

An Improved Cost-Efficient Thinning Algorithm for Digital Image

Liang Jia¹, Zhenjie Hou²

¹ School of Information Science and Engineering, Chang Zhou University,
Changzhou, Jiangsu 213164, China

² School of Information Science and Engineering, Chang Zhou University,
Changzhou, Jiangsu 213164, China

Abstract

Digital image processing plays an important role in our ever-evolving ubiquitous information society with an increasing need of automatic and efficient information collection. Thinning algorithms have been widely developed and applied in Optical Character Recognition (OCR) to eliminate the redundant data as well as keep the essential features of digital images. Inspired by the in-depth analysis of results obtained from Davies's classical algorithm, this paper proposes an improved and cost-effective thinning algorithm to enhance the accuracy of digital image skeletonization and also maintain the computation complexity at a low level. The extensive experiments show that the results of this improved thinning algorithm is able to achieve the similar accuracy of other advanced algorithms and inherits the advantage of low complexity of Davies's classical algorithm.

Keywords: Skeletonization, Thinning Algorithm, Medial Axis, Binary Image Processing, Image Erosion

1. Introduction

Since the skeleton greatly decreases the complexity of the object shapes and thus reduces the number of pixels required for further processing, it facilitates many important applications, such as OCR [2, 3, 4] and Hough-based algorithms [1, 5]. To obtain skeletons, various thinning algorithms have been developed. These algorithms can be coarsely classified into two main categories according to their distinct nature, i.e., distance-transform-based and non-distance-transform-based algorithms. The former is a traditional means to extracting skeletons and the latter is a relatively new rising area in skeletonization, e.g., neural networks [6], wavelet [7] and curvature flow [8], etc. Although the number of non-distance-transform-based algorithms rapidly grows and such algorithms can obtain skeletons in 3-dimensional (3-D) space, they are commonly over complicated for processing an object in 2-D digital image and require considerable computation [9]. Therefore, the improvement of classical distance-transform-based algorithms is still required for 2-D digital image processing.

During the past decades, a great number of the distance-transform-based algorithms are developed. There mainly exist two different categories formed by there algorithms, i.e., the raster-scan-based algorithms and the medial-axis-based algorithms [10]. Davies proposed a well-known raster-scan-based algorithm in 1981 [5]. This classical algorithm does not employ a lookup table to estimate connectedness or additional data structures except the simple 2-D array, and thus it is easy to be analyzed and implemented. However the skeleton generated by this algorithm is nearly useless when the shape of the object in 2-D images becomes complicated. Although the modern algorithm, e.g., the one developed by Wong et al in 2006 [9], gives a good result, a heavy cost is required to achieve it. For instance, Wong employed quadtree and octree for analyses. The usage of additional data structures introduces several data-structure-associated methods and therefore adds considerable computation.

Algorithm proposed by Davies is fast and efficient for simple 2-D processing and modern algorithms such as algorithm proposed by Wong are useful but computationally expensive for complicated 2-D processing. If a algorithm can combine their merits and simultaneously eliminates their drawbacks, then this algorithm should be valuable for 2-D processing. This idea motivates the improvements of Davies's classical algorithm. The author made detailed investigation and analysis of the results of algorithm proposed by Davies, and found that most of the false branches leading to the useless skeletons share three special features. First, the isolated ends always consist of 3 or less local maxima, i.e., the centers of inscribed floating circles of the object to skeletonize. Second, the intermediate parts between the isolated ends and the ends joining to the main skeleton only comprise non-maxima pixels. Third, the parts of the main skeleton joining the false branches are formed by clusters of local maxima. Local maxima are mean to denote the core segments of the final skeleton, but through the investigation, the author found that they may not necessarily be the core segments, especially when 3 or less

such local maxima are isolated far from the clusters of local maxima. The improvements are made based on these findings, i.e., to trim the false branches from the skeleton. To seamlessly merge algorithms with the technical trend, the implementations of the improvements are made by using C#. Since the improved algorithm is developed by using C#, it can be directly applied or integrated in various Microsoft platforms.

During the experiments, massive binary images are tested by using the improved algorithm and other modern algorithms of skeletonization, e.g., the algorithms proposed by Wong [9]. Some experimental results show the significant differences between the improved algorithms and other algorithms. These results are presented and analyzed in the experiment section of the paper. The performance of the improved algorithm approximates the algorithms for comparisons, and it's even better than the compared algorithms in some cases. The rest of this paper is organized as follows. The theories and improvements of the algorithm are described in Section 2. There are two subsections in Section 2. In the first subsection, the original thinning algorithm designed by Davies is analyzed. In the second subsection, the improved algorithm is introduced. Experimental results are discussed in Section 3 and the final conclusion is made in Section 4.

