Thinning Algorithm to Generate k-Connected Skeletons

Juan Luis Díaz de León, C. Yánez, and Giovanni Guzmán

Digital Image Processing Laboratory
Centre for Computing Research (CIC)
Mexico City, Mexico
{jdiaz,cyanez,jguzmanl}@cic.ipn.mx

Abstract. This paper presents a novel algorithm intended to generate k-connected skeletons of a digital binary image using a new mask set. These skeletons may be 4 or 8 connected. The new algorithm performs a thinning process that finish when it is not possible to eliminate additional pixels without breaking the connectivity. The end-point criterion and a 3x3 masks set are used to decide if a pixel is eliminated. The proposed masks set for each kind of connectivity covers all the necessary cases, and guarantee to obtain a one pixel wide and k-connected skeleton without parasitic branches. The new algorithm yields some advantages to developers. It is not just oriented to written characters or some kind of object in particular; this means that the algorithm can be adapted easily to any application generating good results. Besides, the user can work with different classes of connectivity; note that several recent algorithms use 4-connectivity while 8-connectivity is used for others. Additionally, the skeletons produced by the new algorithm are immune to structured noise around the processed objects.

1 Introduction

Skeletonization is an important technique used in many areas in digital image processing. Its goal is to reduce an object R within an image in order to generate an object S (commonly called skeleton) whose thickness is just 1 pixel and it is generally connected, preferable on a specific metric. It is well known that the general topologic characteristics based on connectivity are not sufficient to describe a given object, because the skeleton of a simple connected object can be just a point, like was raised by Rosenfeld [1], and the representation yielded it is useless in many applications. For this reason, it is recommendable to take into account the objects geometric properties as Pavlidis's end-point condition [2]. This condition helps to preserve the "geometric form" of the object in the final skeleton.

Actually, there exist two types of algorithms to obtain the skeleton from a given image: the pixel based algorithms, and the not pixel based ones [3]. In the former manner, each background pixel takes part on the compute of the skeleton. Thinning algorithms [4, 5] and the distance transforms [4, 6] belong to this kind of techniques. Most of the more efficient methods are based on thinning techniques.

The basic principle of the thinning methods is to eliminate repeatedly those points from the contour of an object of such a way that the elimination of such points don't alter the object connectivity, and respect the local end-point condition, just as is done in the classic Pavlidis algorithm [2]. In order to decide if the elimination of a pixel alters or not, the connectivity of the object, it is useful to use local connectivity criterions, considering a neighborhood centered at the pixel to be analyzed. These criterions are usually formulated in terms of digital geometry [7], or by means of masks [8].

A. Sanfeliu et al. (Eds.): CIARP 2004, LNCS 3287, pp. 605–612, 2004.

In general, after thinning it is no possible to perform the original object reconstruction and, despite this technique preserves connectivity, there exist two problems where the actual study focuses, namely: the algorithms execution time, and the resulting object shape. The former point it is no critic, because the well-known advances in the speed of the devices used on the computers, and the possibility of use parallel algorithms as the one developed by Zhang and Suen [9]. With respect to the second point, a potential problem is the appearance of parasitic branches, which were not in the original object. On this topic focused the work of Díaz de León [10], Lü and Wang [11], Abdulla, Saleh and Morad [12] among others.

The goal of this work is to present a new sequential thinning algorithm for skeletonizing binary images. The approach is based on the use of masks that allow to determine if a pixel could be or could not be eliminated, taking into account a connectivity criterion and avoiding the elimination of end points. This algorithm is efficient with respect to processing time; that is made possible thanks to the use of only logical binary operations instead of arithmetic operations.

The remainder of the document is organized as follows: in section 2 some important concepts are cited, and the general strategy of the proposed thinning algorithm is formulated. In section 3 the masks used in connectivity of type 4 and 8, are deduced. In section 4 the system proofs are presented, and finally, in section 5 the conclusions on the present paper are given.

