A Robust Parallel Thinning Algorithm for Pattern Recognition

Peter Tarabek*
* University of Zilina, Department of Transportation Networks, Zilina, Slovakia
peter.tarabek@fri.uniza.sk

Abstract— **Thinning algorithms have been used in pattern recognition and image analysis for a long time. They reduce a binary digital pattern to obtain a unit width skeleton which retains geometrical and topological properties. These properties are important for robust recognition of characters, handwritings, fingerprints, transportations infrastructure and other. A robust parallel thinning algorithm based on the popular Zhang and Suen algorithm is presented. The ZS algorithm is very good in respect to both connectivity and insensitivity to boundary noise but it tends to remove diagonal line segments and whole 2x2 square patterns, and does not produce a unit width skeleton. The experimental results show that the proposed method preserves good properties of ZS algorithm and it overcomes the disadvantages by incorporating additional conditions for identifying the crucial patterns and by applying a post-processing step that removes all redundant pixels so the one pixel thick skeleton is produced.**

I. Introduction

Thinning plays important role in pattern recognition, which is an essential part of artificial intelligence. It is often used as a pre-processing step in wide variety of applications such as character and handwriting recognition and analysis [1, 2], medical imaging [3, 4], vectorization of transportation infrastructure [5], fingerprint recognition [6] and robot vision. Thinning is a process of reducing a digital binary pattern to a unit width representation called skeleton. As the result, all redundant points are removed and the topological and geometric properties of pattern such as connectivity, topology, length and width are preserved. The skeleton is often useful for representing elongated patterns because it is closer to the human conception, which makes further analysis simpler and more intuitive.

The design of thinning algorithms attracts lots of attention, therefore many papers have been written on this subject [6-12]. Besides the conventional approaches there have been some attempts to apply soft computing techniques such as neural networks [13] and fuzzy sets [14, 15] to obtain skeleton. Although the exact definition of skeleton is missing, there is a general agreement about the properties of a good thinning algorithm. These include:

- Preservation of topological and geometric properties: connectivity, shape and position of the end points and junction points should be preserved.
- One pixel thickness or unit width skeleton.
- Mediality: skeleton should lie in the middle of the shape.
- Insensitivity to boundary noise.
- Isotropy or rotation invariance.
- Efficiency: high processing speed.

Because some of the requirements may act in contradiction, different class of algorithms have different trade offs between these properties.

The thinning algorithms can be classified as iterative or non-iterative. Iterative algorithms remove boundary pixels layer by layer until an appropriate skeleton is obtained. According to the way of examining pixels, they can be further divided into sequential and parallel. A sequential algorithm processes pixels in a fixed sequence in every iteration. The deletion of pixels depends on all previous operations which have been performed including those made in current iteration. In a parallel algorithm, all pixels can be processed simultaneously, because the deletion of pixels depends only on the results of the previous iterations. Non-iterative methods, such as medial axis transforms, are not processing all individual pixels and produce a skeleton in one pass. Algorithms based on medial axis transforms are very sensitive to boundary noise which allows them to exactly recreate the original pattern, but for the same reason they are not useful for pattern recognition [7].

Iterative parallel algorithms are considered to be superior over sequential algorithms in many ways [8]:

- The deletion of pixels in a sequential algorithm is more unpredictable, because the result depends partly on the order in which pixels are processed.
- The result of parallel algorithm could be more isotropic.
- Parallel algorithms can be implemented using parallel hardware.

The main problem of parallel approaches is to maintain connectivity using a reasonable number of operations, which in general is easier to do for sequential methods.

Over the years the ZS algorithm proposed by Zhang and Suen [9] has become one of the most cited and used parallel thinning algorithms. Although rather old, it is still probably the most used among the researchers in various applications. The ZS algorithm uses simple conditions and is very good with respect to both connectivity and insensitivity to boundary noise. It can process all pixels simultaneously and is relatively fast also on sequential hardware. However, there are three main difficulties considering the ZS algorithm: excessive erosion in thinning diagonal lines, complete deletion of patterns

978-1-4673-1014-7/12/$31.00 ©2012 IEEE

which can be reduced to 2x2 square and creation of skeleton which is not one pixel thick. These problems are illustrated in Fig. 1.

