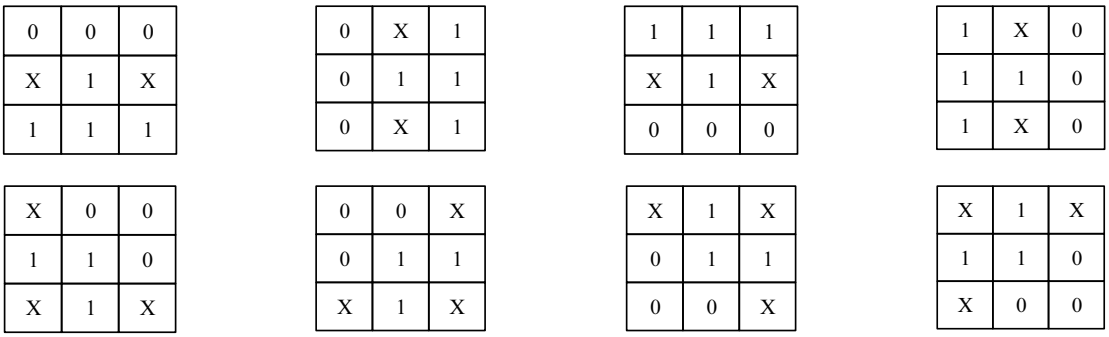


Fig. 4. Deletion templates used in the IOPTA algorithm

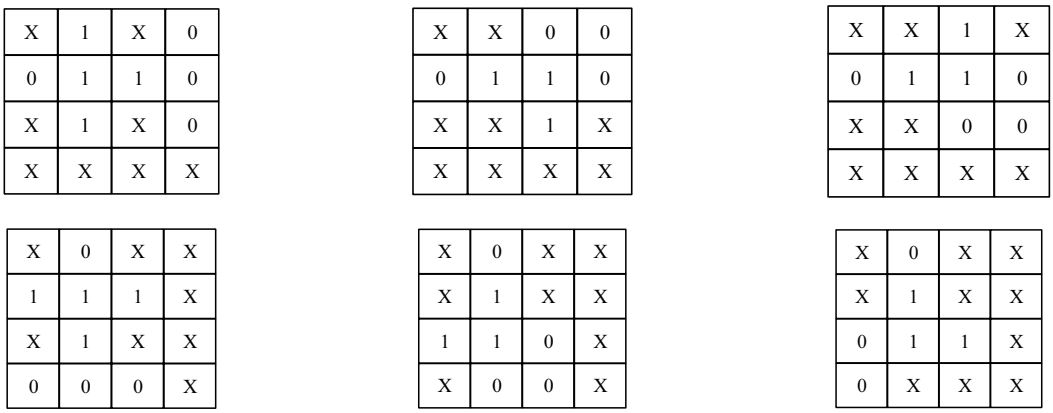


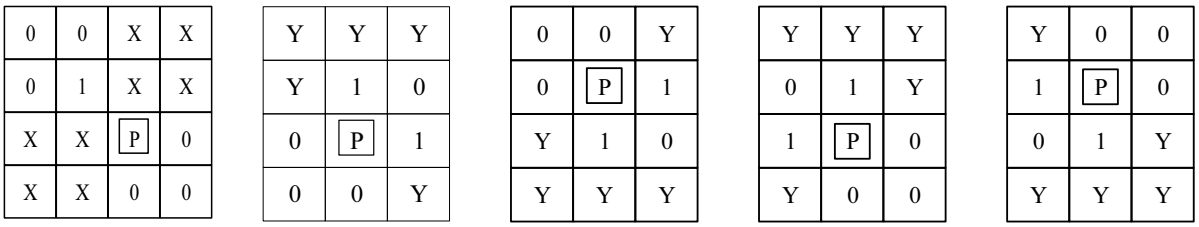
Fig. 5. Restoring templates used in the IOPTA algorithm

Regarding the IOPTA, although the result exhibited a lower occurrence of burr, the IOPTA was only compared with other OPTA-series algorithms. The method was not compared with sub-iterative parallel thinning algorithms. Therefore, it is unclear whether the performance of the IOPTA is better than that of the ZS algorithm. In fact, the robustness of their approach is worse than that of the ZS algorithm.

As a result, it is necessary to propose a new parallel algorithm that not only has similar robustness to the ZS algorithm, but also preserves the original topology and realizes a one-pixel skeleton.

3 Proposed Method

The proposed method is a sub-iterative parallel thinning algorithm based on the ZS algorithm. In order to deal with the ZS's drawback, we add a template match procedure, in which four deletion templates and one restoring template are deployed. These templates are shown in Fig. 6.



(a) restoring template

(c) are two deletion templates used in first sub-iteration

(c) are two deletion templates used in first sub-iteration

(d) are two deletion templates used in second sub-iteration

(e) are two deletion templates used in second sub-iteration

Fig. 6. Templates used in the proposed algorithm

In Fig. 6, the symbols '0', '1', 'P' and 'x' in these templates denote a background pixel, a foreground pixel, the currently tested pixel and an ignorable pixel whose value can be either 0 or 1, respectively, whereas 'Y' denotes that at least one of the pixels represented by the set of symbols should be a foreground pixel. The deletion conditions used in the proposed algorithm are presented in Table 3.