2. Related Works

The core of the algorithm proposed by Davies [5] is the distance function for finding the local maximum. The finding procedure is essentially implemented by the ultimate erosion [13]. If p denotes a pixel in the object of a binary image denoted by X ; the erosion employs an inscribed ball B of a given radius inside the object; n and N respectively represent the times of erosions and the set of positive integers, then the ultimate erosion $Ult(X)$ can be expressed in the following formula [13].

$$Ult(X) = \bigcup_{n \in N} ((X \bullet nB) \setminus \rho_{(X \bullet nB)}(X \bullet (n+1)B)) \quad (1)$$

where \cup denotes union operation of sets, \bullet denotes erosion operation, \setminus denotes the subtraction of sets and $\rho_A(B)$ denotes the reconstruction of set A from set B [13]. Since the different parts of the object may be completely erased in different n steps, i.e., these parts are of different thicknesses. The $Ult(X)$ is a union of erosions in different parts. Assuming X contains one object that can be totally erased by a single procedure of erosions, i.e., the pixels in the core of the object are of same distances from the borders.

$$Ult(X) = (X \bullet nB) \setminus \rho_{(X \bullet nB)}(X \bullet (n+1)B) \quad (2)$$

The part $\rho_{(X \bullet nB)}(X \bullet (n+1)B)$ actually represents the condition of halting erosion. Since the erosion will finally erase the whole object by using B , there is a moment in which the erosion just completely erased the object. Assuming this moment occurs at the $(n+1)$ th step of erosion, then $(X \bullet (n+1)B)$ is an empty set. Consequently, the $\rho_{(X \bullet nB)}(X \bullet (n+1)B)$ is also an empty set inasmuch as $(X \bullet nB)$ can not be reconstructed from an empty set. Therefore,

$$\begin{aligned} Ult(X) &= (X \bullet nB) \setminus \rho_{(X \bullet nB)}(X \bullet (n+1)B) \\ &= (X \bullet nB) \setminus \emptyset \\ &= (X \bullet nB) \end{aligned} \quad (3)$$

Namely, $Ult(X)$ represents the set of pixels surviving through n times of erosions, but can not survive one additional erosion. They are core of the object and definitely should be the part of the final skeleton. Once $Ult(X)$ is obtained, it can be employed to guide the thinning process. The distance function denoted by $dist_x(p)$ is defined by the the following formula[13].

$$\forall p \in X, \quad dist_x(p) = \min\{n \in N, p \text{ not in } (X \bullet nB)\} \quad (4)$$

The above formula gives the value of distance for each pixel of the object. If p is erased in the $(n+1)$ th time of erosion, then n is recorded and compared with any other such values, the smallest one is taken as the $dist_x(p)$. The whole procedure of the improved algorithm is depicted by Fig. 3 which satisfies the standards of Unified Modeling Language (UML) activity diagram [14, 15]. There are three main steps, Step 1 Compute Local Maximum, Step 2 Generate Skeleton and Step 3 Trim False Branches. The sub steps excluding the one labeled by Delete Isolated Local Maximum in Step 1 and Step 2 represent the procedure of the algorithm proposed by Davies. The sub step marked by Delete Isolated Local Maximum and Step 3 are originally developed by the author.

In Fig. 3, Step 1 involves two main variables: *binaryImage* and *maxImage*. The former is a container of the input binary image and the latter is an image space only consisting of local maximum found by the algorithm. Both are of the same dimension of the input image. The distance function defined by (4) is implemented by two raster scans, the forward scan and the reverse scan. For making the explanation more comprehensible, every pixel in a 3-by-3 mask in *binaryImage* is denoted by one of notations $A0, A1, \dots, A8$ as following:

$$\begin{bmatrix} A4 & A3 & A2 \\ A5 & A0 & A1 \\ A6 & A7 & A8 \end{bmatrix} \quad (4)$$

The pseudo codes for the first three sub steps except the one of initialization in Step 1 in Fig. 3 are as following:

forward scan : if ($A0 > 0$) $A0 = \min(A2, A3, A4, A5) + 1$;
 reverse scan : if ($A0 > 0$) $A0 = \min(A6, A7, A8, A1) + 1$;
 locate max : if ($A0 > 0$ & & $A0 > \max(A1, A2, A3, A4, A5,$
 $A6, A7, A8)$ $B0 = A0$; else $B0 = 0$; (5)

The symbol $B0$ in (5) denotes the pixel in *maxImage* with the same coordinates of $A0$ in *binaryImage*. Note the direction of moving mask in curly brackets of *reverse scan* is the inverse direction of *forward scan* and *locate max*, and the raster scans are sequential and *locate max* is parallel. Fig. 1 shows the illustrative procedure.

