An Improved Adaptive Thinning Framework

Jun Ma, Xunhuan Ren, Yuan Liu, Viktar Yurevich Tsviatkou, Valery Kanstantinavich Kanapelka

***Abstract*—The thinning algorithm is one of the fastest approaches to extract skeletons from an object, especially when adopting the parallel strategy. Skeletons are very useful descriptors and can be applied in many recognition fields. However, one of the drawbacks that limits the use of these techniques is that thinning algorithms are not robust against inner noise and outer noise, which may produce many unwanted branches. To alleviate the influence of noise and increase the robustness, pruning methods and scale-space methods have been proposed in the past, in which pruning methods are aimed at suppressing the outer noise (boundary noise) and scale-space methods are aimed at suppressing the inner noise (such as scratch noise and dithering noise). In this paper, we proposed an improved framework that can deal with both inner noise and outer noise. The experiment proved that the proposed framework has better visual effects than the existing pruning method and existing scale-space method. In addition, the proposed framework is an adaptive framework that does not require manual tuning of parameters.**

***Index Terms*—Adaptive framework, Thinning algorithm, Robustness against noise, pruning algorithm, scale-space filtering.**

I. Introduction

Thinning algorithms are classical digital approaches of skeletonization methods [1] used in many different fields of pattern recognition, such as biometric authentication using retinal images [2], fingerprint identification [3-5], and sketch matching [6]. Thinning algorithms have been used to extract skeletons (also named medial axes) from given patterns. The extracted skeletons are important descriptors because they provide a low-dimensional and intuitive shape representation. Thinning methods can be classified into iterative algorithms and noniterative algorithms [7]. Iterative algorithms can be further divided into parallel and sequential algorithms in terms of their computational efficiency [8, 9]. Desirable thinning algorithms should have some of the following properties: they should preserve the original pattern's connectivity, maintain its visual topology, and be robust against noise [10]. Although many thinning algorithms have been proposed in the past and have satisfying requirements regarding connectivity and topology, they tend to be sensitive to noise [7,10-14].

There are two major noise types: outer noise (border noise) and inner noise (including scratch noise and dithering noise) [15]. All these noises greatly alter the resulting skeleton extracted by the thinning algorithm, which may enhance the difficulties of the later recognition operation.

To reduce the interference of outer noise, the most popular type of method is based on skeleton pruning, and many pruning methods have been proposed in the past [16-20]. These methods can dramatically alleviate the influence of border noise and avoid producing redundant skeleton branches. However, these methods fail to suppress inner noise.

Another type of method uses scale space to make thinning algorithms robust against noise [21-23]. These methods can offset the effects of all types of noise. However, these methods fail to remove all the redundant skeletal branches and sometimes may violate the connectivity and topology of the original pattern.

In this paper, we propose a novel framework that combines the merits of the pruning method and methods based on scale space. The proposed framework can offset the influence of all types of noise but also maintain the connectivity and topology of the original patterns. In addition, the proposed framework has the property of automatic parameter tuning.

The remainder of this paper is organized as follows: Some pruning algorithms and scale-space methods are briefly reviewed in Section 2. Then, the proposed framework is described in Section 3. Next, experiments are conducted in Section 4, and the results are discussed. Finally, the conclusion and future work are given in Section 5.

II. Related Works

A. Pruning Methods

The purpose of the pruning methods is to delete unwanted skeleton branches caused by the boundary noise. They are generally applied after skeletonization. The core problem of the pruning method is finding a suitable saliency measurement for evaluating skeletal points or branches. The skeletal branches or points whose saliency measurement is above the given threshold will be preserved, and others will be removed.

The most cited pruning method is the discrete curve evolution (DCE) pruning algorithm [16], which was proposed by Xiang Bai and his team. The DCE method first conducts contour partitioning and then removes all the skeleton points whose corresponding boundary points lie on the same contour segment. Generally, the DCE method requires manually tuning the parameter according to the input images, which costs much effort; therefore, an automatic

Manuscript received Feb 25, 2022; revised Jul 22, 2022.

Jun Ma is a postgraduate student at Belarusian State University of Informatics and Electronics, Minsk, Belarus. (e-mail: majun1313@hotmail.com).

Xunhuan Ren is a postgraduate student at Belarusian State University of Informatics and Electronics, Minsk, Belarus. e-mail: rxh1549417024@gmail.com).

Yuan Liu is a postgraduate student at Belarusian State University of Informatics and Electronics, Minsk, Belarus. (e-mail:1359799048@qq.com).

Viktar Yurevich Tsviatkou is a Professor in Department of Info-communication technology, Belarusian State University of Informatics and Electronics, Minsk, Belarus. (e-mail: vtsvet@bsuir.by).

Valery Kanstantinavich Kanapelka is a Professor in Department of Info-communication technology, Belarusian State University of Informatics and Electronics, Minsk, Belarus. (e-mail: volos@bsuir.by).

DCE framework was recently proposed [17]. The advantage of the DCE is that it maintains the skeleton topology, does not shift the original skeleton, and does not shrink the remaining branches. However, the obtained skeletons may contain some redundant points and some unimportant branches.

To overcome the mentioned drawbacks and obtain a better skeleton, Wei Shen proposed a pruning method based on the bending potential ratio (BPR) [18], in which the deletion decision of a skeletal branch depends on the context of the boundary segment that corresponds to the branch. As a saliency measure, BPR can better evaluate the contribution of the boundary segment to the overall shape. The skeleton obtained by their pruning method was proven to be medially placed and insensitive to boundary noise.

Another interesting pruning method is based on information fusion, which was proposed by Liu et al. [19]. In their opinion, different objective measurements have different advantages and limitations. Therefore, they proposed skeleton pruning based on various measurements of branch significance, including region reconstruction, contour reconstruction and visual contribution. The merits of their method are that they are stable and robust to boundary noise and can effectively generate multiscale skeletons according to visual judgment.

Recently, a skeleton pruning method based on saliency sort [20] was proposed by Guo et al. Their method aims to overcome the flaw of the saliency measures of existing pruning methods, which are not intuitive and have difficulty finding a suitable threshold. The results proved that their method is simple for conducting manual tuning and can generate a satisfying skeleton.

