AN IMPROVED APPROACH FOR THINNING BY PRESERVING LOCAL COUPLING POINTS

Lalit Kumar, Abhay Bansal, Dinesh Ganotra and Neeraj Jain
Newgen Software Technologies Limited, A-6, Satsang Vihar Marg, Qutab Institutional Area, New Delhi–110067 India

Keywords: Binarization, Thinning, Skeleton, Local Coupling Point, LCP, ICR, OCR, Offline Signature Verification.

Abstract: Image thinning, while preserving geometrical properties of the image, remains one of the most challenging areas in Image Processing. In this paper, we introduce an image thinning approach that takes as input gray scale images and produces binary thinned images as output. Our approach uses bi-directional iterative thinning process to get single-pixel-thinned image (primary skeleton), while preserving the geometric properties of the image. The gray scale image is binarized using dyamic threshold, and further processed to remove noise. Black pixels are classified as contour pixels (pixels exposed to the background) and body pixels (black pixels other than the contour pixels). Thinning process involves scanning contour pixels from two opposite directions simultaneously, while preserving Local Coupling Points (LCP) and removing the rest of pixels.

1 INTRODUCTION

Thinning of images is a fundamental requirement in a wide range of application areas, such as Courtesy Amount Recognition and Legal Amount Recognition (CAR, LAR) (Lecce et al., 2000), Handwritten Signature verification (Ammar et al., 1986), Optical Character Recognition(Cowell and Hussain, 2003), Fingerprint Classification (Mehtre, 1993), Printed Circuit Boards (El Mesbahi and Chaibi, 1993), etc.

A number of algorithms implementing various approaches for thinning of images have been proposed. The algorithms are broadly divided into two categories: iterative and non-iterative. Iterative algorithms repeatedly apply a procedure on the input image till the skeleton image is obtained. On the contrary, non-iterative algorithms take a set of predetermined parameters and try to obtain primary skeleton in a single iteration (Neuslds and Olszewski, 1994).

The iterative algorithms are further divided into two sub-categories, sequential and parallel. In sequential algorithms all the pixels are considered to take decision for a single pixel. On the other hand, in parallel algorithms (Ubeda, 1993), (Hang and Wang, 1994) individual pixels can be independently judged for decision-making. The proposed algorithm is a combination of sequential and parallel approach.

In the past decade, a number of algorithms have been proposed for thinning of gray scale images. Most of these deal with simple images, such as handwritten characters and machine-printed text in specific languages (Ahmed et al., 2002). Few approaches effectively deal with complex images such as signatures while at the same time preserving the intricate shapes of such images. For example, both Han's algorithm (Datta and Parui, 1994) and Datta's algorithm (Han et al., 1997) are able to preserve the shapes of the simple objects, but split the edges into spurious branches. Shape preservation is a key requirement for off-line signature verification. This paper targets the thinning of signature image while preserving geometrical features, such as long strokes, curves cross points, and end points. With threshold changes, the proposed approach would find extensive use in application areas, such as detecting the CAR-LAR amount and enhancing the ICR engines' accuracy in reading text written with thick pen.

The paper is organized as follows: Section 2.1 deals with preprocessing that includes binarization of image and noise removal. Section 2.2 deals with the classification of the black pixels as Local Coupling Point (LCP) and body pixels. In section 3, experimental results are discussed.

Kumar L., Bansal A., Ganotra D. and Jain N.
AN IMPROVED APPROACH FOR THINNING BY PRESERVING LOCAL COUPLING POINTS.
DOI: 10.5220/0002156700900095
In *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications (VISIGRAPP 2009)*, page
ISBN: 978-989-8111-69-2

The following conventions are used in the paper:

G	Graph
M	Matrix
n	Number of elements
V	One dimensional vector
$M_{(x,y)}$	Element at xth column and yth row of matrix M

2 PROPOSED ALGORITHM

The proposed algorithm uses an iterative thinning process that preserves the geometrical features in complex images. The algorithm is based on the concept of classifying pixels as LCP and preserving them.

For understanding LCP, consider a graph G_n in the form of an nxn matrix, where nodes are represented by black pixels and edges by connected black pixels. For a 3×3 matrix, there exists only one interior node, $M_{(2,2)}$, connected to all exterior nodes.