2 Preliminaries and General Strategy

Next we will introduce some important concepts about digital geometry. Let $p=(x,y)$, $q=(u,v)$ be points in Z^2. The following metrics in Z^2 are widely used:

$$\begin{aligned} d_4 &= |x-u|+|y-v|, \\ d_8 &= máx\left(|x-u|,|y-v| \right), \\ d_{6R} &= máx\left(|x-u|,|y-v|,|x-u+y-v| \right), \\ d_{6L} &= máx\left(|x-u|,|y-v|,|x-u-y+v| \right). \end{aligned} \tag{1}$$

Each one of these metrics $d_k, k \in \{4,6L,6R,8\}$, induce a symmetric and not reflexive relation N_k in Z^2 , which is defined by $(p,q) \in N_k \Leftrightarrow d_k(p,q)=1$. The structure (Z^2,N_k) is usually called the k-neighborhood graph, and the points *p, q* are k-neighbors if $(p,q) \in N_k$; additionally, $N_k(p)$ denotes the set of all k-neighbors of *p*. In the graph (Z^2,N_k) the connectivity concept from the indirect graphs theory is applied. Then, a subset $R \subseteq Z^2$ is k-connected if for each arbitrary $p,q \in R$, exists a sequence of points in $R, p=a_1,a_2,...,a_n=q$, where a_i y a_{i+1} are k-neighbors for $i=1,...,n-1$.

Now, let *R* be a k-connected subset in Z^2. A point *p* in *R* is called end point in *R* if it has just one k-neighbor in *R*. Note that the only element in a set containing just one point in Z^2, is an end point. Besides, if *R* has at least two points, then $p \in R$ is an end point in *R* if *p* has exactly one k-neighbor in *R*. The idea behind the thinning is to remove iteratively points in *R* preserving not only the connectivity, but also preserving the end point condition into the actual points set through each iteration. It is clear that the contour in *R* are candidates to be eliminated without altering the connectivity. Recalling, *p* is a contour point in *R* if *p* has a *k*'-neighbor *q* fulfilling $q \notin R$, where

$k'=8$ if $k=4$, $k'=4$ if $k=8$, and $k'=k$ if $k\in\{6L,6R\}$. Thanks to the way the contour is defined, it is possible to warranty that the contour is k-connected.

The general strategy of the algorithm is as follows:

1. $R_0=R; i=1; R_i:=R.$
2. Determine the contour C from R_i.
3. For $p\in C$: If p fulfills the k-condition:
4. "p is not an end point in R_i, and $R_i\setminus\{p\}$ is k-connected", then $R_i=R_i\setminus\{p\}$. Repeat step 3 with the remain points in the contour.
5. If $R_i=R_{i-1}$, then stop; R_{i-1} is the k-skeleton of R. Else, increase i in 1, do $R_i=R_{i-1}$, and go to step 2.

3 Thinning Masks

Next, for a 4-connected skeleton the T_4 masks set will be deduced. Let suppose that R is a 4-connected subset in (Z^2,N_4). The needed masks must cover all possible local configurations, where 4-connectivity and end points shall be preserved. As a starting point, at least 6 masks are required, which are shown in the Figure 1.

$$P_1=\begin{bmatrix}X&0&X\\1&p&1\\X&0&X\end{bmatrix},\ P_2=\begin{bmatrix}X&1&X\\0&p&0\\X&1&X\end{bmatrix},\ P_3=\begin{bmatrix}X&X&X\\X&p&X\\0&1&0\end{bmatrix},P_4=\begin{bmatrix}X&X&0\\X&p&1\\X&X&0\end{bmatrix},\ P_5=\begin{bmatrix}0&1&0\\X&p&X\\X&X&X\end{bmatrix},\ P_6=\begin{bmatrix}0&X&X\\1&p&X\\0&X&X\end{bmatrix}$$

Fig. 1. Cases where *p* is needed to preserve 4-connectivity or as an end-point

The X symbol used along the entire document, denotes don't care condition. Note that the masks P_1 and P_2 are useful to cover local situations as the shown in the Figure 2.