Table 3. Deletion Conditions used in the proposed algorithm

Deletion Conditions in Sub-iteration 1		Deletion Conditions in Sub-iteration 2	
$2 \le B(P_1) \le 6$	(P1.1)	$2 \le B(P_1) \le 6$	(P2.1)
$P_2 \times P_4 \times P_6 = 0$	(P1.2)	$P_2 \times P_4 \times P_8 = 0$	(P2.2)
$P_4 \times P_6 \times P_8 = 0$	(P1.3)	$P_2 \times P_6 \times P_8 = 0$	(P2.3)
($A(P_1)$ = 1 ***and*** its neighbors do not match template (a)) ***or*** ($A(P_1)$ = 2 ***and*** match template (b) or (c))	(P1.4)	($A(P_1)$ = 1 ***and*** its neighbors do not match template (a)) ***or*** ($A(P_1)$ = 2 ***and*** match template (d) or (e))	(P2.4)

Deletion templates are used to remove some pixels that do not satisfy the ZS conditions. However, the deletion of these pixels may reduce excessive erosion and improve the performance at a one-pixel thickness. The restoration template is used to preserve some tiny patterns. The flowchart of the proposed algorithm is shown in Fig. 7.

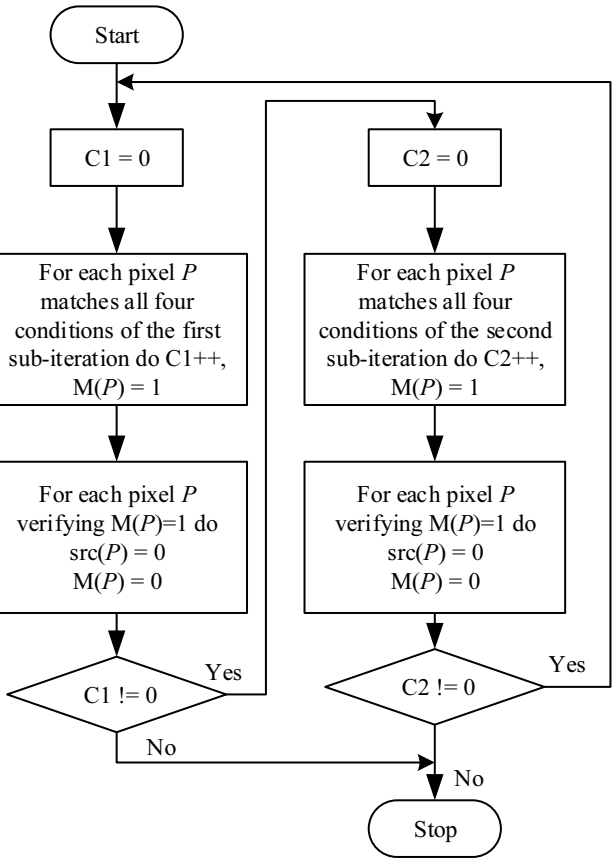


Fig. 7. The flowchart of the proposed method

4 Parameters for Comparison

To evaluate the proposed algorithm, several comparison measures are defined in this section; these are measures for evaluating the performance of the algorithm in terms of unit-width pixels, robustness, speed, and connectivity. Since there is no ideal parameter that can properly describe the performance of topology preservation, we decide not to parameterize it. For convenience of description, suppose that there is a binary image whose size is *m* rows and *n* columns.

4.1 Measure for Evaluating Unit-width Pixels

Thinning rate (TR) was proposed in [23] to evaluate the extent of the thinness of the image.

$$TR = 1 - TM1 \div TM2 \tag{1}$$

$$TM1 = \sum_{i=0}^{n}\sum_{j=0}^{m} TC(P) \tag{2}$$

$$TM2 = 4 \times [\max(m,n) - 1]^2 \tag{3}$$

$$TC(\mathrm{P}) = P \times (P_7 \times P_8) + P_8 \times P_1 + (P_1 \times P_2) + (P_2 \times P_3) \tag{4}$$

If P is a foreground pixel, then the value of TC(P) is equal to the number of triangles whose three vertices are all black pixels as shown in Fig. 8.

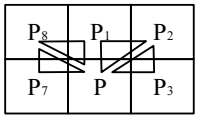


Fig. 8. Vertices of black triangles

It is obvious that if the object is not single pixel wide anywhere in the image, then there exists a triangle composed of three black pixels.

