Thinning algorithms comparison for vectorization of engineering drawings

Matúš Gramblička[1, a], Jozef Vaský[1, b]

[1]SLOVAK UNIVERSITY OF TECHNOLOGY in Bratislava FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA PAULÍNSKA 16,917 24 TRNAVA, SLOVAK REPUBLIC

[a]matus.gramblicka@stuba.sk, [b]jozef.vasky@stuba.sk

Abstract

The thinning algorithms are used for a creation the skeleton of an object. The thinned image consists of the lines one pixel wide. The thinning or skeletonization reduces the image complexity. In this article the comparison of nine iterative parallel thinning algorithms and the one proposed is presented and their performance evaluation on sets of the engineering drawings. The results are evaluated in regard to suitability to vectorization of engineering drawings. The vectorization of raster image is a conversion process of a raster image consisting of pixels to image that consists of vectors. The thinning process is widely used in vectorization based on the thinning methods.

Keywords: thinning algorithms, skeleton, vectorization, engineering drawings

Introduction

The vectorization is transformation process of the raster image consisting of pixels to vector representation consisting of vectors. The vectorization methods can be very roughly divided into two groups, thinning and non-thinning vectorization methods. Vectorization methods based on thinning yield better results in terms of the line geometry, but natively do not preserve line width. In this paper it is examined ten thinning algorithms on the set of engineering drawings and evaluated which methods are most suitable for the further vectorization of these drawings.

Thinning or skeletonization is technique commonly used in raster vector conversion [97], [119], [151], [152]. Thinning is the process of identifying the skeleton of an object. The thinned version of a shape is called the skeleton. The skeleton is basically a central line extraction of an object, resulting from thinning [107], [63]. A skeleton captures essential topology and shape information of an object in a simple form [170]. Thinning is a very important preprocessing step for the analysis and recognition of different types of images. Thinning technique can be divided into two groups. The first group is based on an iterative thinning of the original image using a boundary erosion process [63]. All thinning algorithms used in this paper belong to this group. The iterative process removes pixels while sequence of pixels one pixel wide remains. The second group of thinning techniques is based on distance transform [120] to compute the skeleton of an object. The iterative approaches need more passes before attaining the final result, so the computing time can be high [65]. They are also sensitive to presence of a noise. The second category of approaches based on the distance transform gives results more efficiently. Works [100], [121] for instance use this second thinning approach.

The scanned line images are composed of segments multiple pixels wide. As a result, conventional approaches commonly used morphologic thinning techniques for reducing the complexity of the data. The purpose of the thinning operations is to reduce the original pixel width to one pixel wide skeleton that retains the essential features of the original image. The skeleton of the digital object may be used in many applications to allow easier structural analysis and for more intuitive design of the recognition algorithms. Skeletons are important shape descriptors in object representation and recognition.

A skeleton is very useful in the fields such as a 3D model matching and retrieval, medical image analysis [170], bubble-chamber image analysis, text and handwriting recognition and analysis. The representation with the thin lines is more suitable for an extraction of the endpoints, the connecting points and the connections between the components [94].

The thinning is also a key step of the preprocessing in many systems of the pattern recognition. In the OCR applications an extraction process usually follows the thinning process. Thinning also plays an important role in reducing data complexity for the vectorization of the binary line drawings such as electrical circuit diagrams, maps and architectural blueprints, mechanical and technical drawings [61] and for the recognition of characters and the other objects [193].

The thinning algorithms have good accuracy of the line location however they tend to add a number of hooks, when the image is not regular. Moreover, the thinning algorithms are poor in terms of correct positioning of the junctions and the end points. The original line width recovery is straightforward using the methods based on distance transform, but not using an iterative thinning [112]. Thinning usually introduces incorrect connecting points. Short branches connected through these connection points can deform the shape of the skeleton and oppose further recognition and vectorization [96]. The frequent access to the pixels at iterative thinning slows the rate of a vectorization [64].