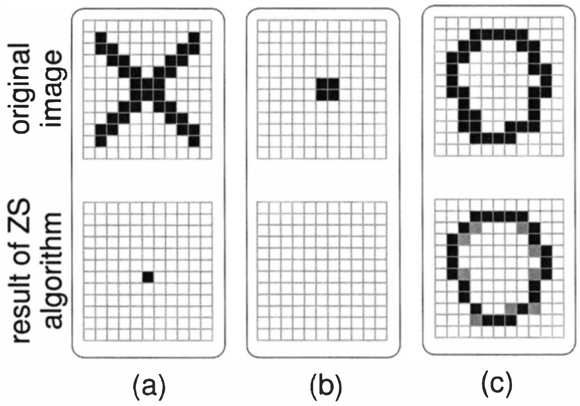

Figure 1. Problems of ZS algorithm: (a) excessive erosion of diagonal lines, (b) complete deletion of 2x2 squares, (c) redundant pixels in skeleton

In this paper an improved parallel thinning algorithm based on ZS is proposed. It preserves good properties of the original ZS algorithm, is robust to boundary noise and excessive erosion, produces one pixel thick skeleton and maintains approximately the same computational speed. Experimental results confirm efficiency and robustness of the proposed method.

The ZS algorithm and some of the existing modifications are briefly described in section 2. In section 3, the existing problems are discussed in greater detail and proposed modifications are presented. Section 4 shows experimental results and conclusions are drawn in section 5.

II. ZS ALGORITHM AND EXISTING MODIFICATIONS

The following notation is adopted in this paper. A binary digital image is defined by matrix $I(i,j)$, where each pixel is either background 0 or foreground 1. The pixels $P_1,\ldots,P_8$ are the 8-neighbors of the pixel P denoted by $N_8(P)$ and labeled as shown in Fig. 2. The $N_8(P)$ consist of two types of neighbors: the 4-neighbors ($N_4(P)$) also called edge neighbors and D-neighbors ($N_D(P)$) also called diagonal or point neighbors. The number of foreground pixels in $N_8(P)$ is denoted as $B(P)$ and $A(P)$ denotes the number of 0 to 1 patterns in the ordered set $P_1, P_2,\ldots, P_8$ in $N_8(P)$. Also it should be noted than it is well established to use black color as the foreground color in images and illustrations.

$P_8[i\text{-}1,j\text{-}1]$	$P_1[i\text{-}1,j]$	$P_2[i\text{-}1,j+1]$
$P_7[i,j\text{-}1]$	$P[i,j]$	$P_3[i,j+1]$
$P_6[i+1,j\text{-}1]$	$P_5[i+1,j]$	$P_4[i+1,j+1]$

Figure 2. 8-neighborhood of pixel P

In the ZS algorithm, the problem of connectivity preservation is solved by dividing each iteration into two subiterations. By using two subiterations only information from 3x3 neighborhood need to be considered for pixel deletion. In the first subiteration, the foreground pixel P is deleted if it satisfies the following conditions:

$$2 \le B(P) \le 6 \tag{1}$$

$$A(P) = 1 \tag{2}$$

$$P_1 * P_3 * P_5 = 0 \tag{3}$$

$$P_3 * P_5 * P_7 = 0 \tag{4}$$

In the second subiteration, only conditions (3) and (4) are changed as follows:

$$P_1 * P_3 * P_7 = 0 \tag{5}$$

$$P_1 * P_5 * P_7 = 0 \tag{6}$$

The remaining conditions are the same.

The already mentioned problems of the ZS algorithm have been studied in several papers. In order to preserve diagonal lines, the LW algorithm [10] modified condition (1) to:

$$3 \le B(P) \le 6 \tag{7}$$

Although the diagonal lines are not erased, they are also not thinned to unit width. In [11] additional pass was implemented to further improve the previous method, but the algorithm creates needless trees [12] and don't produce the unit width skeleton. The KWK algorithm [12] also uses condition (7) and proposed a postprocessing step in which the pixel is removed by the following conditions:

$$P_8 * P_7 * P_5 = 1 \quad \& \quad P_2 = 0 \tag{8}$$

$$P_2 * P_3 * P_5 = 1 \quad \& \quad P_8 = 0 \tag{9}$$

$$P_4 * P_5 * P_7 = 1 \quad \& \quad P_2 = 0 \tag{10}$$

$$P_3 * P_5 * P_6 = 1 \quad \& \quad P_8 = 0 \tag{11}$$

Authors claim that the diagonal lines are not erased and are thinned to one pixel width. Furthermore the position of end points is improved compared to ZS algorithm. The KWK along with the ZS algorithm are used for comparison with the proposed method and their results are further discussed in section 4.