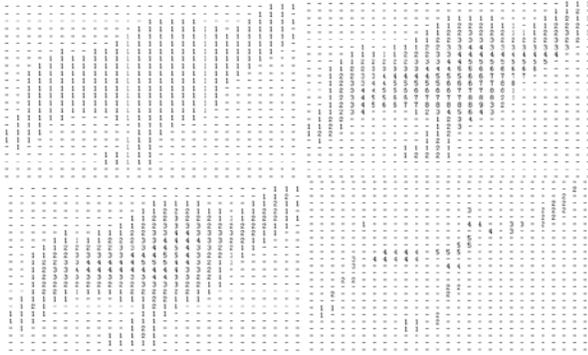


Fig. 1 Finding local maxima.

The subfigures in clockwise order in Fig. 1 respectively are the original binary image, the visualized result of forward scan, the result of reverse scan and the local maxima finally obtained by Step 1 excluding its last sub step in Fig. 3. Once the maxima are found and recorded in *maxImage*, thinning represented by Step 2 in Fig. 3 is performed according to the connectedness of each pixel, which is estimated by computing the crossing number χ (*chi*) [5]. The pseudo code is given by:

$$\begin{aligned} \chi = & (A1 \neq A3) + (A3 \neq A5) + (A5 \neq A7) + (A7 \neq A1) + \\ & 2 * \{ (!A1 \& \& A2 \& \& !A3) + (!A3 \& \& A4 \& \& A5) + \\ & (!A5 \& \& A6 \& \& !A7) + (!A7 \& \& A8 \& \& !A1) \}. \end{aligned} \quad (6)$$

Where $\&\&$, \neq and $!$ respectively denote logical AND, NOT EQUAL and NOT operators, and the connectedness criterion must be 8-connectedness. The value of χ is preserved in the variable *chi* in Step 2 and the main procedure in this step is essentially iterative which is implemented by iteratively setting and checking the states of the variable *changed*. This iterative procedure can be described by the following formula [13].

$$X \otimes \{B_{(i)}\} = (((X \otimes B_{(1)}) \otimes B_{(2)}) \dots \otimes B_{(n)}) \quad (7)$$

The notation $\{B_{(i)}\}$ denotes the Goly alphabet [7] and operator denotes the stripping operation. Here n is 4, and $B_{(1)}$, $B_{(2)}$, $B_{(3)}$, $B_{(4)}$, i.e., the north, south, west and east templates, are shown in the following formula:

$$\begin{bmatrix} * & 0 & * \\ * & 1 & * \\ * & 1 & * \end{bmatrix} \begin{bmatrix} * & 1 & * \\ * & 1 & * \\ * & 0 & * \end{bmatrix} \begin{bmatrix} * & * & * \\ 0 & 1 & 1 \\ * & * & * \end{bmatrix} \begin{bmatrix} * & * & * \\ 1 & 1 & 0 \\ * & * & * \end{bmatrix} \quad (8)$$

Notation $*$ denotes the pixel whose value can be 0 or 1. The condition for stripping directional pixels is:

$$\begin{aligned} & (A0 > 0) \& \& (A0 \text{ local maxima}) \& \& \\ & (A0 \in \text{directional pixels}) \& \& (\chi == 2) \& \& \\ & (\text{sum of neighborhood} \neq 1) \end{aligned} \quad (9)$$

The four combinations of (8) and (9) lead to the four sub steps involving stripping in Step 2. When the iterative procedure finally ends, the last sub step of Step 2 employs the following condition to remove spurs of the skeleton.

$$(A0 > 0) \& \& (\chi == 2) \& \& (\text{sum of neighborhood} \neq 1) \quad (10)$$

The original algorithm proposed by Davies terminates when Step 2 ends and offers the skeleton stored in the variable *thinnedImage* as the final result.

3. The Improved Thinning Algorithm

The algorithm proposed by Davies [5] is simple and easy to be understood and implemented, but the simplicity can simultaneously leads to the merit as well as the drawback of the algorithm. Fig. 2 shows some negative results.



Fig. 2 Negative results.