B. Scale-Space Method

The scale-space method generally conducts the noise filtering operation on the original pattern by using a Gaussian filter before skeletonization to degrade the influence of the noise.

The advantage of the scale-space method is that it can deal with all three types of noise to some extent. The anti-noise ability of the scale-space methods depends on the value of the smoothing parameter σ, which is used in the Gaussian filter. A larger value of σ denotes that it can filter out larger noises, but the original pattern may suffer deformation, which may cause a change in the partial topology of the skeleton. Therefore, existing scale-space methods employ various judgment conditions to find a suitable σ, by which most of the noise can be filtered out and the topology of the original pattern can be maintained.

Hoffman and Wong proposed a scale-space method [21] for thinning binary and grayscale images. Their method first yields filtered versions of an image and then extracts the union of topologically significant points, which include peak, ridge and saddle points. The skeleton is formed by these extracted pixels, which are also named “The Most Prominent Ridge-Line pixels (MPRL)”. The sensitivity of the skeleton to noise strongly depends on the parameter manually set by the user.

Cai presented a scale-space method based on oriented Gaussian filters [22] to decrease the influence of noise caused by pen perturbations and scanning of documents and images, during thinning of handwriting and fingerprint images. In his method, all the pixels are first classified into edges, valleys and ridges by using oriented Gaussian filters, and then noise points are removed by trimming negative parts of ridge energy images. Finally, skeletons can be extracted based on the smooth intensity surfaces of ridge energy images and principal directions.

Houssem Chatbri and Keisuke Kameyama proposed an adaptive thinning framework (ATF) [23] to make thinning algorithms robust against noise in sketch images. Their framework applied scale-space filtering to produce multiple representations of an original pattern within multiple scales. Then, their framework estimates the optimal filtering scale automatically according to the value of the sensitivity measure conducted in their framework. In addition, the authors also note that any thinning algorithm can be used in their framework. Experiments have proven that their framework is robust against different types of noise that exist in sketch images.

C. Summary

Both the pruning method and the scale-space method can suppress the influence of noise and help to produce a clean skeleton. However, they have different advantages and disadvantages and are applied at different stages. The pruning methods are generally applied after skeletonization, whereas the scale-space methods are applied before skeletonization. Therefore, it is very promising to propose a framework that combines the merits of the two methods by introducing both a pruning procedure and a scale-space procedure.

III. PROPOSED FRAMEWORK

The proposed framework has 2 main procedures: The scale-space procedure and the post -pruning procedure. The scale-space procedure automatically determines a suitable smooth parameter σ that can filter out the inner noise and avoid significant deformation. This procedure is the modified version of the adaptive thinning framework [23] mentioned in the previous section. The post-pruning procedure is an automatic pruning method based on the DCE pruning method that computes the relative reconstruction increment between skeletons to select a suitable parameter to control the pruning power.

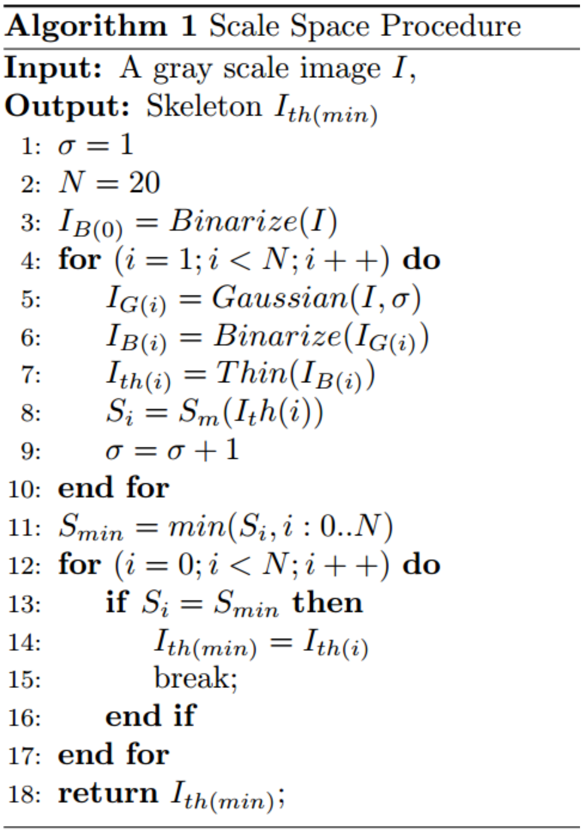
A. Scale-Space Procedure

The core of the scale-space procedure is based on the Gaussian filter, which is defined as:

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}e^{-\frac{(x+y)^2}{2\sigma^2}} \tag{1}$$

where σ is the smooth parameter that controls the scale, and x and y are pixel coordinates.

The proposed procedure works as follows: First, we blur the input image I and obtain a grayscale image I_G by using a Gaussian filter of scale σ_i. Then, I_G is binarized, and a binary image I_B is obtained. Next, we use an embedded thinning algorithm to extract the skeleton I_{th} from I_B. Finally, a sensitivity measure S_i is calculated on skeleton I_{th}. The Gaussian scale σ_i is then increased, and the entire process is repeated. The output of the scale-space procedure is a skeleton image having the minimum value of the sensitivity measure, which is denoted by $I_{th(min)}$. The pseudocode of this procedure is presented in Algorithm 1.



```
Algorithm 1 Scale Space Procedure
Input: A gray scale image $I$,
Output: Skeleton $I_{th(min)}$
$\sigma = 1$
$N = 20$
$I_{B(0)} = Binarize(I)$
for $(i = 1; i < N; i++)$ do
    $I_{G(i)} = Gaussian(I, \sigma)$
    $I_{B(i)} = Binarize(I_{G(i)})$
    $I_{th(i)} = Thin(I_{B(i)})$
    $S_i = S_m(I_th(i))$
    $\sigma = \sigma + 1$
end for
$S_{min} = min(S_i, i : 0..N)$
for $(i = 0; i < N; i++)$ do
    if $S_i = S_{min}$ then
        $I_{th(min)} = I_{th(i)}$
        break;
    end if
end for
return $I_{th(min)}$;
```

The sensitivity measure is used to evaluate the skeleton and select the minimum thinned image from the scale space. We modify the original sensitivity measure used in [24].