III. THE PROPOSED METHOD

In this section the problems of the ZS algorithm are discussed in more detail. We show what configuration of patterns lead to the problems and we present modifications to resolve these issues.

A. Diagonal lines preservation

The problem of excessive erosion of diagonal lines lays in limited information contained in the 3x3 neighborhood of the processing pixel. From this local point of view it is not possible to distinguish between some situations which need to be processed differently. An example is shown in Fig.3. Here, both pixels have the same properties and are removed by the ZS algorithm. While the pixel in Fig.3(a) should be deleted, deletion of the pixel shown in Fig.3(b) will result in excessive erosion of the pattern (Fig.4).

One solution of this problem is to use modified condition (7) proposed in [10] and adopted by several algorithms discussed in section 2. This will result in preservation of both pixels shown in Fig.3. As a result, diagonal segments are retained, but the skeleton contains additional redundant pixels. Another drawback is that the algorithms based on this modification are more sensitive to boundary noise as will be shown in section 4.

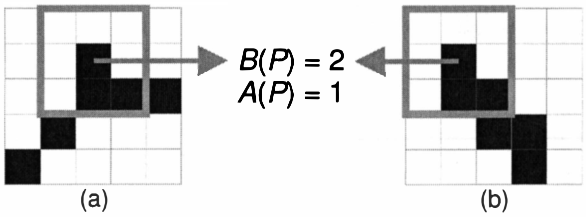

Figure 3. Limited local information

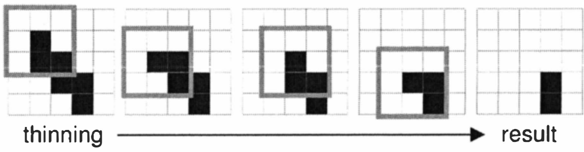

Figure 4. Excessive erosion of diagonal line segment

We propose to process each pixel differently. This is ensured by examining wider neighborhood of the pixel P shown in Fig.5.

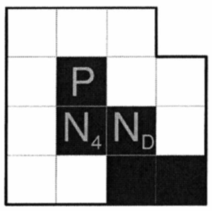

Figure 5. Examined neighborhood

The pixel P is retained if the following conditions are satisfied:

$$B(P)=2 \quad \& \quad A(P)=1 \tag{12}$$

$$B(N_4)=3 \quad \& \quad A(N_4)=2 \tag{13}$$

$$B(N_D)=4 \quad \& \quad A(N_D)=2 \tag{14}$$

Where N_4 and N_D are neighbors of the pixel P. These conditions are sufficient to preserve diagonal lines.

B. 2x2 squares preservation

Preservation of 2x2 square pattern is important from topological point of view. In ZS algorithm all four pixels of square are removed during one iteration making the square pattern disappear. In general, every pattern which can be reduced by previous iterations to 2x2 square will be completely removed by the ZS algorithm. In compliance with previous notation the 4x4 neighborhood of the pixel P is defined as shown in Fig. 6.

P_8	P_1	P_2	P_9
P_7	P	P_3	P_{10}
P_6	P_5	P_4	P_{11}
P_{15}	P_{14}	P_{13}	P_{12}

Figure 6. 4x4 neighborhood

To resolve this issue, the top left pixel of the square was selected to represent the 2x2 square pattern in resulting skeleton. Because the $B(P)$ and $A(P)$ are already computed by conditions (1) and (2), the fastest way to calculate, if P is the top left pixel of square pattern, is by the following conditions:

$$B(P)=3 \quad \& \quad A(P)=1 \tag{15}$$

$$P_3+P_5=2 \quad \& \quad \sum_{i=9}^{15} P_i = 0 \tag{16}$$

If the conditions are met the pixel is retained. The examined pixels from condition (16) are shown by gray dots in Fig.7.