4.2 Measure for Evaluating Sensitivity to the Boundary Noise

Let S be a given image without any boundary noise and S'' be its noisy version caused by randomly removing or adding pixels along the edge of S. In [22], the concept of the signal-to-boundary-noise ratio (*SBNR*) was defined:

$$SBNR = \frac{Area[\partial S]}{Area[S''/S] + Area[S/S'']} \tag{5}$$

Area[] is the operation that counts the number of foreground pixels, ∂S denotes the boundary of S and "S''/S" denotes those pixels belong to set S but not belong to set S''. Thus, the error caused by boundary noise at a particular SBNR can be measured by the normalized quantity.

$$m_e = \min[1, \frac{Area[S_m^{''}/S_m] + Area[S_m/S_m^{''}]}{2 \times Area[S_m]}] \tag{6}$$

where S_m is the resulting skeleton of S and S_m'' is the resulting skeleton of S''. The algorithms that are highly sensitive to boundary noise yield a m_e close to 1.

4.3 Measure for Evaluating Speed

We use the measure of the number of iterations (*NIT*) to describe the computational speed of the proposed algorithm. In general, a faster algorithm conducts fewer iterations. One whole iteration can be considered a full scan of the input image, in which many foreground pixels are changed into background pixels.

4.4 Measure for Evaluating Connectivity

For properly describing the performance of the connectivity, indicators *CM* (connectivity measurement) is introduced. *CM* is proposed in [23] by the following formula.

$$CM = \sum_{i=0}^{n} \sum_{j=0}^{m} S(P) \tag{7}$$

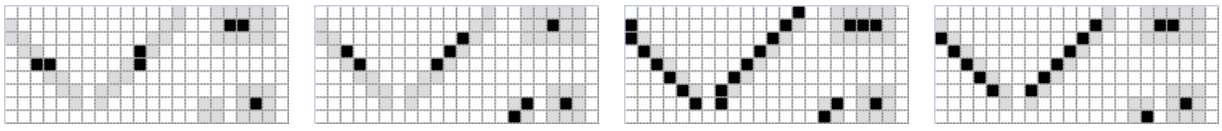
$$S(P) = \begin{cases} 1, & \text{if } B(P) < 2 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

5 Experiments and Results

The ZS algorithm, the MZS algorithm, the IOPTA, and the proposed method are implemented on the MATLAB platform. Three experiments are conducted in this section: a simple pattern experiment, a boundary noise experiment, and a complicated image test. In the simple pattern experiment, we only compare the topology preservation performances among these methods. In the boundary noise experiment, we make many efforts to look for the differences between the robustness performances of these approaches. The complicated image experiment is conducted to observe the effects of all these methods on real images; the test images come from well-known benchmarks.

5.1 Simple Pattern Experiment

Several simple patterns, such as 2×2 squares, slope lines and rectangles, are collected for the experiment. There are two reasons for choosing these patterns rather than others. The first is that they are so simple that their topologies are very easy to recognize. In addition, these patterns usually appear in the last iteration of the thinning process for more complex patterns. To enhance the visual effect, the test image is marked in gray, and the black color imposed on the test image is the resulting skeleton, as shown in Fig. 9.



(a) Skeleton generated by the ZS algorithm (b) Skeleton generated by the MZS algorithm (c) Skeleton generated by the IOPTA algorithm (d) Skeleton generated by the proposed algorithm

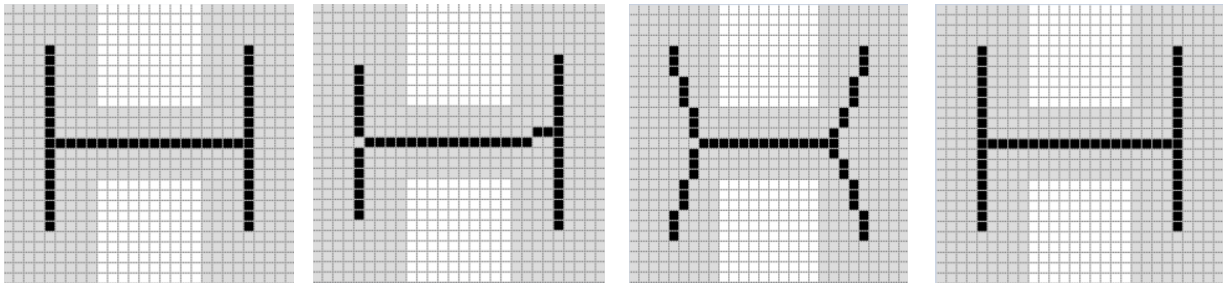
Fig. 9. Result of the Simple pattern experiment

It is apparent that the proposed method has almost the same high-quality result as that of the IOPTA algorithm. Both of them generate better skeletons than the ZS and MZS algorithms in terms of topology preservation. The ZS and MZS methods suffer different degrees of excessive erosion. In detail, the original diagonal lines degenerate to straight lines, and the 2-by-2 squares are completely removed in the skeleton resulting from the ZS method. On the other hand, the MZS algorithm produces the same dots for both squares and rectangles.