$$S_{\mathrm{m}}(\mathrm{I}_{th}) = \frac{1}{n}\sum_{i=1}^{N}\sum_{j=1}^{M} S_1(\mathrm{i, j}) \qquad (2)$$

where:

$$S_1(\mathrm{i, j}) = \begin{cases} 1,\ T_{BW}(i, j) > 2 \text{ OR } (I(i,j)=0) \\ 5,\ F_{BD}(i,j) = 1 \text{ AND } \ T_{BW}(i, j) > 1 \\ 10,\ \dfrac{|Area(I_{B(i)})\text{-}Area(I_{B0})|}{Area(I_{B0})} > 0.02 \\ \qquad \text{OR Region}(I_{B(i)}) \neq \text{Region}(I_{B0}) \\ 0,\ \text{Otherwise} \end{cases} \qquad (3)$$

Here, n is the number of foreground pixels in I_{th}. N and M are the number of rows and columns of the image I_{th}. T_{BW} is the number of transitions from foreground pixels to background pixels in the 8-neighborhood window of (i, j) (see Fig. 1), which is defined as follows:

$$T_{BW}(i, j) = \sum_{k=1}^{8} transition(\mathrm{P}_k) \qquad (4)$$

where:

$$transition(P_k) = \begin{cases} 1,\ P_k = 1 AND \ \ P_{(k+1)\mathrm{mod}8} = 0 \\ 0, \text{Otherwise} \end{cases} \qquad (5)$$

Formula (3) is applied to all the foreground pixels in the skeleton, and the first condition and second condition are used to penalize those foreground pixels of the skeletal image, which are introduced by the filters rather than existing in the original pattern. The difference between them is that the second condition is only aimed at penalizing the skeletal points that are outside the edge of the pattern. Here, F_{BD} is an image whose foreground pixels consist of the background pixels that are not surrounded by the foreground pixels in $I_{B(0)}$.

P_8	P_1	P_2
P_7	P_0	P_3
P_6	P_5	P_4

Fig.1 Definition of the 8-neighborhood window

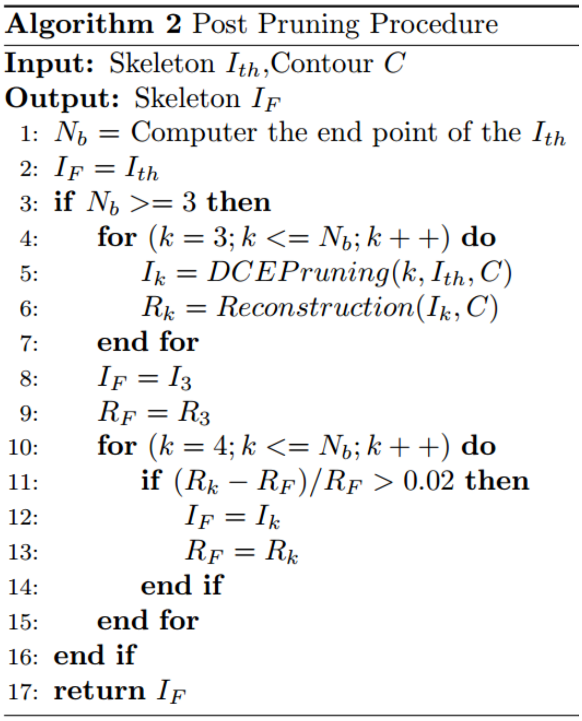
The third condition is used to maintain that the topology of a filtered binary image is the same as that of the original image, so it is a global condition. The former part of the condition is used to limit the dramatic deformation of the pattern caused by a larger smooth parameter because a larger deformation may cause an alteration of the topology. If there is a larger difference between the filtered image and the original image (above 2%), a penalty needs to be introduced. The latter part of the condition is to compare the number of connected components constituted by the foreground pixels between the filtered binary image and the original binary image. Because it is easy to consider that if the number of connected components of two images is different, their topology is also unequal. Here, Area() is a function that counts all the foreground pixels in a given binary image. Region() is a function that counts the total number of distinct connected components.

The output of the scale-space procedure is a skeletal image that has the minimal value of $S_m(I_{th})$.

B. *Post-pruning Procedure*

The post-pruning procedure is a modification of the DCE pruning method that introduces the concept of a relative reconstruction increment to enable the automatic selection of a reasonable pruning strength. For convenience, the details of the implementation of DCE pruning are not covered in this section. Here, it is only considered as a black box whose inputs are the pruning strength k, skeleton I_{th}, and the corresponding contour C and the output is a pruned skeleton I_f.

The proposed procedure works as follows: First, it is necessary to compute the number of skeleton branches of the original skeleton and denote it as N_b. The DCE pruning algorithm is only called when the number of skeletal branches is not less than three. Here, let us assume that N_b is greater than three. Then, we call the DCE algorithm with different pruning parameters, which will increment from 3 to N_b, and a series of various pruned skeletons is obtained. These various skeletons reconstruct patterns with different areas. It is obvious that the area of the patterns constructed by the skeleton with the minimum number of branches is the smallest, whereas that of the skeleton with N_b number of branches is the largest. The pattern reconstructed by the skeleton with the minimum branches is named the initial pattern, and this skeleton is named the initial skeleton. Except for the initial skeleton, for each later skeleton, it is necessary to calculate the differences in the area between the patterns reconstructed by the current skeletons and those reconstructed by the former skeletons. This area difference is called the reconstruction increment of the skeleton in this paper. In addition, the relative reconstruction increment is a value that divides the reconstruction increment by the current area of reconstruction. Finally, the default result skeleton is set as the initial skeleton and compared with the later skeleton. If the relative reconstruction increment is above a given value (here, we set it as 0.02), then we will update the result skeleton with the latter skeleton and continue conducting the comparison. Otherwise, we only maintain the result skeleton and continue conducting the comparison. The pseudocode of this procedure is presented in Algorithm 2.



```
Algorithm 2 Post Pruning Procedure
Input: Skeleton $I_{th}$,Contour $C$
Output: Skeleton $I_F$
$N_b$ = Computer the end point of the $I_{th}$
$I_F = I_{th}$
if $N_b >= 3$ then
    for $(k = 3; k <= N_b; k++)$ do
        $I_k = DCEPruning(k, I_{th}, C)$
        $R_k = Reconstruction(I_k, C)$
    end for
    $I_F = I_3$
    $R_F = R_3$
    for $(k = 4; k <= N_b; k++)$ do
        if $(R_k - R_F)/R_F > 0.02$ then
            $I_F = I_k$
            $R_F = R_k$
        end if
    end for
end if
return $I_F$
```

IV. Experiments And Results

A. Experimental Platform

1. Hardware

The computations were conducted on a standard PC laptop with a Core i7-4720Q CPU (2.6 GHz) and 16 GB memory.