We define an LCP as an interior node, which when removed causes discontinuity in graph, G; i.e., a path, which traces all the exterior nodes, cannot be established without including the LCP.

Figure 1(a)(i) represent the 3x3 pixels matrix, M, where $M_{(2,2)}$ is the pixel to be checked for LCP; (a)(ii) represents M in the form of a graph where $M_{(1,1)}$, $M_{(2,2)}$, $M_{(3,1)}$, and $M_{(3,3)}$ corresponds to 1, c, 2, and 3 nodes of graph G_n respectively. If node c is removed, as shown in (a)(iii), there is no connection between nodes 1, 2, and 3. So, $M_{(2,2)}$ is an LCP. On the other hand, in Figure 1(b)(iii), even if we remove node c, there exists a path connecting nodes 1, 2, 3 and 4, and therefore $M_{(2,2)}$ is not an LCP.

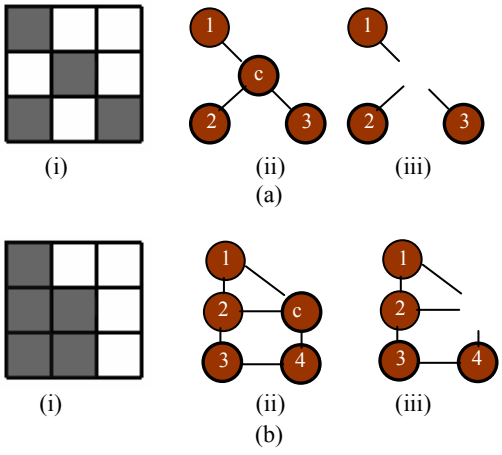


Figure 1: (i) 3×3 Pixel matrices, M; (ii) Graphs of matrices represented as nodes and edges; (iii) Graphs of matrices with nodes 'c' removed.

2.1 Preprocessing

All gray scale images are binarized with the use of the modified Niblack algorithm (Chandra et al., 2007). This algorithm works fine for handwritten-text images. Once we obtain the binarized image, a noise removal algorithm is applied to clean the spatial noise (small components having size of 1-3 pixels). A simple 5x5 morphological filter is applied to achieve this.

2.2 Identification of Local Coupling Points

Identifying an LCP involves the following steps:

i. Calculate the number of adjacent Black&-White-pixels pairs.
ii. If 3×3 pixels matrix has more than one adjacent black-&-white-pixels pair, then apply template matching. Template matching is explained in section 2.2.2.

2.2.1 Black-&-White-Pixel Pairs

In a 3×3 mask, we represent all the elements of $M_{(x,y)}$, except $M_{(2,2)}$, as a vector, V_n , as shown in Figure 2. Next, we count the number of transitions from '0' to '1' or '1' to '0' ('1' represents black and '0' represents white) considering each pixel only once. As shown in figure 2, there are three transitions.

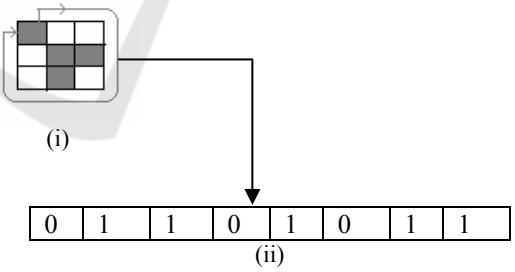


Figure 2: (i) 3x3 Pixels matrix, M; (ii) Vector representation, V_n of M.

2.2.2 Template Matching

Template matching for $M_{(x,y)}$ can be done by two methods. In the first method, an exhaustive search is done, where $M_{(x,y)}$ is matched with predefined templates, pixel by pixel, as shown in Figure 3. If a perfect match is found, $M_{(2,2)}$ is not an LCP. Such an exhaustive search is not efficient. Improvement can be achieved by categorizing the templates on the basis of number of black pixels that surround $M_{(2,2)}$.