5.2 Boundary Noise Experiment

To study the robustness of these algorithms, two comparisons should be conducted. Here, we care about which of the approaches can produce a skeleton with the fewest changes when faced with different levels of boundary noise. As a result, for each algorithm, we compare the result generated from a noiseless image with that generated from a version with a certain level of nose and record the degree of change as data. Then, a second comparison is conducted between the different algorithms on these data. Thus, it is first necessary to extract the clean skeleton from the noiseless test image for later comparisons.

We take the image used in [12] as the initial test image. It is simple and does not contain any boundary noise. The skeletons extracted from it by all four algorithms are shown in Fig. 10.

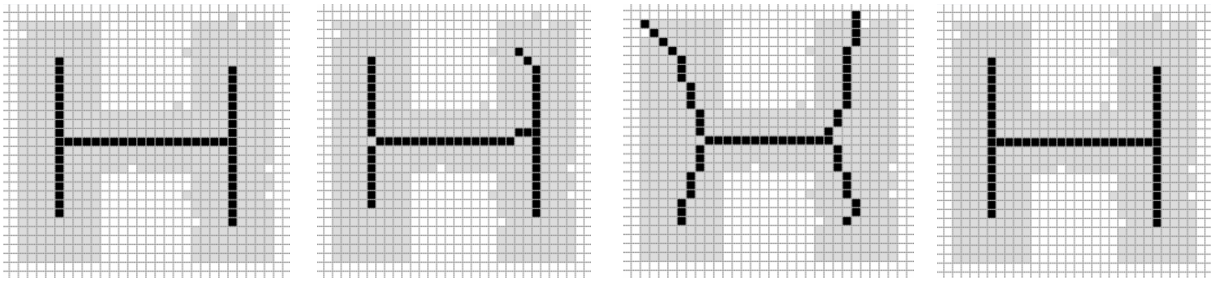


(a) Skeleton yielded by the ZS method | (b) Skeleton yielded by the MZS method | (c) Skeleton yielded by the IOPTA method | (d) Skeleton yielded by the proposed method

Fig. 10. Skeleton extract from the noiseless image

Next, the edge pixels in the original image are randomly flipped according to a certain probability, where the foreground pixels change into background pixels and background pixels change into foreground pixels. The different flip probabilities are considered different levels of noise. We first introduce 2.5% noise to the edges of the original image, and the noise continues to increase until it reaches 15%. Then, all implemented thinning algorithms are used to process these noisy images, and the results are observed. The results are evaluated by the SBNR and m_e parameters, which are introduced in Section 4. At each level of noise, the same test is repeated 50 times independently. Fig. 11 to Fig. 13 are actual samples of illuminated results obtained under the different noisy images.

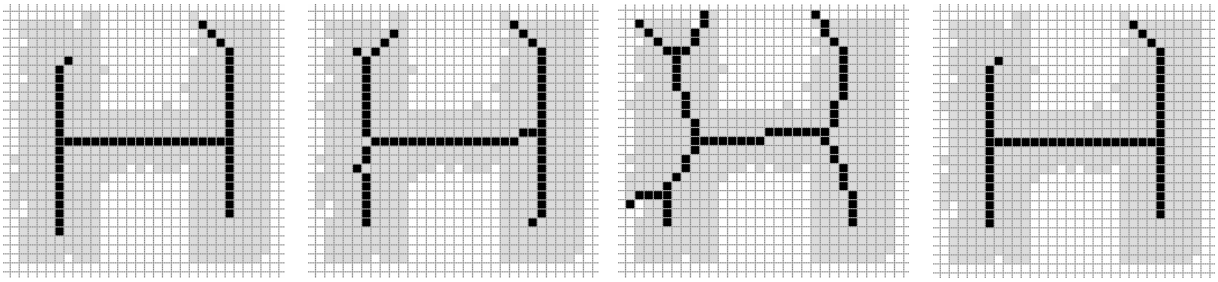
Fig. 11 shows the skeletons extracted from the image with 5% boundary noise. From the visual effects, we notice that the result extracted by the proposed method roughly maintains a clean skeleton, where there are no apparent changes when compared with the initial skeleton in Fig. 10. In contrast, more changes appear in the results of the MZS algorithm and the IOPTA algorithm, whereas the ZS algorithm exhibits a similar performance to that of the proposed algorithm. From the perspective of the measured m_e, the proposed method achieves a value of 0.0185, which is better than that of the MZS algorithm (m_e = 0.0408) and the IOPTA (m_e =0.19), but slightly worse than that of the ZS algorithm (m_e = 0.0092).



(a) Skeleton yielded by the ZS method (m_e = 0.0092) | (b) Skeleton yielded by the MZS method (m_e = 0.0408) | (c) Skeleton yielded by the IOPTA method (m_e = 0.19) | (d) Skeleton yielded by the proposed method (m_e = 0.0185)

Fig. 11. Skeleton extract from the image with 5% boundary noise, whose *SBNR* = 14.6

Fig. 12 and Fig. 13 present the results extracted from noisy images with 10% and 15% boundary noise, respectively. From the visual effects, the appearances of the output of the proposed algorithm and that of the ZS algorithm are very close. Both algorithms produce a little variety in the skeletons, but these changes are less than those induced in their counterparts produced by the MZS algorithm and the IOPTA. On the other hand, it can be learned that the proposed algorithm has the smallest m_e value, which demonstrates that the proposed algorithm is the most robust algorithm when extracting skeletons from these two specific noisy images.