2. Software

All experiments were performed in MATLAB R2017b. The operating system of the PC was Windows 8.1.

B. Basic Information of Experiments

1. Dataset

To objectively evaluate the performance of the proposed framework, all the test images are obtained from the well-known dataset MPEG7, which includes 70 different classes, each class with 20 images.

2. Implemented Thinning Methods for the Experiments

The experiments were conducted using four latest thinning algorithms: the SIPS method [12], MZS method [14], IOPTA method [13] and FPSA method [11].

The SIPS method conducts thinning of binary images by repeating two sub-iterations. In each sub-iteration, the candidate pixels that match the deletion templates or logical conditions but do not match the restoring templates are removed.

The MZS method is a parallel thinning algorithm that combines the property of the sub-iterative approach and subfield approach. The removal of the candidate pixels not only depends on the predefined logical conditions but also depends on the counts of the past iterations.

The IOPTA thinning method is a one-pass thinning method in which eight elimination templates and six restoring templates are used. The merit of the one-pass thinning method is that it requires fewer iterations than other methods to extract skeletons from binary images.

FPSA is the latest one-pass thinning method. It introduces a series of restoring templates, compulsory deletion templates, and extra deletion templates to remove the pixels layer-by-layer. In addition, the FPSA has proven to have good performance in terms of (8,4) connectivity preservation.

3. Evaluation measures

To evaluate the performance of the topology preservation of the proposed framework, we calculate the measures T_1 [25] and T_2 [26]. The definitions of T_1 and T_2 are shown in Formulas (6) and (7).

$$T_1 = \frac{\mathrm{Area}(I_{MD})}{Area(I)} \tag{6}$$

I_{MD} is the image reconstructed from the skeleton by referring to the max disc of the skeleton point. Area() is a function that counts all foreground pixels in a given binary image, which was introduced in the previous section. I is the original image. The value of T_1 is between 0 and 1. Larger values denote that the result well preserves the original topology, whereas small values near 0 express that there is significant distortion.

$$T_2 = 1 - \left| \frac{1}{2} - \frac{N_{th}}{N_c} \right| \tag{7}$$

N_{th} is the number of skeletal pixels, and N_c is the number of contour pixels in the original image. The idea behind this measure is that the original binary image generally has a relatively smooth contour, whose number of foreground pixels may be roughly twice the number of foreground pixels in the skeleton. To make the values limited in the range of 0 to 1, the measures are normalized. The T_2 value and distortion are negatively correlated, whereas the T_2 value and topology preservation are positively correlated.

In addition, we count the total number of end-points (EPs) and cross-points (CPs) in the final skeleton for convenience of comparison.

4. Introduce of the Experiment

We conducted two experiments to illustrate the performance of the proposed framework.

The first experiment aims to analyze the effects by using different pruning methods; therefore, we compare the results of the pruned skeleton processed by the automatic DCE method [17], the adaptive thinning framework [23], and the proposed framework. In the first experiment, the embedded thinning algorithm is the FPSA thinning algorithm.

The second experiment investigates the improvement degree of the robustness for different thinning algorithms caused by introducing the proposed framework.

C. First Experiment.

Figs. 2 to 6 present the results by using different pruning methods for different images. Note that the original pattern and skeleton are colored gray and black, respectively.

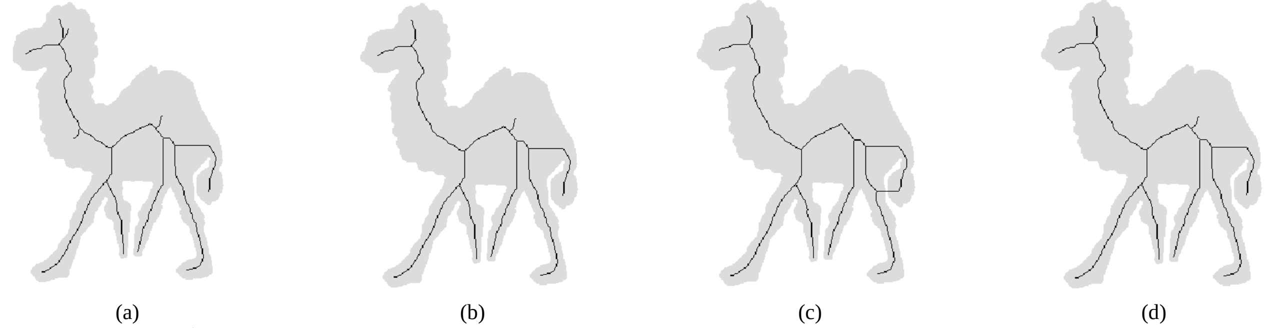


Fig. 2. Resulting skeleton from Camel-17: (a) Original skeleton directly extracted by FPSA; (b) pruned skeleton by using the automatic DCE method;(c) pruned skeleton by using the adaptive thinning framework; (d) pruned skeleton by using the proposed framework.