Obviously, the skeletons obtained by using algorithm proposed by Davies are nearly useless and misleading for further processing. Conversely, the modern thinning algorithms can offer the appropriate skeletons of the above binary images, but the costs are heavy. For instance, the algorithm proposed by Wong [9] employs a lookup table to estimate the connectedness, quadtree and octree to support the skeleton analysis. The introduced data structures except 2-D array not only complicate the programming but also make the algorithm computationally expensive. Hence, there are two possible ways to improve. One is enhancing the classical algorithm so that its results can approximate those of the modern algorithms as much as possible, and the other is reducing the computation complexity of the modern algorithm. The latter is hard clearly and even impossible, e.g., the algorithm proposed by Wong intrinsically depends on quadtree and octree. The algorithm proposed by Imiya [8] does not work without the curvature flow. Thus, the former is more feasible to be achieved. As a classical thinning algorithm, the algorithm proposed by Davies is based on the traditional medial axis and thus is simple and relatively fast for processing binary image. As a result, this algorithm is an ideal classical algorithm to be improved.

To improve the algorithm, the reason of the negative results must be investigated through extensive experiments. The author observed that some tiny spurs on the edge of the object in the original image generate wrong local

maxima. After further analysis, these wrong local maxima are not the real generators of the false branches shown in the above figure. Since even the wrong local maxima are deleted, the false branches can still occur. Actually, this reflects the nature of the algorithm that purely depends on the simple local operators. Therefore, two main improvements are made with respect to these facts, which are shown in Fig. 3, i.e., the sub step marked by Delete Isolated Local Maximum in Step 1 and the whole procedure of Step 3. The former is expanded in Fig. 4 and the latter is visualized in details by Fig. 5 and Fig. 6.

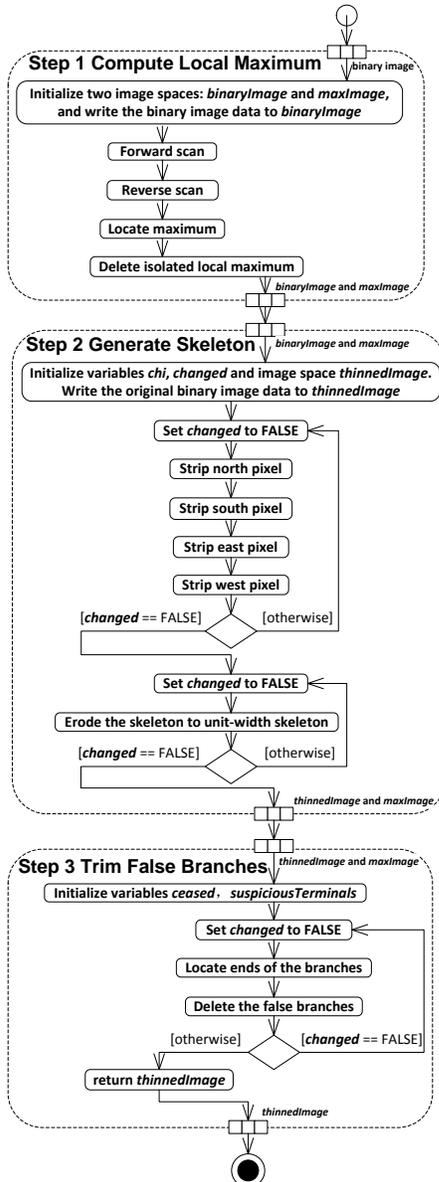


Fig. 3 The improved thinning algorithm.

Fig. 3 depicts the whole procedure of the proposed algorithm obtained by improving the algorithm proposed by Davies. Step 1 and Step 2 are inherited nearly unchanged except the sub step shown in Fig. 4 which is originally developed by the author. This sub step is an additional post processing of Step 1. It is aimed to eliminate the isolated local maxima generated by Step 1. The local maxima are represented by non-zero pixels in the variable *maxImage* in Fig. 4 and the elimination is implemented by checking the neighborhood of these non-zero pixels to see whether there are other non-zero pixels. If none is found, then the checked pixel is set to 0, i.e., the value of the background. Through the observation of the massive experimental results of the algorithm proposed by Davies, a few false branches originate from these isolated local maxima. Since the elimination of isolated local maxima can not stop the growing of all false branches. The elimination can only trim a small part or even just several points of each branch. A through trimming is necessary.