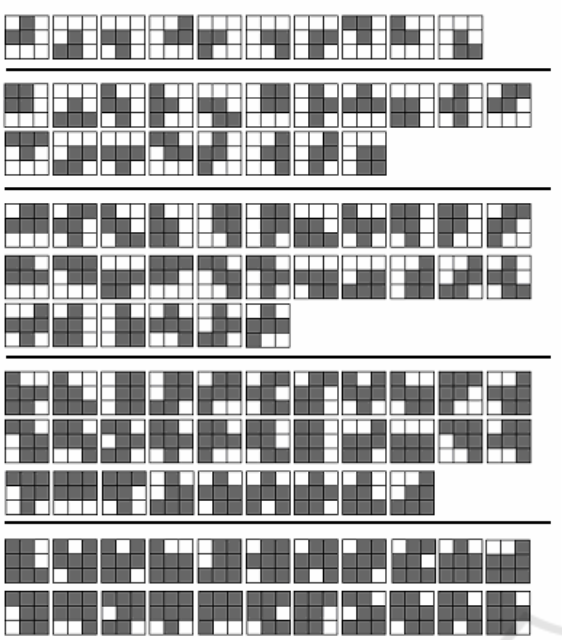


Figure 3: LCP templates based on the number of connected pixels; first set of templates with two black pixels connected to the central pixel, second set with three connected pixel, and so on.

The cases where seven surrounding pixels are connected to the central pixel is not considered because for such a case all the possible templates will not be LCP.

The alternate method is to represent M(x,y), in the form of a number between 0-255, and check the value in the Lookup table Li (Table 1) to determine whether the pixel under consideration is an LCP. This is discussed next.

Assume that M(x,y)=0 denotes a white pixel, and M(x,y)=1 denotes a black pixel. Represent exterior pixels of M(x,y), using byte data type, B, where each bit, bi, of B corresponds to the exterior pixel of M(x,y), that is b1, b2, b3, b4, b5, b6, b7, and b8 correspond to M(1,1), M(1,2), M(1,3), M(2,3), M(3,3), M(3,2), M(3,1), and M(2,1) respectively. Calculate the decimal equivalent, D, of B. Look for the value at Dth index in Lookup Table, Li. If LD is 1, M(2,2) is an LCP, otherwise not. For example, as shown in Figure 2, binary representation of B is 01101011_b, and equivalent decimal representation, D, is 10710_d. As L107 is 0, M(2,2) is not an LCP.

For thinning, all the pixels, which lie on the contour of the image, are removed. For the strokes that have single pixel width, single-pixel lines are preserved to prevent any discontinuity. This is discussed next.

In the first step, all the black pixels are classified either as boundary pixels or body pixels.

i. A *Boundary pixel* is one that is exposed to the background, i.e., a pixel having at least one white pixel among eight neighbourhood pixels (Top, Bottom, Left, Right, Top-Right, Right-Bottom, Bottom-Left, and Left-Right).

ii. A *Body pixel* is one that is completely surrounded by the eight black pixels in its immediate neighbourhood.

In the second step, LCPs are identified among boundary pixels. All Boundary pixels and LCPs are labelled as black pixels, as shown in Figure 5 and Figure 6 respectively.

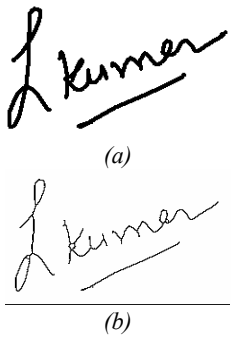
The third step involves performing a row-wise scan, and removing those boundary pixels, which are adjacent to the body pixel and are not an LCP.

These three steps are performed iteratively until no body pixels can be found, or further boundary pixels cannot be removed.

The pseudo code for this algorithm is presented below:

Algorithm1: Basic thinning pseudo code.

```
Mark all the boundary pixels and body pixels
While ((body pixels exist) And (Boundary pixels can be removed))
    If pixel under consider (is adjacent to body pixel)
    And (Not a LCP) Then
        Remove the pixel under consideration.
    Else
        Do not remove the pixel
    End If
End While
```



(a)

(b)

Figure 4: (a) Input image (b) Thinned image.

The method results in a thinned image as shown in Figure 4. However, some applications need one-pixel-thick image that can be segmented in disjoint polylines. This is discussed next.