The examples of specific thinning algorithms used for thinning the raster images that are not tested in this paper may be mentioned Rosenfeld's parallel algorithm used in the vectorization application AutoTrace, voronoi diagrams, Wang Zhang and Holt thinning algorithms. As the examples of the newer algorithms these works can be mentioned [179], [162], [195], [190*].

A good thinning algorithm should accomplish the following requirements: maintaining connectivity of the skeleton, producing one pixel wide skeleton, to be resistive to the noise, and time-efficient. The reasons and need for a thinning of the images can be formulated as:

a) Reduction of the amount of data required to be processed.
b) Reduction of the time required to process the pattern.
c) Shape analysis can be made more easily on the thinned pattern.

The thinning algorithms can be classified [1], [180], [188]:

- Iterative (pixel based)
    - Sequential
    - Parallel
- Non-Iterative (non-pixel based)
    - Medial axis transform
    - Line following
    - Other

Iterative parallel thinning algorithms

All tested thinning algorithms in this paper are iterative parallel algorithms. Whether the pixels will be deleted in n^{th} iteration depends in parallel thinning on the result from the previous $(n-1)^{th}$ iteration. Values of the pixels and its neighbors at the $(n-1)^{th}$ iteration determine the values of the pixels at the next n^{th} iteration. Parallel thinning algorithms usually use a 3*3 matrix that represents neighborhood around the examined pixel as shown in table 1.

Table 1 Matrix 3x3 represents 8-neighboorhood of pixel P1

P9	P2	P3
P8	P1	P4
P7	P6	P5

For the purposes of iterative parallel thinning algorithms let´s propose matrix 3x3 and define the three functions B(P), A(P) and C(P).

B(P1) represents the number of non-zero neighbors of P1. It is computed as:

B(P1) = P2+P3+............................+P9.

A(P1) represents the number of 0,1 patterns in the sequence P2, P3, P4, P5, P6, P7, P8, P9, P2. Examples of these functions can be seen on fig 1.

1	1	0
0	P1	0
0	0	0

a)

0	0	1
0	P1	0
1	0	0

b)

Fig 1 Example of functions B(P1) and A(P1): a) B(p1) = 2 , A(p1) = 1; b) B(p1) = 2, A(p1) = 2

The function C(P) is connectivity number. C(P1) function is little harder to understand so fig. 2 is proposed. C(P1) is the number of distinct 8-connected components count in the neighborhood of the pixel P1. One of the ways how to compute function C(P1) can be:

$C(P1) = !P2 \wedge (P3 \vee P4) + !P4 \wedge (P5 \vee P6) + !P6 \wedge (P7 \vee P8) + !P8 \wedge (P9 \vee P2)$ [191].

1	1	1
0	P1	1
0	0	0

a)

1	1	0
0	P1	1
0	0	1

b)

0	1	0
1	P1	0
0	0	1

c)

1	0	1
0	P1	0
0	1	0

d)

Fig.2 Example of functions C(P1): a) C(P1) = 1;b) C(P1) = 1; c) C(P1) = 2; d) C(P1) = 3.

The Guo hall and Stentiford thinning algorithms use the function C(P).

Zhang Suen thinning algorithm

Zhang Suen thinning algorithm [186] (ZS) is fast parallel thinning algorithm with two sub-iterations. The conditions used in the first sub-iteration in order to remove the south-east pixel are:

1.$2 \leq B(P1) \leq 6$

2.$A(P1) = 1$

3.$P2 \wedge P4 \wedge P6 = 0$

4.$P4 \wedge P6 \wedge P8 = 0$

In the second sub-iteration conditions in steps 3 and 4 change, in order to remove the north-west pixels:

3.$P2 \wedge P4 \wedge P8 = 0$

4.P2 Λ P6 Λ P8 = 0

End points [181] and pixel connectivity should be preserved [1]. Issue with that algorithm is that with the presence of a noise near north-east and south-west corners these are extended instead of deleted [180].

Lu Wang thinning algorithm

The algorithm proposed by Lu and Wang [185] (LW) to solve problem of diagonal lines occurring in ZS algorithm. It is derivative of ZS. The only difference from ZS algorithm is that the first condition is replaced by condition: All pixels whose number of value is 1 in 8-neighborhood is in the range 3 to 6.