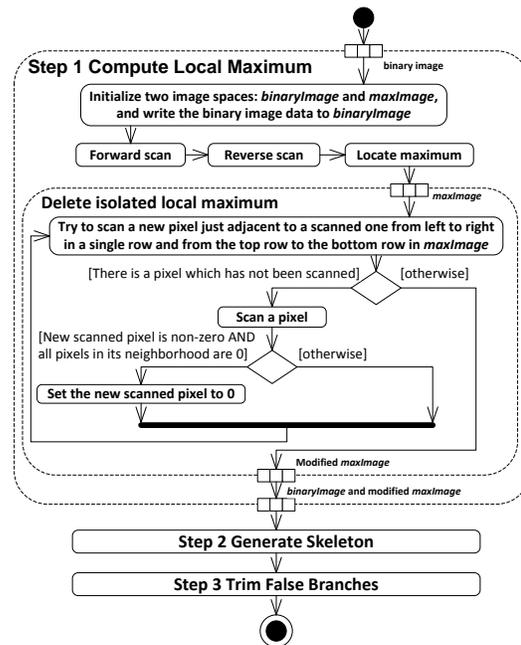


Fig. 4 Sub Step Delete Isolated Local Maximum in Step 1

Step 3 in Fig. 3 consists of two essential sub steps, i.e., Step 3.1: Locate Ends of The Branches and Step 3.2: Delete the False Branches. Step 3 is essentially iterative, i.e., it repeatedly searches and deletes the false branches until there is no false branch can be found in the skeleton. This is implemented by checking the variable *ceased*. This variable indicates whether there is a branch deleted by Step 3.2. If Step 3.2 does not delete any branches, then the value of *ceased* remains unchanged which implies there is no necessity to continue. Step 3.1 is visualized in Fig. 5. This sub step collects the ends of branches around which

there are 2 or less local maxima. Through the investigation, the author found the essential branches of the skeleton always start from clusters of about 2 or more local maxima and end at clusters of 2 or more local maxima. The skeleton generated by Step 2 is represented by the variable *thinnedImage*, the local maxima offered by Step 1 are stored in the variable *maxImage* and the coordinates of the collected ends are preserved in *suspiciousTerminals* in Fig. 5. The collecting is implemented by simultaneously checking the neighborhoods of pixels with the same coordinates in *thinnedImage* and *maxImage*. Variable *suspiciousTerminals* only records the coordinates corresponding to end pixels in *thinnedImage* and non-zero pixels in *maxImage* around which there are 0 or 1 non-zero pixels.

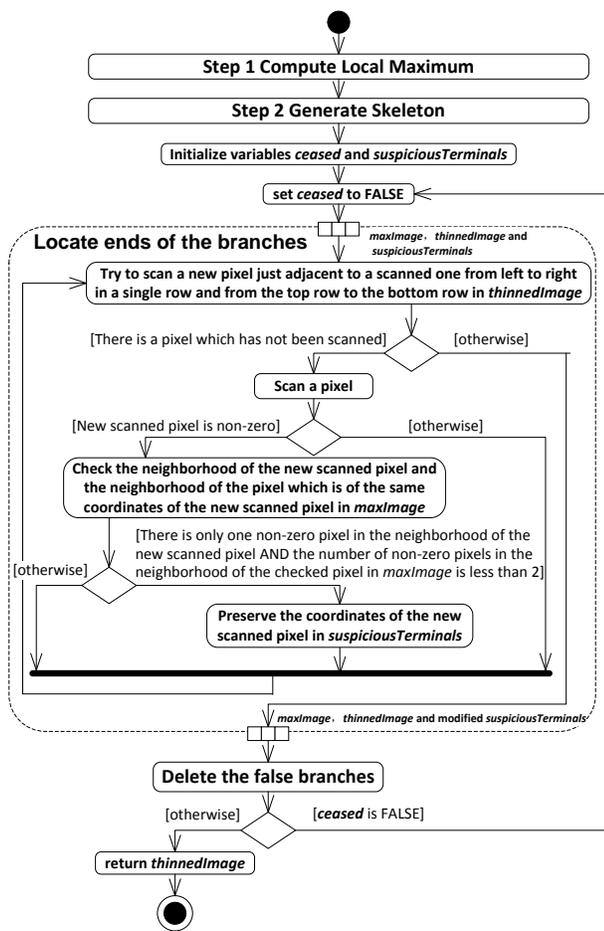


Fig. 5 Step 3.1 locate ends of the branches in Step 3.