For a pixel to be categorised as body pixel, the constraints are relaxed. If a pixel is surrounded by

three black pixels from four immediate neighbourhood pixels (Top, Bottom, Left, Right), it is categorised as a body pixel. Subsequently, T-shaped body pixels are obtained, as shown in Figure 5(a).

Next, an iterative row scan is performed, and boundary pixels, adjacent to T-shaped body pixels, are removed until no T-shaped body pixel exists or further boundary pixels cannot be removed.

The more the body pixel condition is relaxed, more thinned image is obtained. If the condition is relaxed so that a pixel is considered a body pixel if it is connected to two neighbourhood pixels (one pixel at Top or Bottom, and other at Left or Right), *L-shaped* pixels are obtained, as shown in Figure 5(b). In this manner, a single-pixel-thinned image is obtained.

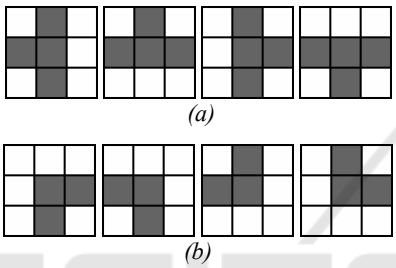


Figure 5: (a) T-shaped pixels (b) L-shaped pixels.

Algorithm 2: Single-pixel thinning pseudo code.

Mark all the *boundary pixels* and *body pixels*
b_pixels ← *body pixels*
While (*(b_pixels exist)* And *(Boundary pixels can be removed)*)
If pixel under consideration *(is adjacent to body pixel)* And *(Not an LCP)* Then
Remove the pixel under consideration.
Else
Do not remove the pixel
End If
End While
Mark all the *boundary pixels* and *T-shaped pixels*
b_pixels ← *T-shaped pixels*
*G*o to *step2*
If (*no more boundary pixels can be removed*)
Remove all the *T-shaped* body pixels
End If
Mark all the *boundary pixels* and *L-shaped pixels*
b_pixels ← *L-shaped pixels*
Go to *step2*
If (*no more boundary pixels can be removed*)
Remove all the *L-shaped* body pixels
End If

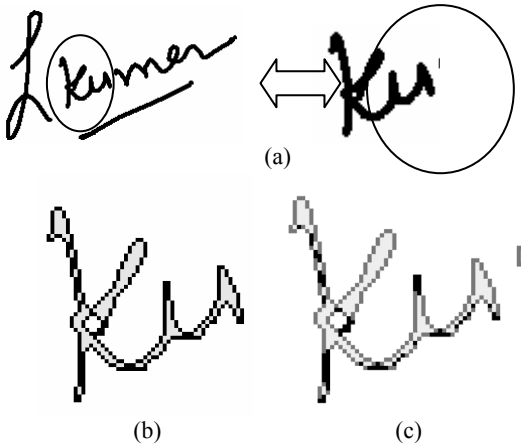


Figure 6: (a) Original image; (b) Boundary pixels, marked as black, are checked whether they are LCP. If a boundary pixel is not a LCP, then it is removed; (c) LCP pixels marked as black, and rest of the boundary pixels to be removed marked as gray. The complete process is repeatedly performed until non-LCP boundary pixels exist.