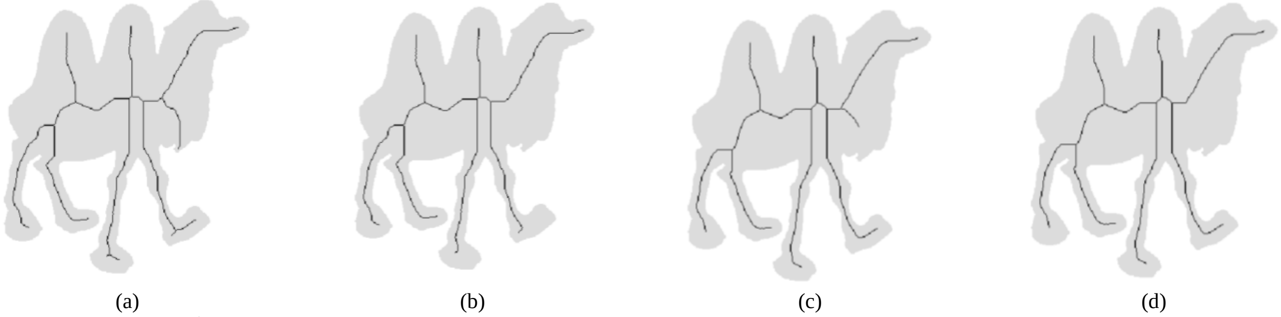


Fig. 3. Resulting skeleton from Camel-18: (a) Original skeleton directly extracted by FPSA; (b) pruned skeleton by using automatic DCE method; (c) pruned skeleton by using the adaptive thinning framework; (d) pruned skeleton by using the proposed framework.

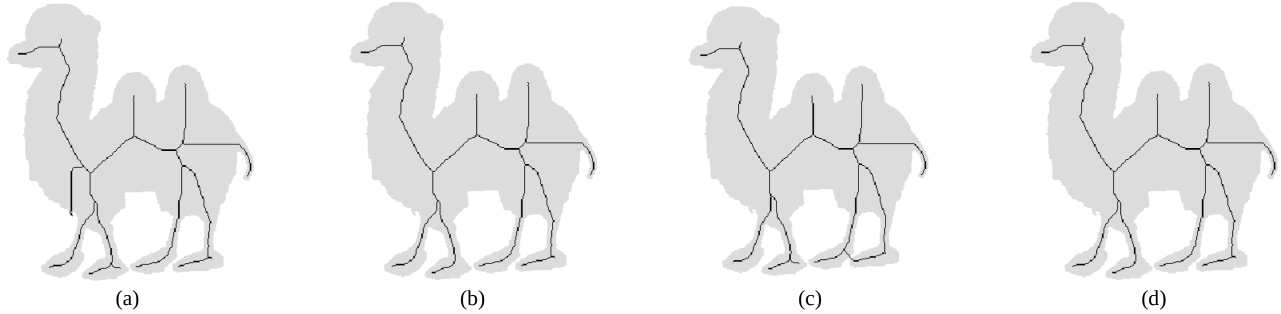


Fig.4. Resulting skeleton from Camel-8: (a) Original skeleton directly extracted by FPSA; (b) pruned skeleton by using the automatic DCE method; (c) Pruned skeleton by using the adaptive thinning framework; (d) pruned skeleton by using the proposed framework.

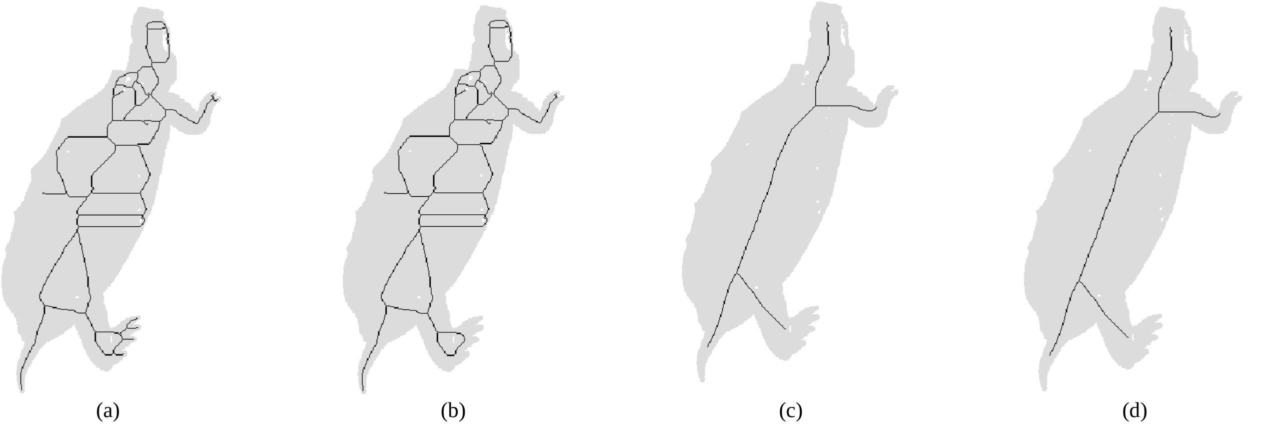


Fig.5. Resulting skeleton from Turtle-12: (a) Original skeleton directly extracted by FPSA; (b) pruned skeleton by the using automatic DCE method;(c) pruned skeleton by using the adaptive thinning framework; (d) pruned skeleton by using the proposed framework.