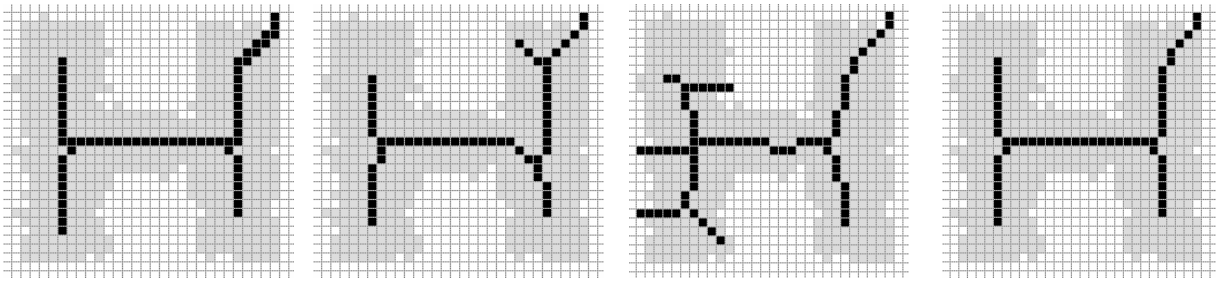
(a) Skeleton yielded by the ZS method (m_e = 0.0741)

(b) Skeleton yielded by the MZS method (m_e = 0.1633)

(c) Skeleton yielded by the IOPTA method (m_e = 0.66)

(d) Skeleton yielded by the proposed method (m_e = 0.0648)

Fig. 12. Skeleton extract from the image with 10% boundary noise, whose *SBNR* = 5.41



(a) Skeleton yielded by the ZS method (m_e = 0.1388)

(b) Skeleton yielded by the MZS method (m_e = 0.2653)

(c) Skeleton yielded by the IOPTA method (m_e = 0.7)

(d) Skeleton yielded by the proposed method (m_e = 0.1296)

Fig. 13. Skeleton extract from the image with 15% boundary noise, whose *SBNR* = 3.18

Table 4 presents the mean values of the SBNR and the mean values of the m_e for all four algorithms under different levels of boundary noise. It is not hard to find that the proposed algorithm is slightly better than the ZS algorithm. It is the most robust approach among these methods because the m_e of the proposed method is lowest under all noise levels.

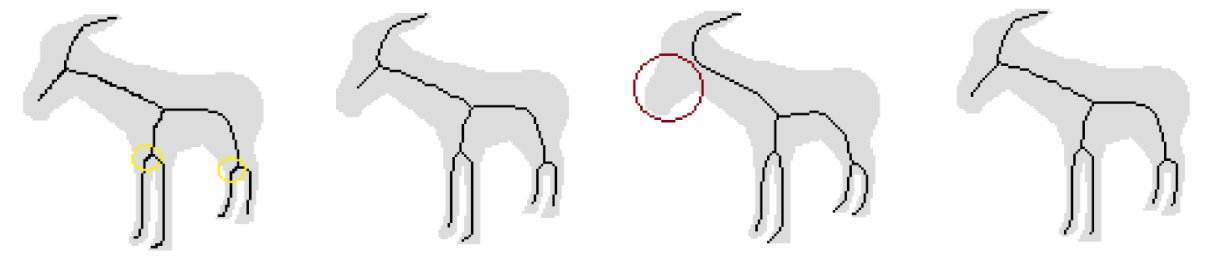
Table 4. The mean effect of me under different noise level for different algorithms (Sample size is 50)

Noise level	Mean of signal-to-boundary noise rate (SBNR)	Mean of the normalized error (m_e)			
		ZS	MZS	IOPTA	Proposed
2.5%	32.21	0.012	0.033	0.128	0.011
5%	12.77	0.033	0.075	0.249	0.030
7.5%	8.25	0.069	0.129	0.365	0.062
10%	5.76	0.110	0.154	0.495	0.094
12.5%	4.76	0.111	0.139	0.598	0.093
15%	4.18	0.162	0.209	0.650	0.143

5.3 Complicate Image Experiment

To evaluate the algorithmic performance in terms of single-pixel thickness, connectivity, and thinning speed, shape preservation, all the implemented algorithms are tested not only on the well-known benchmarks of KIMIA-99 and MPEG-7 but also on several images collected from the Internet. Fig. 14 to Fig. 18 show several different kinds of test images and the results generated by all four thinning algorithms. To effectively present the visual effects of the skeleton, we remove the grid in the image and use yellow color and brown color to highlight the parts of the skeleton that violate the criteria of single-pixel thickness and shape preservation, respectively.