$$\begin{bmatrix}0&0&1\\1&p&1\\1&0&0\end{bmatrix},\begin{bmatrix}1&0&0\\1&p&1\\0&0&1\end{bmatrix},\begin{bmatrix}1&0&1\\1&p&1\\1&0&1\end{bmatrix},\begin{bmatrix}0&0&0\\1&p&1\\1&0&1\end{bmatrix},\begin{bmatrix}1&0&1\\1&p&1\\0&0&0\end{bmatrix},\begin{bmatrix}1&1&0\\0&p&0\\0&1&1\end{bmatrix},\begin{bmatrix}0&1&1\\0&p&1\\1&1&0\end{bmatrix},\begin{bmatrix}1&1&1\\0&p&0\\1&1&1\end{bmatrix},\begin{bmatrix}1&1&0\\0&p&0\\1&1&0\end{bmatrix},\begin{bmatrix}0&1&1\\0&p&0\\0&1&1\end{bmatrix}$$

Fig. 2. Some specific patterns covered by the masks P_1 and P_2

In all these patterns, the point p is needed to preserve 4-connectivity. In order to justify why the masks P_3,P_4,P_5,P_6 are needed, the patterns shown in the Figure 3 can be considered.

$$\begin{bmatrix}X&y&X\\y&p&y\\0&1&0\end{bmatrix},\begin{bmatrix}X&y&0\\y&p&1\\X&y&0\end{bmatrix},\begin{bmatrix}0&1&0\\y&p&y\\X&y&X\end{bmatrix},\begin{bmatrix}0&y&X\\1&p&y\\0&y&X\end{bmatrix}$$

Fig. 3. Cases where the masks P_3,P_4,P_5,P_6 are used

In these cases, point p is necessary to connect any point labeled y with the pixel having intensity 1. If the all y labeled points are 0, then p is an end point, thus p won't be eliminated. The points labeled as X are not relevant in 4-connectivity. Additionally, it is a must to be sure that the skeleton doesn't have parasitic branches, as some of the configurations shown in the Figure 4.

$$\begin{bmatrix} X & 1 & 1 \\ 0 & p & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & X \\ 0 & p & 1 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & p & 0 \\ 1 & 1 & X \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 1 & p & 0 \\ X & 0 & 0 \end{bmatrix}$$

Fig. 4. Possible cases of parasitic branches

Independently of the value in X, the pixel p must be eliminated, because the branches having length 1 are prone to cause structural noise on the generated skeleton. The pixel p can be deleted due to any configuration in the Figure 4 does not fulfill any of the patterns covered by the masks from the Figure 1.

By analyzing the Figure 5 (pixels labeled 3 belonging to the contour), it is clear that only one of the two points labeled as r is necessary within the skeleton being thus redundant the other r labeled point; however, if the all points are analyzed in parallel with the six masks in the Figure 1 both points will be eliminated, resulting in a non-connected skeleton.

3			
3			
3	*r*	3	3
r	1	1	3
3	1	1	3
3	1	1	3

3			
3			
3			
	3	3	
	3	3	
	3	3	

Fig. 5. Redundant connectivity case

Fortunately the previous problem is solved due to the asynchronous execution of the algorithm. This is, one of the r labeled points is analyzed first and then considered redundant being thus eliminated; after that, when the other r labeled point is analyzed, this is not redundant anymore and thus preserved. The remainder cases not covered by the masks in the Figure 1, are covered with the masks in the Figure 6.

$$A_1 = \begin{bmatrix} X & 0 & X \\ 1 & p & x \\ 0 & y & X \end{bmatrix},\ A_2 = \begin{bmatrix} 0 & x & X \\ 1 & p & y \\ X & 0 & X \end{bmatrix},\ \mathrm{A}_3 = \begin{bmatrix} 0 & 1 & X \\ y & p & 0 \\ X & x & X \end{bmatrix}$$

$$A_4 = \begin{bmatrix} X & 1 & 0 \\ 0 & p & x \\ X & y & X \end{bmatrix},\ A_5 = \begin{bmatrix} X & y & 0 \\ x & p & 1 \\ X & 0 & X \end{bmatrix},\ A_6 = \begin{bmatrix} X & 0 & X \\ y & p & 1 \\ X & x & 0 \end{bmatrix}$$

$$A_7 = \begin{bmatrix} X & x & X \\ 0 & p & y \\ X & 1 & 0 \end{bmatrix},\ A_8 = \begin{bmatrix} X & 0 & X \\ y & p & 1 \\ X & x & 0 \end{bmatrix},\ \text{with } x \wedge y = 0$$