Step 3.2 follows Step 3.1. This sub step deletes pixels in *thinnedImage* connecting to the pixels with the coordinates recorded in *suspiciousTerminals*. The deleted pixel must satisfy the condition employed by Step 3.1, i.e., Step 3.2 essentially repeats the checking of Step 3.1 for

the pixels on the false branches. If the condition is satisfied, then the checked pixels in *thinnedImage* and *maxImage* are simultaneously deleted. The checking and deletion repeats until the next pixel does not satisfy the condition. In this case, one more checking is made by estimating the connectedness represented by the variable *chi* in Fig. 6. This checking determines whether the last pixel which does not satisfy the condition should be deleted or not. The last deletion actually removes the tiny spurs generated by Step 3.

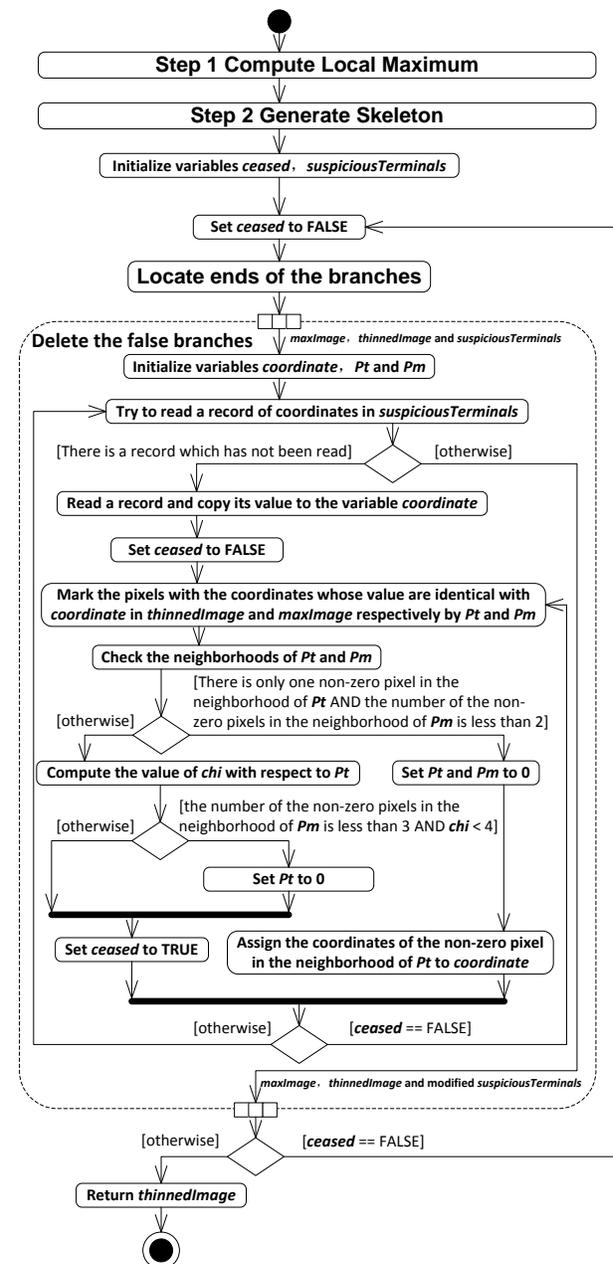


Fig. 6 Step 3.2 Delete the False Branches in Step 3.

During the implementation, several problems arise one after another. The most difficult problems include how to implement logical operations of numbers belonging to the value type Double and how to save computation overloads in subroutines of the thinning algorithm. The former is solved by employing extension methods which are features of C# and the latter is handled by interchanging pointers of image matrices. These may be the best solution in this particular situation.

3. Results and Analysis

The experiments mainly consist of three parts. The first part is designed to prove the validity of the improvement; the rest parts show the comparisons with the advanced thinning algorithms. The experiments are performed by using the image processing application developed in Windows XP Professional SP3 on a laptop with Intel Core 2 Duo 1.6 GHz CPU. Fig. 7 shows the first part. The binary images are randomly chosen from Kimia's database [18]. The sizes of the tested images mostly are of 160*160 in pixel. Clearly, the qualities of most skeletons generated by the improved algorithm are better than the algorithm proposed by Davies. In Fig. 8, the skeletons created by algorithm proposed by Wong are added for comparisons.

| Binary Image | Results | | |
|--------------|--------------------|--------------------|------------------|
| | Davies's Algorithm | Improved Algorithm | Wong's Algorithm |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Fig. 7 Comparisons of three algorithms

The results are listed in three columns. For the original algorithm, the worst cases are F11 and F61, in other cases the number and the lengths of the false branches both are less than the two cases. If only F11 is given, even a human can hardly guess the correct original object. The other case F61 is nearly impossible for a machine to recognize inasmuch as the skeleton is severely distorted by too many false branches. Thus, these cases show the drawbacks of algorithm proposed by Davies [5]. The corresponding results of improved algorithm, i.e., F12 and F62, are more recognizable and clearer. The most false branches are eliminated except a few spurs, such as the spurs on the head and the belly of the chick in F12. But the sizes and the number of remained spurs are acceptable and the general shapes of the skeletons approximate the results of algorithm proposed by Wong.