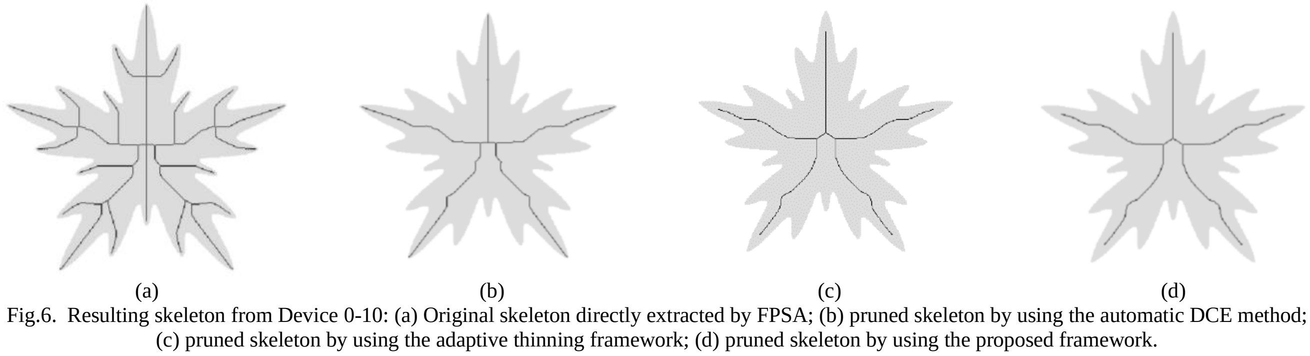
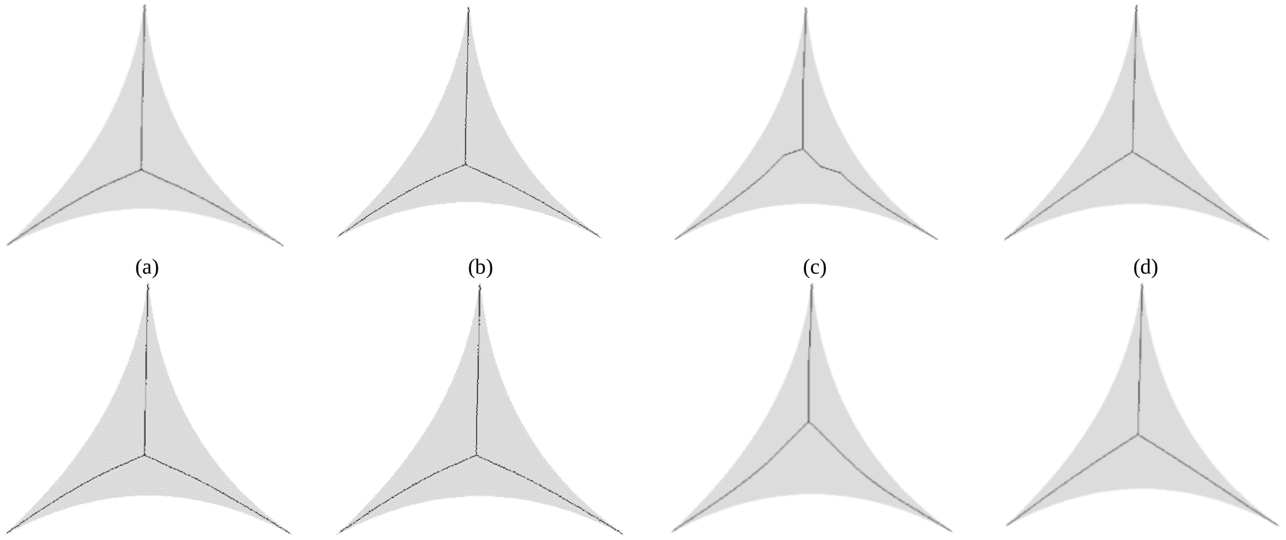


Fig.6. Resulting skeleton from Device 0-10: (a) Original skeleton directly extracted by FPSA; (b) pruned skeleton by using the automatic DCE method; (c) pruned skeleton by using the adaptive thinning framework; (d) pruned skeleton by using the proposed framework.

From the perspective of the visual effects, it is obvious that the method based on automatic DCE outperformed the adaptive thinning framework in two aspects. The first is that the method based on the automatic DCE method can completely remove the unwanted branches (see Fig. 2(b) and Fig. 3(b)). In contrast, the adaptive thinning framework sometimes may only shorten the unnecessary branches (see Fig. 3(c)). In addition, the method based on automatic DCE can maintain the original skeleton connectivity, which may alter the adaptive thinning framework. Fig. 2(c) is a good example. It is obvious that the tail of the camel and its leg compose a ring, which alters the original structure. However, in terms of the inner noise, the adaptive thinning framework has a better result than the automatic DCE method. In Fig. 5, it is easy to observe that the skeleton in (c) is more concise than that in (b). The proposed framework combines the advantages of these two methods. From Figs. 2 to 6, it can be seen the proposed framework can well preserve the original connectivity, totally delete the unwanted branches, and suppress the inner noise. Table 1 presents a quantitative comparison of these images. We can see that compared with the pruned skeletons, the original skeletons have more EP and CP, so their T1 and T2 are closer to 1. The function of the pruning method or framework is to remove unwanted branches, so in the pruned skeletons, all the parameters are decreased to different degrees.

TABLE I
QUANTITATIVE COMPARISON OF DIFFERENT IMAGES

	Parameter	Original Skeleton	Pruned Skeleton using DCE	Pruned Skeleton using ATF	Pruned Skeleton using the proposed Method
Camel-17	EP	10	8	6	8
	CP	8	6	6	6
	T_1	0.8859	0.8761	0.8520	0.8756
	T_2	0.7853	0.7762	0.7775	0.7779
Camel-18	EP	10	7	8	7
	CP	8	5	6	5
	T_1	0.9030	0.8835	0.8739	0.8738
	T_2	0.8370	0.8073	0.8111	0.8035
Camel-8	EP	12	10	10	10
	CP	10	8	10	8
	T_1	0.9138	0.9103	0.9104	0.9113
	T_2	0.8301	0.8121	0.8130	0.8076
Turtle-12	EP	7	2	4	4
	CP	47	44	2	2
	T_1	0.9273	0.9189	0.6402	0.6403
	T_2	0.9091	0.9321	0.6808	0.6808
Device0-10	EP	20	5	5	5
	CP	22	7	3	3
	T_1	0.9616	0.8160	0.7861	0.7965
	T_2	0.9498	0.7164	0.6827	0.6847



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 7. Resulting skeleton from Device4-11: (a) Skeleton extracted by SIPS; (b) skeleton extracted by MZS;(c) skeleton extracted by IOPTA; (d) skeleton extracted by FPSA; (e) skeleton extracted by SIPS with the proposed framework; (f) skeleton extracted by MZS with the proposed framework; (g) skeleton extracted by IOPTA with the proposed framework; (h) skeleton extracted by FPSA with the proposed framework.