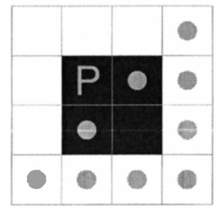

Figure 7. 2x2 squares preservation

C. One pixel thick skeleton

To remove redundant pixels from the skeleton, a post-processing step is proposed. In this step a pixel can be removed if its 3x3 neighborhood matches all defined values in one of the templates shown in Fig. 8(a) and it satisfies at least one of the following conditions:

1. Both N_4 neighbors marked as "X" are background pixels.
2. Exactly one N_4 neighbor marked as "X" is background pixel and at least one N_D neighbor marked as "X" is foreground pixel.

Note: the value of the pixel marked as "Y" is not controlled.

	Y	1	X		X	1	Y
(a)	1	1	X		X	1	1
	X	X	0		0	X	X

	X	X	0		0	X	X
(b)	1	1	X		X	1	1
	Y	1	X		X	1	Y

Figure 8. Templates

After all pixels are processed, the whole process is repeated using the second set of templates (Fig. 8(b)).

IV. Experimental Results

Four experiments were used to evaluate the properties of the proposed method. In each experiment the ZS and the KWK algorithms were used for comparison. In the first two experiments the problem of excessive erosion was tested. There are two main issues: excessive erosion of diagonal line segments and complete removal of patterns which can be reduced to 2x2 squares. In both cases the ZS algorithm produces inaccurate skeleton. Our method is able to preserve the patterns and produce precise skeleton. The KWK algorithm also preserves diagonal lines but remove whole 2x2 square patterns. The results are shown in Fig.9 and 10.

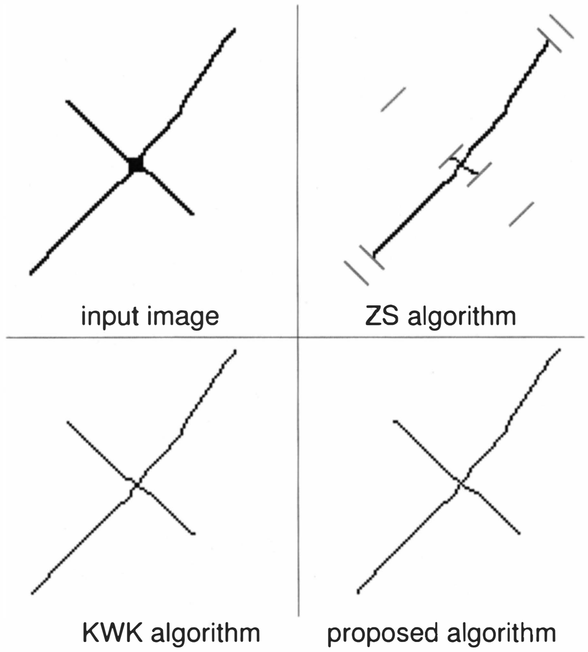

Figure 9. Excessive erosion

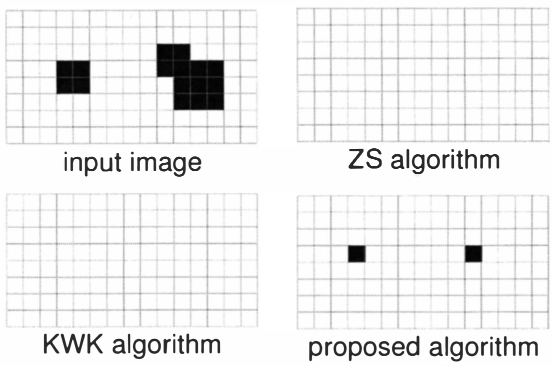

Figure 10. Deletion of patterns which can be reduced to 2x2 squares

One of the most frequently discussed advantages of the ZS algorithm is its insensitivity to boundary noise. The KWK algorithm uses the modified condition (7) to prevent the deletion of diagonal line segments. This is conserving the crucial pixels but also the pixels that should be deleted. These pixels can cause the creation of spurious lines and end points. The proposed method uses original conditions and preserves diagonal lines by recognizing the particular pattern. This ensures that only redundant pixels are removed. As the result, the low sensitivity to boundary noise remains (Fig. 11 and 12).