Fig. 6. Masks that cover the remainder cases for 4-connectivity

For these new masks, x or y must have a zero value in order to avoid single pixel length branches. These new masks eliminate all possible redundancy in the skeleton, and the 4-connectivity is preserved, since the algorithm is sequential. By adding the masks in the Figures 1 and 6, results 14 masks in total, but the last ones can be combined with P_1, P_2 by means of Boolean operations in order to obtain the masks shown in the Figure 7.

$$B_1 = \begin{bmatrix} a & b & X \\ 1 & p & e \\ c & d & X \end{bmatrix}, B_2 = \begin{bmatrix} c & 1 & a \\ d & p & b \\ X & e & X \end{bmatrix}, B_3 = \begin{bmatrix} X & d & c \\ e & p & 1 \\ X & b & a \end{bmatrix}, B_4 = \begin{bmatrix} X & e & X \\ b & p & d \\ a & 1 & c \end{bmatrix}$$
$$\text{where } c \wedge d = 0, a \wedge b = 0, b + d + e = 1$$

Fig. 7. Simplified masks

Notice that the masks P_1 and P_2 are particular cases of B_1, B_3 and B_2 , B_4 respectively, given $c = a = 0$ *y* $e = 1$. Also, A_1, A_2, A_3, A_4, A_5, A_6, A_7 and A_8, are particular cases of B_1, B_2, B_3 *y* B_4 ; furthermore, if the pixel opposed to the one labeled 1 has value 1 in the masks P_3, P_4, P_5 and P_6, the results are also particular cases of B_1, B_2, B_3, B_4.

These masks include all the possible configurations of end points. This means that P_3, P_4, P_5 and P_6 can be replaced by a unique mask, which is shown in the Figure 8. As a consequence, the masks set: $T_4 = \{B_1, B_2, B_3, B_4, B_5\}$ is the final masks set for the 4-connected thinning algorithm.

$$B_5 = \begin{bmatrix} 0 & h & 0 \\ e & p & g \\ 0 & f & 0 \end{bmatrix}, \text{with } e + f + g + h = 1$$

Fig. 8. Mask grouping the cases of P_3, P_4, P_5, P_6

To cover all the cases for the type 8 connectivity, at least the cases shown in the Figure 9 are required.

$$P_1 = \begin{bmatrix} 1 & 0 & X \\ 0 & p & X \\ X & X & X \end{bmatrix}, P_2 = \begin{bmatrix} X & 0 & 1 \\ X & p & 0 \\ X & X & X \end{bmatrix} P_5 = \begin{bmatrix} 0 & 0 & X \\ 1 & p & X \\ 0 & 0 & X \end{bmatrix}, P_6 = \begin{bmatrix} X & X & X \\ 0 & p & 0 \\ 0 & 1 & 0 \end{bmatrix}$$
$$P_3 = \begin{bmatrix} X & X & X \\ X & p & 0 \\ X & 0 & 1 \end{bmatrix}, P_4 = \begin{bmatrix} X & X & X \\ 0 & p & X \\ 1 & 0 & X \end{bmatrix} P_7 = \begin{bmatrix} X & 0 & 0 \\ X & p & 1 \\ X & 0 & 0 \end{bmatrix}, P_8 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & p & 0 \\ X & X & X \end{bmatrix}$$