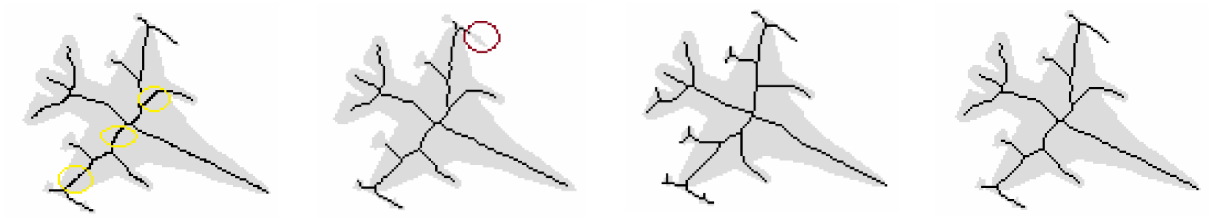
In Fig. 14, the original image is a donkey. The proposed method produces a clean and complete one-pixel skeleton. In contrast, in the output of the classical ZS algorithm, there are many multipixel bony segments, such as the areas marked by the yellow circles. The IOPTA completely deletes the topological structure of the mouth of the donkey (see the red part in (c)).



(a) Skeleton yielded by the ZS method | (b) Skeleton yielded by the MZS method | (c) Skeleton yielded by the IOPTA method | (d) Skeleton yielded by the proposed method

Fig. 14. Skeleton extract from the image of donkey

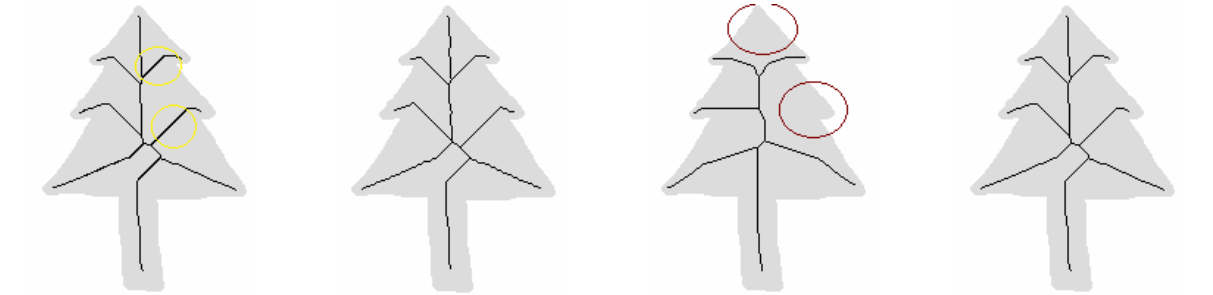
For the airplane image (Fig. 15), the proposed method generates a better skeleton than those generated by the other three methods. The MZS algorithm excessively reduces the length of the small missile located on the top right. Therefore, it is not similar to the one found on the bottom left, which causes asymmetry in the image. The ZS algorithm cannot produce a unit-width skeleton. Therefore, the output of the IOPTA may seem better than the former two in terms of shape preservation and the unit-width skeleton, but it has many unwanted branches that need a further pruning procedure.



(a) Skeleton yielded by the ZS method | (b) Skeleton yielded by the MZS method | (c) Skeleton yielded by the IOPTA method | (d) Skeleton yielded by the proposed method

Fig. 15. Skeleton extract from the image of airplane

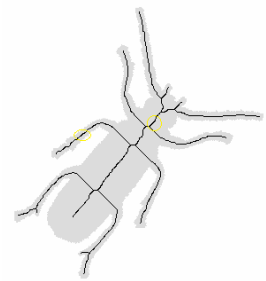
The proposed and MZS algorithms can generate good skeletons for the tree, as shown in Fig. 16. The output extracted by the IOPTA algorithm cannot effectively reflect the original tree because the top of the tree and the right parts are deleted by mistake.



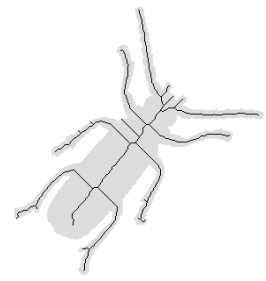
(a) Skeleton yielded by the ZS method | (b) Skeleton yielded by the MZS method | (c) Skeleton yielded by the IOPTA method | (d) Skeleton yielded by the proposed method

Fig. 16. Skeleton extract from the image of tree

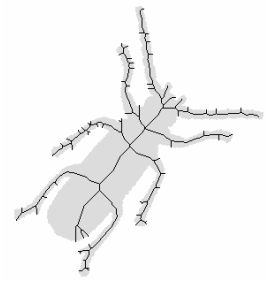
A beetle with an unsmooth edge is presented in Fig. 17. The proposed method can maintain a good skeleton, where each skeleton segment is located at the median line, and there are no unwanted branches. In contrast, the IOPTA and the MZS approach produce many unnecessary spurs, and their partial skeletons are not located at the center of the original image.