Although the KWK algorithm removes majority of redundant pixels, the skeleton is still not one pixel thick (Fig.13). This is not the case of the proposed method where all redundant pixels are removed. The only pixels which are left and can be considered as redundant by some metrics, including the thinning rate measurement described below, are the pixels which form the T-junctions. These pixels are not removed on purpose, because they are important to define more precisely the position of these junctions in skeleton.

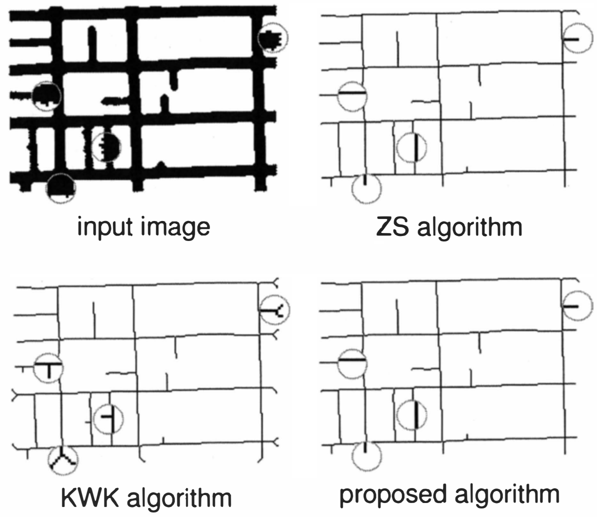

Figure 11. Sensitivity to boundary noise: transportation infrastructure

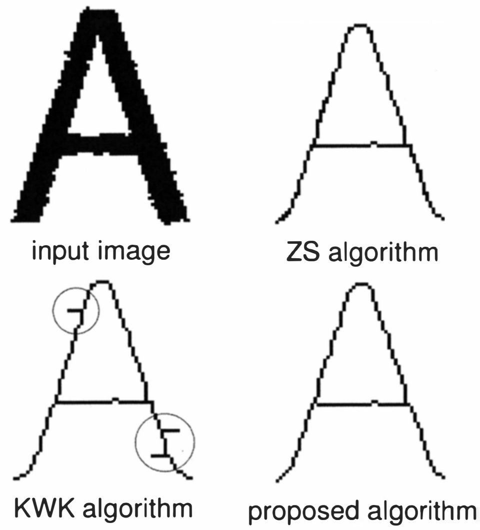

Figure 12. Sensitivity to boundary noise: character "A"

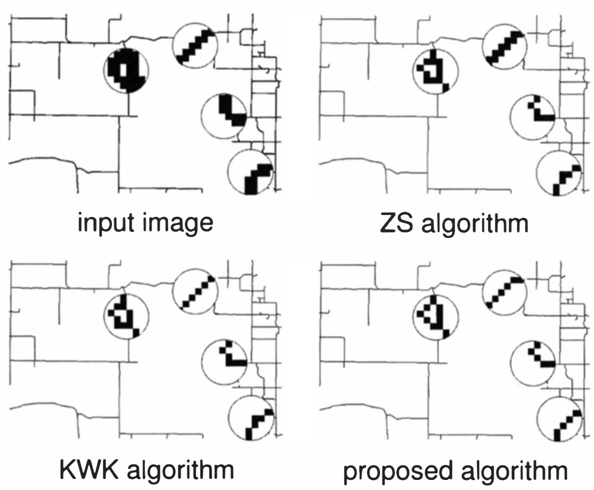

Figure 13. Redundant pixels in ZS and KWK skeletons

To estimate the thickness of the skeleton, thinning rate *TR* introduced in [16] was computed using the following formula:

$$TR = 1 - \frac{TTC_T}{TTC_O} \tag{17}$$

Where TTC_T and TTC_O stand for total triangle count in thinned and in original (input) image respectively. If TR=1 the image can be considered as completely thinned and if TR=0 the image is not thinned at all. The results for Figs.11-13 shown in Table 1 confirm that the proposed method produces the thinnest skeleton.