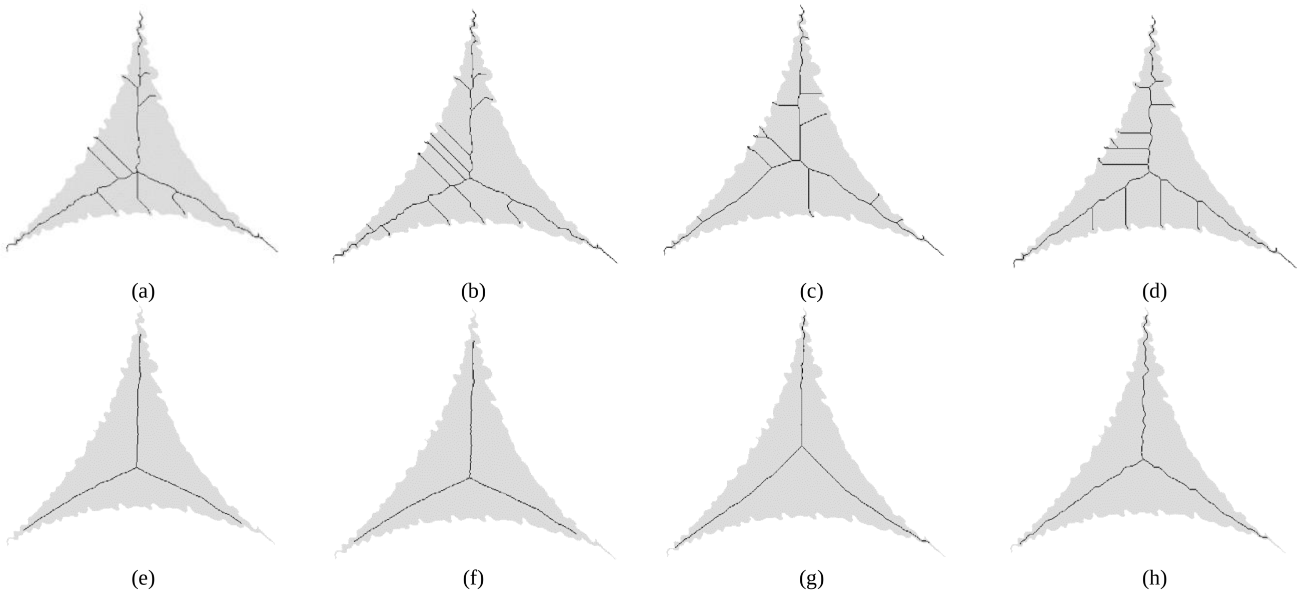


Fig. 8. Resulting skeleton from Device4-12: (a) Skeleton extracted by SIPS; (b) skeleton extracted by MZS; (c) skeleton extracted by IOPTA; (d) skeleton extracted by FPSA; (e) skeleton extracted by SIPS with the proposed framework; (f) skeleton extracted by MZS with the proposed framework; (g) skeleton extracted by IOPTA with the proposed framework; (h) skeleton extracted by FPSA with the proposed framework.

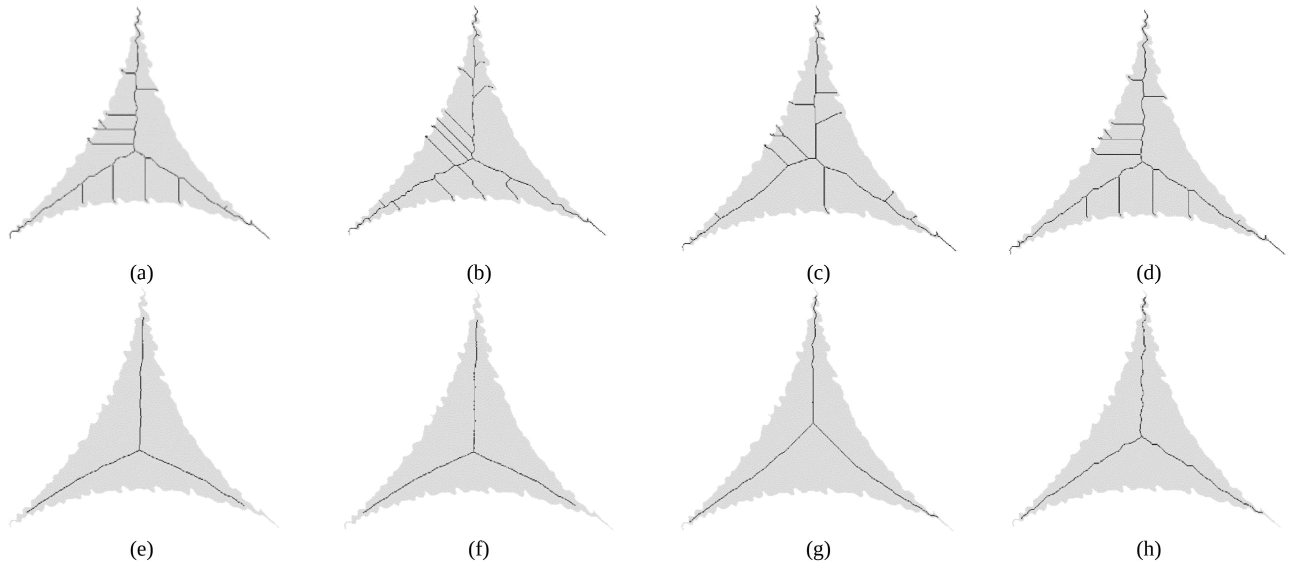


Fig. 9. Resulting skeleton from Device4-17: (a) Skeleton extracted by SIPS; (b) skeleton extracted by MZS; (c) skeleton extracted by IOPTA; (d) skeleton extracted by FPSA; (e) skeleton extracted by SIPS with the proposed framework; (f) skeleton extracted by MZS with the proposed framework; (g) skeleton extracted by IOPTA with the proposed framework; (h) skeleton extracted by FPSA with the proposed framework.

D. Second Experiment.

Figs. 7-9 show the results of four different thinning algorithms when used directly and when plugged into the proposed framework. From Fig. 7, in which the input image is smooth and without any noise, it can be seen there is no prominent difference between the results when using the thinning algorithms alone and when they are embedded in the proposed framework. All of them can produce relatively satisfying skeletons. However, in Fig. 8 and Fig. 9, there are many boundary noises in the input image, and the skeletons processed by the thinning algorithms combined with the framework have fewer skeletal branches than those processed by using the thinning algorithms alone. In addition, it is obvious that the proposed framework can improve the stability of the extracted skeleton if we plug the thinning algorithm into the proposed framework. Therefore, the proposed framework can effectively offset the influence caused by the noise.