Compared to the algorithm proposed by Davies, the skeletons generated by the improved algorithm are acceptable and useful for further processing, but the algorithm has its own drawback, i.e., the over trimness. The results F32, F62 and F72 uncover the over trimness. The worst one among the three is F32. Note F30 just is the rotated version of F50, which implies their corresponding skeleton should be similar. Results F33 and F53 of algorithm proposed by Wong are similar in some degree, but the F32 and F52 of the improved algorithm are just coarsely similar. From this fact, the improved algorithm is proved to be anisotropic. Despite of the isotropy, the general shape of F52 is surprisingly better than the corresponding F53 of algorithm proposed by Wong.

| Binary Image | Results | | Binary Image | Results | |
|--------------|--------------------|--------------------|--------------|--------------------|--------------------|
| | Davies's Algorithm | Improved Algorithm | | Davies's Algorithm | Improved Algorithm |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

Although the improved algorithm is anisotropic which is caused by over trimness, the costs are quite low. This is the merit of the improved algorithm. Unlike any other modern thinning algorithm, the only data structure employed is the simple 2-D array which simplifies programming and does not require additional data-structure-dependent subroutines. Note, the improvements are implemented by using only two additional subroutines and one of them is so simple that it only needs one loop of the array. The other requires several iterations, but it only consists of two main blocks in which there are a few lines of code. Compared to the data-structure-dependent subroutines, it's relatively simple. More experimental results can be found in the following figure.

| Binary Image | Results | | |
|--|--|--|--|
| | Davies's Algorithm | Improved Algorithm | PCNN based Algorithms |
|  C00 |  C01 |  C02 |  C03 |
|  C10 |  C11 |  C12 |  C13 |
|  C20 |  C21 |  C22 |  C23 |
|  C30 |  C31 |  C32 |  C33 |
|  C40 |  C41 |  C42 |  C43 |

Fig. 7 Comparisons of three algorithms

The skeletons in the last column are obtained by using thinning algorithms based on pulse coupled neural network (PCNN). Skeletons C03, C13 and C23 are generated by using the algorithm proposed by Gu [16], and the rest skeletons are obtained by using Shang's algorithm [17]. Generally, the improved algorithm generates better results than original algorithm except C32 whose little distortion is clearly caused by over trimness. Since the binary images C30 and C40 are too simple, results from different algorithms are similar. There's also a better result of the improve algorithm than the algorithm proposed by Gu, i.e., C12. There is little difference between C11 and C13. Both have obvious spurs. These spurs are eliminated or shortened in C12, which again proves the improvements are valid.

The computational complexity of the trimness can be easily deduced to be $O(m^2)$ where m is the number of all ending pixels in a binary image whose size is n^2 . The first step of the Fig.3 consumes $8 \times n^2$ times of searching to find ending pixels. Then it requires $8 \times m$ times of searching to detect the neighboring local maxima. There are two branches in the iterations. The more expensive branch is

the one involving the computation of χ and its complexity is $O(c_1 + 8 \times m + 1) = O(m)$ where c_1 is a constant of computing χ . The worst case of iterations is $O(m \times (c_2 + m)) = O(m^2)$ where c_2 denotes the number of iterations which actually is a constant. Finally, the complexity of the proposed algorithm is $O(n^2) + O(m^2) = O(n^2)$ where $O(n^2)$ is the complexity of the original algorithm proposed by Davies. This result is identical with the algorithm proposed by Wong.

Generally, the results of the improved algorithm approximate the algorithm proposed by Wong and the improvements are implemented by adding two relatively simple subroutines, but the improved algorithm is anisotropic, which is its the main drawback.