TABLE I.
RESULTS OF THINNING RATE MEASUREMENT

	ZS algorithm	**KWK algorithm**	**proposed algorithm**
Fig.11	0,9989	0,9989	0,9992
Fig.12	0,9930	0,9979	0,9995
Fig.13	0,9853	0,9902	0,9916

V. CONCLUSIONS

In this paper a robust parallel thinning algorithm is presented. It preserves connectivity and is insensitive to boundary noise. The algorithm also prevents from deletion of diagonal line segments and 2x2 squares and produces unit width skeleton which are the main drawbacks of the ZS algorithm. The execution times of the proposed method and the ZS algorithm were approximately the same in all experiments described in section 4. We believe that the proposed method has potential to improve the accuracy of existing applications which use thinning for shape analysis and recognition tasks such as character and handwriting recognition, medical image analysis and fingerprint recognition. The modified method is already used in application for automatic recognition of infrastructure from drawn maps.

ACKNOWLEDGMENT

This work has been supported by grant VEGA 1/0296/12.

REFERENCES

[1] J. Sadri, C.Y. Suen, and T.D. Bui, "Automatic segmentation of unconstrained handwritten numeral strings," in *Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR 2004)*, 2004, pp. 317–322.

[2] M.S. Khorsheed, "Recognising Handwritten Arabic Manuscripts Using a Single Hidden Markov Model," *Pattern Recognition Letters*, vol. 24, pp. 2235-2242, 2003.

[3] J.-Y. Cheng, and Y.-H. Liu, "Human body image segmentation based on wavelet analysis and active contour models," in *International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR '07)*, 2007, pp. 265 – 269.

[4] R. Xia, P. Wang, and Q. Zhao, "An Image Segmentation Method for Quasi-circular Immune Cells," in *International Symposium on Intelligence Information Processing and Trusted Computing (IPTC)*, 2010, pp. 353 – 356.

[5] P. Cenek, and P. Tarabek, "Acquisition of Input Data for Transportation Models," in *7th EUROSIM congress on modelling and simulation*, 2010.

[6] H. Xu, Y. Qu, and F. Zhao, "FPGA Based Parallel Thinning for Binary Fingerprint Image," in *IEEE Chinese Conference on Pattern Recognition*, 2009, pp. 1–4.

[7] L. Lam, S. Lee, and C. Suen, "Thinning Methodologies—A Comprehensive Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 869-885, 1992.

[8] L. Lam, and C. Suen, "An Evaluation of Parallel Thinning Algorithms for Character Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 9, pp. 914-919, 1995.

[9] T.Y. Zhang, and C.Y. Suen, "A Fast Parallel Algorithm for Thinning Digital Patterns," *Commun. ACM*, vol. 27, pp. 236-239, 1984.

[10] L.E. Lu, and P.S.P. Wang, "A Comment on 'A Fast Parallel Algorithm for Thinning Digital Patterns'," *Comm. ACM*, vol. 29, pp. 239-242, 1986.

[11] P. S. P. Wang, L. Hui, and Jr. T. Fleming, "Further improved fast parallel thinning algorithm for digital Patterns," in *Computer Vision, Image Processing and Communication Systems and Applications*, pp. 37-40, 1986.

[12] J.S. Kwon, J.W. Gi, and E.K. Kang, "An enhanced thinning algorithm using parallel processing," in *International Conference on Image Processing*, 2001, pp. 752 -755.

[13] X. D. Gu, D. H. Yu, and L. M. Zhang, "Image thinning using pulse coupled neural network," *Pattern Recognit. Lett.*, vol. 25, no. 9, pp. 1075–1084, Jul. 2004.

[14] S. K. Pal, "Fuzzy sets in image processing and recognition," in *IEEE Int. Conf. on Fuzzy Syst.*, San Diego, 1992, pp. 119-126.

[15] R. Krishnapuram and J. M. Keller, "Fuzzy set theoretic approach to computer vision: An overview," in *IEEE Int. Conf. on Fuzzy Systems*, San Diego, 1992, pp. 135-144.

[16] P. Tarabek, "Performance measurements of thinning algorithms," *Journal of Information, Control and Management Systems*, vol. 6, pp. 125-132, 2008.