V. Conclusion And Future Works

In this paper, we proposed a novel framework for enhancing the robustness of thinning algorithms. Our framework consists of a scale-space procedure and a post-pruning procedure. The scale-space procedure is used to suppress the inner noise, and the post-pruning procedure enhances the robustness to the boundary noise. Therefore, the proposed framework can deal with different types of noise. The first experiment proved that the proposed framework has better performance than existing pruning methods. In addition, the second experiment proved that any thinning algorithm could be plugged into the proposed framework to improve the robustness. Furthermore, the proposed framework is an adaptive framework without manual tuning, so it is very convenient for different applications.

In the future, we would like to apply our framework in parallel devices such as GPUs and FPGAs to achieve a satisfying processing speed.

References

[1] P. K. Saha, G. Borgefors, and G. Sanniti di Baja, “A survey on skeletonization algorithms and their applications,” Pattern Recognition Letters, vol. 76, pp. 3-12, 2016.

[2] C. G. Owen and S. A. Barman, “Blood vessel segmentation methodologies in retinal images-A survey,” Computer Methods Programs Biomed, vol. 108, no. 1, pp. 407-433, 2012.

[3] M. Fons, F. Fonsaa and E. Canto, “Fingerprint Image Processing Acceleration Through Run-Time Reconfigurable Hardware,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 57, no. 12, pp. 991-995, Dec. 2010.

[4] M. Sepasian, W. Balachandran, and C. Mares, "Image enhancement for fingerprint minutiae-based algorithms using CLAHE, standard deviation analysis and sliding neighborhood," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2008, WCECS 2008, 22-24 October, 2008, San Francisco, USA, pp. 22-24.

[5] I. G. Babatunde, A. O. Charles, A. B. Kayode and O. Olatubosun, "A multi-level model for fingerprint image enhancement, "Lecture Notes in Engineering and Computer Science: Proceedings of The International Multiconference of Engineers and Computer Scientists 2012, IMECS 2012, 14-16 March, 2012, Hong Kong, pp.155-174.

[6] A. Chalechale, G. Naghdy and A. Mertins, “Sketch–based image matching Using Angular partitioning,” IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 35, no. 1, pp. 28-41, Jan. 2005.

[7] Jun Ma, Xun-Huan Ren, and T. V. Yurevich, “A Novel Fast Iterative Parallel Thinning Algorithm,” in Proceedings of the 2020 4th International Conference on Vision, Image and Signal Processing, 2020.

[8] L. Lam, S. W. Lee and C. Y. Suen, “Thinning methodologies–a comprehensive survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 9, pp. 869-885, Sep. 1992.

[9] B. B. Lynda, B. Solaiman, and A. Tari, “Implementation and comparison of binary thinning algorithms on GPU,” Computing 101, pp. 1091-1117, 2019.

[10] P. Tarabek, “A robust parallel thinning algorithm for pattern recognition,” 2012 7th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), pp. 75-79, 2012.

[11] Jun Ma, Xun-Huan Ren, T. V. Yurevich, and V. K. Kanapelka, “A novel fully parallel skeletonization algorithm,” Pattern Analysis Application, 2021.

[12] Jun Ma, Xun-Huan Ren, T. V. Yurevich, and V. K. Kanapelka, “A Novel Sub–Iterative Parallel Skeletonization Method,” Journal of Computers (Taiwan), vol. 32, no. 6, pp. 83-97, 2021.

[13] Rui-Zheng Wang, et al., “Fingerprint Refinement Model Based on Improved OPTA,” Journal of Computers (Taiwan), pp. 274-283, 2020.

[14] L. Ben Boudaoud, B. Solaiman, and A. Tari, “A modified ZS thinning algorithm by a hybrid approach,” Vision Computer, vol. 34, no. 5, pp. 689-706, 2018.

[15] H. Chatbri and K. Kameyama, “Towards making thinning algorithms robust against noise in sketch images,” in Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), 2012, pp. 3030-3033.

[16] X. Bai, L. J. Latecki and W. Liu, “Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 3, pp. 449-462, Mar. 2007.

[17] Jun Ma, Xun-Huan Ren, T. V. Yurevich, and V. K. Kanapelka, “An automatic pruning method for skeleton images”, in Pattern Recognition and Information Processing (PRIP'2021): Proceedings of the 15th International Conference, 2021, pp.232-235.

[18] Shen Wei, Xiang Bai, Rong Hu, Hong-Yuan Wang, and L. J. Latecki, “Skeleton growing and pruning with bending potential ratio,” Pattern Recognition, vol. 44, no. 2, pp. 196-209, 2011.

[19] Hong-Zhi. Liu, Zhong-Hai. Wu, Xing Zhang, and D. F. Hsu, “A skeleton pruning algorithm based on information fusion,” Pattern Recognition Letters, vol. 34, no. 10, pp. 1138-1145, 2013.

[20] Si-yu Guo, Ping-ping Hu, Zhi-gang Ling, He Wen and Min Liu, “A skeleton pruning method based on saliency sorting,” 2019 14th IEEE International Conference on Electronic Measurement & Instruments *(ICEMI)*, 2019, pp. 593-599.

[21] M. E. Hoffman and E. K. Wong, “Scale-space approach to image thinning using the most prominent ridge-line in the image pyramid data structure,” in Photonics West’98 Electronic Imaging, International Society for Optics and Photonics, 1998.

[22] J. Cai, “Robust Filtering-Based Thinning Algorithm for Pattern Recognition,” The Computer Journal, vol. 55, no. 7, pp. 887-896, 2012.

[23] H. Chatbri and K. Kameyama, “Using scale space filtering to make thinning algorithms robust against noise in sketch images q,” Pattern Recognition, vol. 42, pp. 1-10, 2014.

[24] R. W. Zhou, C. Quek, and G. S. Ng, “A novel single–pass thinning algorithm and an effective set of performance criteria,” Pattern Recognition Letters, vol. 16, no. 12, pp. 1267-1275, 1995.

[25] B. K. Jang and R. T. Chin, “One-pass parallel thinning: analysis, properties, and quantitative evaluation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 11, pp. 1129-1140, Nov. 1992.

[26] J. Dong, W. Lin and C. Huang, “An improved parallel thinning algorithm,” in 2016 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), 2016, pp. 162-167.