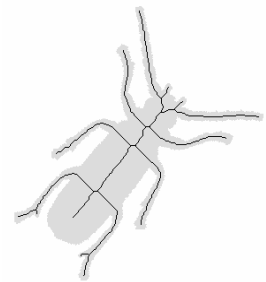
(a) Skeleton yielded by the ZS method



(b) Skeleton yielded by the MZS method



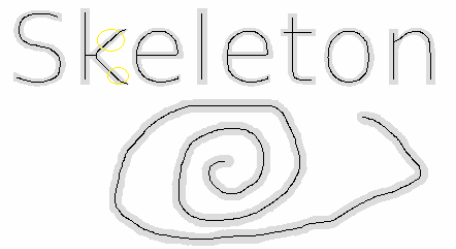
(c) Skeleton yielded by the IOPTA method



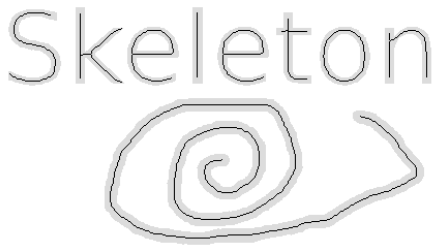
(d) Skeleton yielded by the proposed method

Fig. 17. Skeleton extract from the image of beetle

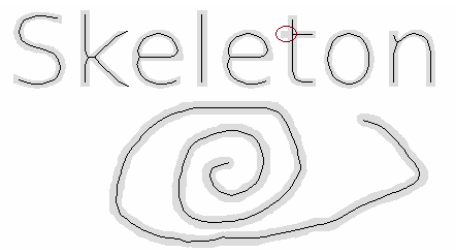
In Fig. 18, the original test image is composed of some letters and a large spiral. Our novel parallel thinning method succeeds in generating the centerlines with respect to the input patterns. However, by observing the result of the IOPTA, the left part of the letter 't' is missing.



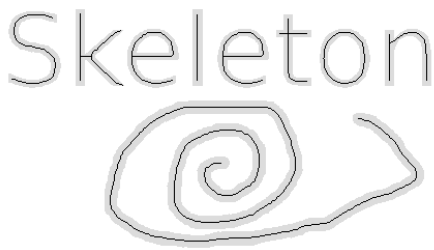
(a) Skeleton yielded by the ZS method



(b) Skeleton yielded by the MZS method



(c) Skeleton yielded by the IOPTA method



(d) Skeleton yielded by the proposed method

Fig. 18. Skeleton extract from the image of vocabulary

Table 5 summarizes the numerical results of Fig. 14 to Fig. 18. It can be seen that the proposed algorithm is pretty good. In terms of thinning speed, our method requires one more iteration than the ZS algorithm and generally requires fewer iterations than the IOPTA algorithm. Next, in terms of the thinning rate, the proposed algorithm is similar to the MZS algorithm, and both modifications are better than the classical ZS method. Then, the CM parameter of the IOPTA is sometimes below or above those of the other methods, which denotes that the IOPTA sometimes mistakenly removes some important object pattern structures or sometimes produces unwanted branches.

Table 5. Comparison table under four algorithms

Image	Parameter	ZS	MZS	IOPTA	Proposed
Donkey	NIT	30	31	33	31
	TR	0.9990	1	0.9999	1
	CM	6	6	5	6
Airplane	NIT	35	36	43	36
	TR	0.9986	0.9998	0.9997	0.9998
	CM	12	12	17	12
Tree	NIT	73	74	89	74
	TR	0.9994	1	1	1
	CM	8	8	6	8
Beetle	NIT	55	56	70	56
	TR	0.9997	1	0.9999	1
	CM	13	19	67	13
Vocabulary	NIT	14	15	17	15
	TR	0.9994	1	1	1
	CM	19	19	18	19

6 Conclusion

In this paper, for extracting a clean and entire one-pixel skeleton, based on the classical ZS algorithm, a novel sub-iterative parallel thinning method is proposed. We modify the partial deletion conditions of the original algorithm and introduce several templates for eliminating redundancy and preserving topology throughout the thinning process. Experimental results reflect that the proposed method has a good effect on suppressing edge noise, fully maintaining topology, and extracting single-pixel skeletons.

7 Future Work

There are two research directions that I would like to try in the future. The first is attempting to apply our approach to the three-dimensional image after some necessary modifications. Since skeletons are usually considered compact descriptions of the original patterns, it also will be an attractive idea to study classification problems based on the skeletons produced by our algorithm.

References

[1] H. Blum, A transformation for extracting new descriptors of shape, Cambridge, MA: MIT press (1967) 362-380.

[2] P.K. Saha, G. Borgefors, G. Sanniti di Baja, A survey on skeletonization algorithms and their applications, Pattern Recognition Letter 76(2016) 3-12.

[3] R.L. Ogniewicz, O. Kübler, Hierarchic Voronoi skeletons, Pattern Recognition 28(1995) 343-359.

[4] J.W. Brandt, V.R. Algazi, Continuous skeleton computation by Voronoi diagram, CVGIP: Image Understanding 55(3) (1992) 329-338.