5. Conclusions

In this paper, the Davies's classical thinning algorithm [5] is totally analyzed. The simplicity of the algorithm makes the understanding and implementation easy, but it also leads to some valueless results when the algorithm is applied in some binary images. The author observed the resulting matrices and found several facts which actually cause such valueless results. The improvements are made based on these found facts, and tested on a series binary image which is also used to test an advanced thinning algorithm designed by Wong [9]. Compared to the results of algorithm proposed by Wong, the results of the improved algorithm are acceptable and even better in some case. Consider the complexity and additional sophisticated data structures employed by algorithm proposed by Wong, the costs of achieving such results are quite low, but there are few tiny spurs scattering on the skeleton and the algorithm is anisotropic. The isotropy and the spurs require deeper analysis and more subtle design to be removed.

Acknowledgments

This work is supported by the project "A Research on 3D Reconstruction and Uncontact Measurement of Cow Body Conformation Based on Machine Vision" (Grant 61063021/2011) sponsored by the national natural science foundation of China. The author would like to thank Geyong Min of university of Bradford in UK for his comments and insightful suggestions.

References

- [1] H. Blum, "A transformation for extracting new descriptors of shape", Models for the Perception of Speech and Visual Form, pp. 362-380, 1967.

- [2] V. Bansal, R.M.K. Sinha, "Segmentation of touching and fused Devanagari characters", *Pattern Recognition*, vol. 35, no. 4, pp. 875-893, Apr. 2002.
- [3] R.G. Casey, E. Lecolinet, "A survey of methods and strategies in character segmentation", *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 18, no. 7, pp. 690-706, Jul. 1996.
- [4] H. Ma, D. Doermann, "Adaptive Hindi OCR using generalized hausdorff image comparison", *ACM Trans. on Asian Language Information Processing*, vol. 2, no. 3, pp. 193-218, Sep. 2003.
- [5] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*, 3rd ed. CA: Morgan Kaufmann, 2005.
- [6] R. Krishnapuram, F. Chen, "Implementation of parallel thinning algorithms using recurrent neural networks", *IEEE Trans. Neural Netw.*, vol. 4, no. 1, pp.142-147, Jan. 1993.
- [7] Y.Y. Tang, X.G. You, "Skeletonization of ribbon-like shapes based on a new wavelet function", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1118-1133, Sep. 2003.
- [8] A.Imiya, M. Saito, "Thinning by Curvature Flow", *Journal of Visual Communication and Image Representation*, vol. 17, no. 1, pp.27-41, Feb. 2006.
- [9] W. Wong, F. Y. Shih, T. Su, "Thinning algorithms based on quadtree and octree representations", *Information Sciences*, vol. 176, no. 10, pp.1379-1394, May. 2006.
- [10] S. Bag, G. Harit, "An improved contour-based thinning method for character images", *Pattern Recognition Letters*, vol. 32, no. 11, pp. 1836-1842, Oct. 2011.
- [11] J. Albahari, B. Albahari, *C# 4.0 In A Nutshell 4th ed.* CA: O'Reilly, 2010.
- [12] L. Jia, Y. Sun, M. Wang, Y. Gu, "A Research on Implementation of Image Scattergram by Using C#", in 2011 Int. Conf. System Design and Data Processing, Tai Yuan, 2011, pp. 353- 355.
- [13] M. Sonka, V. Hlavac, R. Boyle, *Image Processing, Analysis, and Machine Vision*, 3rd ed. CT: Cengage Learning, 2008.
- [14] I. Jacobson, G. Booch, and J. Rumbaugh, *Unified Modeling Language User Guide*, 2nd ed. MA: Addison-Wesley, 2005.
- [15] J. Rumbaugh, I. Jacobson, and G. Booch, *Unified Modeling Language Reference Manual*, 2nd ed. MA: Addison-Wesley, 2010.
- [16] X. Gu, D. Yu b, L. Zhang, "Image thinning using pulse coupled neural network", *Pattern Recognition Letters*, vol. 25, no. 9, pp. 1075-1084, Jul. 2004.
- [17] L. Shang, Z. Yi , "A class of binary images thinning using two PCNNs", *Neurocomputing*, vol. 70, no. 4, pp. 1096-1101, Jan. 2007.
- [18] D. Sharvit, J. Chan, H. Tek, B. Kimia, "Symmetry-based indexing of image databases", *J. of Visual Commun. and Image Representation*, Vol. 9, no. 4, pp. 366-380, Dec. 1998.
- processing, 3D reconstruction and virtual reality.

Liang Jia He is a faculty member at Changzhou university of Jiangsu province in China. He received the MS degree in computer science from the Nanjing university of Science and Technology in 2009 and the BS degree in computer science from Beifang university of Nationalities in 2004. His research interests mainly include image processing, computer vision and database application development.

Zhenjie Hou He was born in Hohhot, China. He received PH.D degree in 2005. His research interests include digital image