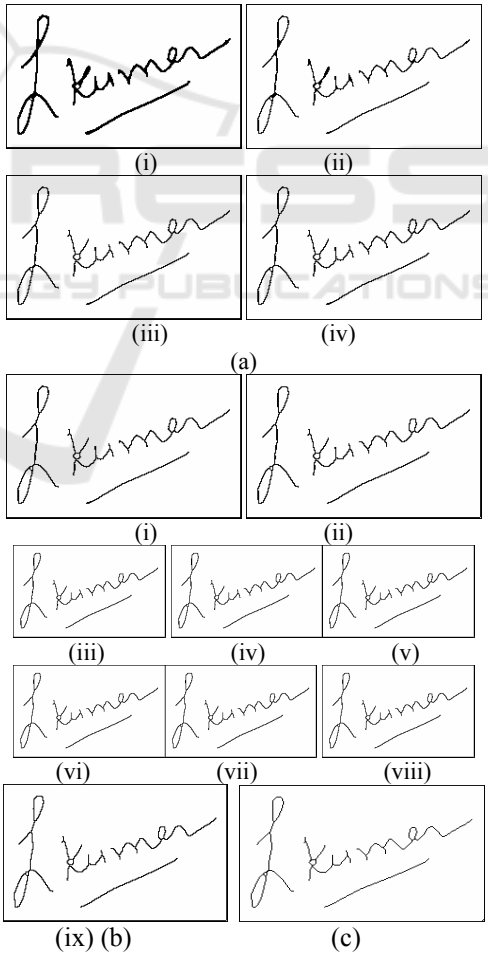


Figure 7: (a) Pass I, steps (i) to (iv); (b) Pass II, steps (i) to (ix); (c) Pass III.

The results, for every step of each pass of the algorithm on a signature image, are shown in Figure 7. As can be seen, the algorithm takes four iterations to complete Pass I. Pass II consist of nine iterations to remove T-shaped boundary pixels. Finally, pass III takes a single iteration to remove L-shaped pixels.

3 EXPERIMENTAL RESULTS

Thinning algorithms are usually evaluated on four parameters: connectivity preservation, skeleton width, robustness to noise, and information preservation. In addition to the above parameters, we consider some other important aspects, such as preservation of edges and robustness of the algorithm to complex structures.

A large number of thinning algorithms that have been proposed are inflicted with basic problems. For example, the output of Datta's algorithm (Datta and Parui, 1994) has spurious branches. Another example is Han's algorithm (Han et al., 1997), which certainly is an improvement over Datta's algorithm as it can remove spurious branches, but skeleton width is the major problem. Moreover the test patterns taken for Han's are simple machine printed characters that are less prone to noise and irregular shapes at the ends. Further, the algorithm by Lei Huang (Hunag et al., 2003) effectively handles both the above problems, however, it is not able to preserve the shapes of more complex images, such as handwritten signatures. It also does not offer user the flexibility of selecting the skeleton width.

The approach presented in our paper solves a number of potential problems including those mentioned previously, such as spurious branches, noise, shape preservation, etc., while at the same time it effectively and efficiently addresses requirements of custom skeleton width selection and thinning of complex structures.

In Figure 8, some handwritten signature images and their corresponding thinned images, obtained by our method, are shown.

The proposed algorithm is highly efficient. It is easy to implement using programming languages, such as C, which support bit-wise operators. The algorithm's superior performance is due to the use of two unique approaches, a) Vector representation followed by matching with look-up table, and b) template matching. Leveraging these two techniques, the algorithm provides unmatched flexibility in processing images containing handwritten signatures. The method has been tested

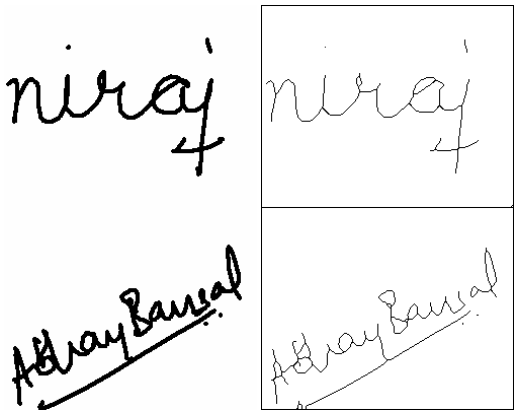


Figure 8: Original and thinned images.

with Grupo de Procesado Digital de Senales" (GPDS) Signature Database (2016 images) (Martinez et al., 2004). The average time taken for the random set of 100 images (300x250) on Pentium IV 2.5 GHz, 512MB RAM is approximately 9 seconds.

4 CONCLUSIONS

This paper presents an approach for thinning as well as shape preservation of more complex structures, such as handwritten signatures. The result presented here clearly demonstrates that the proposed method/algorithm is not only time efficient but also preserves information.

REFERENCES

Ahmed, S., Sharmin, M. & Chowdhury Mofizur Rahman, 2002. A Generic Thinning Algorithm with Better Performance, *Fifth International Conference on Computer and Information Technology*.

Ammar, M., Yoshida, Y. & Fukumura T., 1986. A new effective approach for off-line verification of signatures by using pressure features. In: *Proceedings of the International Conference on Pattern recognition*.