[5] F. Leymarie, M.D. Levine, Simulating the grassfire transform using an active contour model, IEEE Transactions on Pattern Analysis and Machine Intelligence 14(1992) 56-75.

[6] C. Aslan, A. Erdem, E. Erdem, S. Tari, Disconnected skeleton: Shape at its absolute scale, IEEE Transactions on Pattern Analysis and Machine Intelligence 30(2008) 2188-2203.

[7] L. Ben Boudaoud, A. Sider, A. Tari, A new thinning algorithm for binary images, in: Proc. of 3rd International Conference on Control, Engineering & Information Technology (CEIT), 2015.

[8] C. Arcelli, G. Sanniti Di Baja, L. Serino, Distance-driven skeletonization in voxel images, IEEE Transactions on Pattern Analysis and Machine Intelligence 33(2011) 709-720.

[9] L. Lam, S.W. Lee, Thinning methodologies – a comprehensive survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 14(1992) 869-885.

[10] L. Ben Boudaoud, B. Solaiman, A. Tari, Implementation and comparison of binary thinning algorithms on GPU, Computing 101(2018) 1091-1117.

[11] H. E. Lu, P. S. P. Wang, A comment on “a Fast Parallel Algorithm for Thinning Digital Patterns”, Communications of the ACM 29(1987) 239-242.

[12] T.Y. Zhang, C.Y. Suen, A Fast Parallel Algorithm for Thinning Digital Patterns, Communications of the ACM 27(1984) 236-239.

[13] A. Jagna, V. Kamakshiprasad, New parallel binary image thinning algorithm, ARPN Journal of Engineering and Applied sciences 5(2010) 64-67.

[14] P. Tarabek, A robust parallel thinning algorithm for pattern recognition, in: Proc. of 7th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), 2012.

[15] J. Dong, Y. Chen, Z. Yang, B.W.K. Ling, A parallel thinning algorithm based on stroke continuity detection, Signal, Image and Video Processing 11(2017) 873-879.

[16] R. Li, X. Zhang, Research on the Improvement of EPTA Parallel Thinning Algorithm, Advances in Intelligent Systems Research 147(2018) 994-1001.

[17] L. Ben Boudaoud, B. Solaiman, A. Tari, A modified ZS thinning algorithm by a hybrid approach, The Visual Computer 34(2018) 689-706.

[18] C. Ma, S. Wan, J. Lee, Three-Dimensional Topology Preserving Reduction on the 4-Subfields, IEEE Transactions on Pattern Analysis and Machine Intelligence 24(2002) 1594-1605.

[19] M. Kmenandrichard, Parallel Shrinking Algorithms Using 2-Sub-fields Approaches, Computer vision, graphics, and image processing 52(1990) 191-209.

[20] R.T. Chin, H.K. Wan, D.L. Stover, R.D. Iverson, A one-pass thinning algorithm and its parallel implementation, Computer Vision, Graphics, and Image Processing 40(1987) 30-40.

[21] R.Y. Wu, W.H. Tsai, A new one-pass parallel thinning algorithm for binary images, Pattern recognition letters 13(1992) 715-723.

[22] B.K. Jang, R.T. Chin, One-Pass Parallel Thinning: Analysis, Properties, and Quantitative Evaluation, IEEE Transactions on Pattern Analysis & Machine Intelligence 14(1992) 1129-1140.

[23] R.W. Zhou, C. Quek, G.S. Ng, A novel single-pass thinning algorithm and an effective set of performance criteria, Pattern Recognition Letters 16(1995) 1267-1275.

[24] L. Zhao, Q. Chen, Z. Chen, A new Improved OPTA Refinement Algorithm, Computer Applications 10(28)(2008) 2639-2642.

[25] L. Guan, Z. Li, Z. Wang, A Fast and Complete Thinning Algorithm for Character Image, in: Proc. of 2nd International Conference on Mechatronics, Control and Automation Engineering, 2017.

[26] R. Wang, Y. Zhao, T. Ji, L. Liu, Fingerprint Refinement Model Based on Improved OPTA, Journal of computer (Taiwan) 31(2020) 274-283.

[27] O. Panichev, A. Voloshyna, U-net based convolutional neural network for skeleton extraction, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[28] S. Nathan, P. Kansal, Skeletonnet: Shape pixel to skeleton pixel, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[29] R. Atienza, Pyramid u-network for skeleton extraction from shape points, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[30] M. Ahmed, R. Ward, A rotation invariant rule-based thinning algorithm for character recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 24(2002) 1672-1678.

[31] A. Jagna, V. Kamakshiprasad, New parallel binary image thinning algorithm, ARPN Journal of Engineering and Applied sciences 5(2010) 64-67.

[32] D. Kocharyan, A Modified Fingerprint Image Thinning Algorithm, American journal of software engineering and applications 2(1)(2013) 1-6.

[33] J. Zang, J. Yuan, S. Du, Research on Fingerprint Matching Algorithm based on Fourier-Mellin Transform, Computer Simulation 3(2)(2009) 25-28.