Fig. 9. Former masks for the type 8 connectivity

If at least one of the pixels marked with the don't care value (*X*) has an intensity of 1, the *p* pixel must remain in order to keep connectivity. If none of these pixels is one, *p* is an end point, and it must not be eliminated. Additionally, the two masks shown in the Figure 10 are needed, since they are cases where the connectivity must be preserved, and they aren't covered by the masks in the Figure 9.

$$P_9 = \begin{bmatrix} X & 0 & X \\ 1 & p & 1 \\ X & 0 & X \end{bmatrix}, P_{10} = \begin{bmatrix} X & 1 & X \\ 0 & p & 0 \\ X & 1 & X \end{bmatrix}$$

Fig. 10. Additional masks

The order in which the algorithm scans the image affects to some of the former 4 masks from the mentioned above. For example, if the scanning is done *top–down* (left to right, and upper to bottom), the P_1 must be changed by the mask shown in the Figure 11.

$$P_1 = \begin{bmatrix} 1 & 0 & d \\ 0 & p & e \\ h & g & f \end{bmatrix}, \text{with } d+e+f+g+h>0$$

Fig. 11. Mask changed due to the order of the image scan

With respect to masks P_5, P_6, P_7 and P_8, when the pixel located in opposition to the pixel marked with 1 has value 1, particular cases of the masks P_9 and P_{10} arises as is shown in the Figure 12a. If on the contrary, this opposite pixel has a zero value but some of their lateral pixels are 1-valued, then a special case of the former 4 masks (P_1, P_2, P_3 and P_4) arise. This situation is shown in the Figure 12b.

$$\begin{bmatrix} 0 & 0 & X \\ 1 & p & 1 \\ 0 & 0 & X \end{bmatrix}, \begin{bmatrix} X & 1 & X \\ 0 & p & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} X & 0 & 0 \\ 1 & p & 1 \\ X & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & p & 0 \\ X & 1 & X \end{bmatrix}$$

(a)

$$\begin{bmatrix} 0 & 0 & X \\ 1 & p & 0 \\ 0 & 0 & X \end{bmatrix}, \begin{bmatrix} X & 0 & X \\ 0 & p & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} X & 0 & 0 \\ 0 & p & 1 \\ X & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & p & 0 \\ X & 0 & X \end{bmatrix}$$

(b)

Fig. 12. (a) Particular cases of P_9 y $P_{10.}$ (b) Particular cases of the former four masks

Because of this, the masks P_5, P_6, P_7 and P_8 are only necessary for configurations as the ones shown in the Figure 13a. The masks P_5, P_6, P_7 and P_8 can be eliminated and replaced by a unique one. This new mask is shown in the Figure 13b.

Finally, the masks set for 8-connectivity is $T_8 = \{P_1, P_2, P_3, P_4, P_9, P_{10}, P_{11}\}$.

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & p & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & p & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & p & 1 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & p & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

(a)

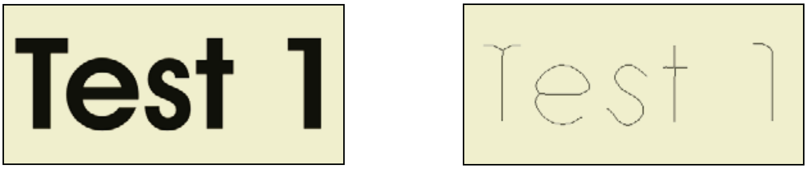
$$P_{11} = \begin{bmatrix} 0 & a & 0 \\ d & p & b \\ 0 & c & 0 \end{bmatrix}, \text{with } a+b+c+d=1$$

(b)

Fig. 13. (a) Cases where the masks P_5, P_6, P_7 and P_8 are used. (b) Mask that replace them

4 Final Tests and Results

The proposed algorithm was implemented in the Integrated Development Environment (IDE) C++ Builder release 5.0. We probe the technique using two types of images: synthetics and real. In both cases, we obtained good results. In Figure 14, we shown the result using a synthetic image, generated with a commercial drawing tool.