Chandra, Lal, Lal, Puja, Gupta, Raju, Tayal, Arun & Ganotra, Dinesh, 2007. Improved Adaptive Binarization Technique for Document Image Analysis, *2nd International Conference on Computer Vision Theory and Applications*.

Cowell, Dr John, Hussain, Dr. Fiaz, 2003, Amharic Character Recognition using a Fast Signature Based Algorithm, *Proceedings of the Seventh International Conference on Information Visualization.*

Datta, A. & Parui, S.K., 1994. A Robust Parallel Thinning Algorithm for Binary Images. *Pattern Recognition*, Vol. 27, No. 9 pp.1181-1192.

El Mesbahi, J., Chaibi, M, 1993. Printed circuit boards inspection using two new algorithms ofdilatation and connectivity preserving shrinking, Neural Networks for Signal Processing (1993) III. *Proceedings of the 1993 IEEE-SP Workshop.*

Han, N.H., La, C.W. & Rhee, P.K., 1997. An Efficient Fully Parallel Thinning Algorithm, *Proceedings of the 4th International Conference on Document Analysis and Recognition,* Page 137, ISBN 0-8186-7898-4.

Hang, Z. & Wang, Y.Y., 1994. A New ParallelThinning Methodology, *IJPRAI(8)*, pp.999-1011.

Hunag, Lei, Huang, Genxun & Liu, Chnagping, 2003. An improved parallel thinning algorithm. *Proceedings 7th ICDAR Conference.*

Lecce, V.Di, Dimauro, G., Guerriero, A., Impedovo, S., Pirlo, G. & Salzo, A., 2000, A new hybrid approach for legal amount recognition, In: L.R.B. Schomaker and L.G. Vuurpijl (Eds.), *Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition,* Amsterdam, ISBN 90-76942-01-3, Nijmegen: International Unipen Foundation, pp 199-208.

Martinez, L.E., Travieso, C.M., Alonso, J.B. & Ferrer, M., 2004. Parametrization of a forgery Handwritten Signature Verification using SVM, *IEEE 38th Annual 2004 International Carnahan Conference on Security Technology*, pp. 193-196.

Mehtre, B., 1993. Fingerprint image analysis for automatic identification, *Machine Vision and Applications*, vol. 6, pp. 124–139.

Neuslds, Christian & Olszewski, Jan, 1994. A noniterative Thinning Algorithm, *ACM Transactions on Mathematical Software*, Vol. 20, No. 1, March 1994, Pages 5-20.

Ubeda, V.,1993. A Parallel Thinning Algorithm Using KxK Masks, *IJPRAI(7)*, pp. 1183-1202.

APPENDIX

Table 1: Detailed Lookup table snippet.

Decimal	Binary	LCP	Decimal	Binary	LCP
0	00000000	×	20	00010100	√
1	00000001	×	21	00010101	√
2	00000010	×	22	00010110	√
3	00000011	√	23	00010111	√
4	00000100	×	24	00011000	√
5	00000101	√	25	00011001	×

Table 2: Detailed Lookup table snippet.

Decimal Representation	Is LCP
0-2, 4, 8-11, 16-19, 25-27, 32-47, 49-51, 57-59, 64, 66, 68, 70, 72-76, 78, 80, 82, 85, 88, 90, 96, 98, 100, 102, 104-108, 110, 114, 117, 121-122, 127-128, 130, 132, 134, 136-140, 142, 144-148, 150, 152-156, 158, 160-180, 182, 184-188, 190-191, 194, 196, 198, 200-204, 206, 210, 218, 223, 226, 228-232, 242, 250	×
3, 5-7, 12-15, 20-24, 28-31, 48, 52-56, 60-63, 65, 67, 69, 71, 77, 79, 81, 83-84, 86-87, 89, 91-95, 97, 99, 101, 103, 109, 111-113, 115-116, 118-120, 123-126, 129, 131, 133, 135, 141, 143, 149, 151, 157, 159, 181, 183, 189, 192-193, 195, 197, 199, 205, 207-209, 211-217, 219-222, 224-225, 227, 233-241, 243-249, 251-255	√