(a) (b)

Fig. 14. Algorithm test. (a) Image to process. (b) 8-connected algorithm

In some practical cases, like the one shown in Figure 15, it is convenient to apply the opening morphological operator, using a structured element of size 3 x 3, to remove parasitic pixels which can generate not desired branches in the final skeleton.

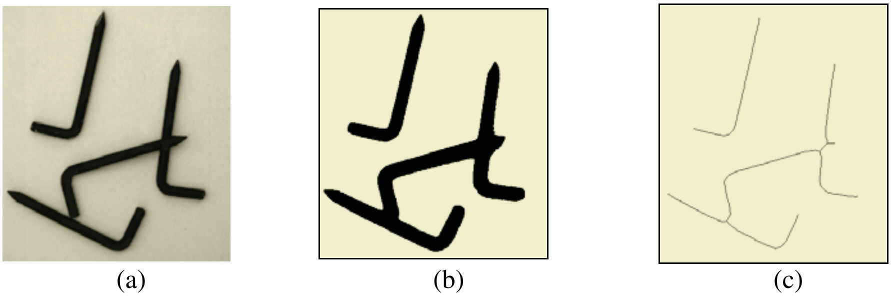

(a) (b) (c)

Fig. 15. (a) Source image. (b) Thresholding and opened image (c) 8-connected skeleton

5 Conclusions

In this paper an efficient thinning algorithm is presented, using a small set of templates that allow us to decide when a contour pixel can be removed without losing connectivity. One advantage of our proposal is that due to the low time obtained, the algorithm can be applied in systems with strong time constrains such as real time applications. A very important fact is that the templates are not affected by the presence of structured noise and the final skeleton preserves a topological structure based on a single metric space.

Acknowledgments

The authors of this paper wish to thank to the Centre for Computing Research (CIC), the General Coordination of Postgraduate Study and Research (CGEPI) and to the National Polytechnic Institute (IPN) for their support. Additionally, the authors wish to thank the reviewers of this work for their pertinent comments.

References

1. Rosenfeld, “Characterization of parallel thinning algorithms”, Information and Control, 29, pp. 286 – 291, 1975.
2. T. Pavlidis, “Algorithms for Graphics and Image Processing”, Ed. Computer Science Press, 1982.
3. J. J. Zou, H.-H. Chang, H. Yan, “Shape skeletonization by identifying discrete local symmetries”, Pattern Recognition, vol. 34 pp. 185 – 1905, 2001.
4. R. W. Smith, “Computer processing of line images: a survey”, Pattern Recognition, vol. 20, pp. 7 – 15, 1987.
5. L. Lam, S.-W. Lee, C. Y. Suen, “Thinning methodologies – a compressive survey”, IEEE Tran. Pattern A., vol. 14, pp. 869 – 885, 1992.
6. F. Leymarie, M. D. Levine, “Simulating the grassfire transform using an active contour model”, IEEE Trans. Pattern A., vol. 14, pp. 56 – 75, 1992.
7. H. Tamura, “A comparison of line thinning algorithms from the digital viewpoint”, Proc. of 4th ICPR, pp. 715 – 719, 1987.

8. E. S. Deutsch, “Thinning algorithms on rectangular, hexagonal, and triangular arrays”, Communications of the ACM, vol. 15, pp. 827 – 837, 1972.
9. T. Y. Zhang, C.Y. Suen, “A fast parallel algorithm for thinning digital patterns”, Image Processing and Computer Vision, vol. 27, pp. 236 – 239, 1984.
10. J. L. Díaz de León S, “Algoritmos de esqueletización de imágenes digitales binarias”, Master thesis, Mexico, 1993.
11. H. E. Lü, S. P. Wang, “A comment on a fast parallel algorithm for thinning digital patterns”, Image Processing and Computer Vision, vol. 29, pp. 239 – 242, 1986.
12. W.H. Abdulla, A. O. Saleh, A. H. Morad, “A preprocessing algorithm for hand written character recognition”, Pattern Recognition Letters, vol. 7, pp. 13 – 18, 1988.
13. R. González, E. Woods, “Digital Image Processing”, Second Edition, Prentice Hall.