LW algorithm should shrink horizontal lines well, but doesn't get one pixel wide skeleton in sloping lines [182].

Modified thinning algorithm

Zhang and Wang [183] proposed Modified algorithm (MA). It uses slightly extended neighborhood as shown in table 2. Point P1 can be deleted if all conditions are met:

$2 \leq B(P1) \leq 6$

$A(P1) = 1$

(P2*P4*P8 = 0) or P11 = 1

(P2*P4*P6 = 0) or P15 = 1

MA should be faster than ZS [183].

Table 2 neighborhood of pixel P1 in algorithm MA

P10	P11	P12	P13
P9	P2	P3	P14
P8	P1	P4	P15
P7	P6	P5	P16

Kwon Woong Kang thinning algorithm

The algorithm proposed by Kwon, Woong and Kang [187] Enhanced parallel thinning algorithm (KWK) is concentrated to obtain skeleton of 1-pixel width as well as maintain 8-connectivity what are the problems of ZS and LW algorithms [180].

It is a two pass algorithm. On the image the first pass criterions are applied and then on the result the second pass criterions further produced the final skeleton.

In the Pass 1 steps one and two are the same as in ZS algorithm, expect that the step 2 uses condition that number of value 1 in 8-neighbour pixels are in the range 3 to 6 instead of 2 to 6.

In the second pass conditions are:

1. P9 = 1 Λ P8 = 1 Λ P6 = 1 Λ P3 = 0
2. P3 = 1 Λ P4 = 1 Λ P6 = 1 Λ P9 = 0
3. P5 = 1 Λ P6 = 1 Λ P8 = 1 Λ P3 = 0
4. P4 = 1 Λ P6 = 1 Λ P7 = 1 Λ P9 = 0

It can get the 1-pixel wide lines, handles the end point preserving and perfect 8-connectivity [182].

Hilditch thinning algorithm

Hilditch algorithm [192], [189] presented by Naccache and Shinghal, provides good results when thin the image edges [190*]. Hilditch algorithm consists of four conditions before deleting the pixel:

$2 \leq B(P1) \leq 6$

$A(P1) = 1$

$(P2 \wedge P4 \wedge P8 = 0)$ or $A(P2) \,!= 1$

$(P2 \wedge P4 \wedge P6 = 0)$ or $A(P4) \,!= 1$

Hilditch algorithm does not work on all patterns. For example, it removes staircase patterns almost completely.

Guo Hall thinning algorithm

Algorithm proposed by Guo and Hall [191] (GH) is two sub-iteration parallel thinning algorithm. The GH gives thinner skeletons than ZS and LW algorithms.

In that algorithm new function N(P) is introduces. It allows to preserve the end points as well as to remove redundant pixels. The function NP is defined:

$N(P) = Min(N1(P1), N2(P1))$; where

$N1(P1) = (P9 \vee P2) \wedge (P3 \vee P4) \wedge (P5 \vee P6) \wedge (P7 \vee P8)$

$N2(P1) = (P2 \vee P3) \wedge (P4 \vee P5) \wedge (P6 \vee P7) \wedge (P8 \vee P9)$

An edge point will be deleted if it satisfies these conditions:

1. $C(P1) = l$;
2. $2 \leq N(P1) \leq 3$;
3. Then apply one of the following depending of the iteration:
   a) $(P2 \vee P3 \vee P5) \wedge P4 = 0$ in odd iterations; or
   b) $(P6 \vee P7 \vee P9) \wedge P8 = 0$ in even iterations

Condition 1 preserves local connectivity that means the deletion of pixel P1 does not break the connectivity and guarantees that pixel P1 is not a break point [1]. Condition 3(a) deletes north-east pixels and 3(b) deletes south-west pixels [180]. The GH algorithm detects the end points better than the ZS algorithm [1].

Stentiford thinning algorithm

Stentiford and Mortimer proposed algorithm [194] that uses four templates for decision to remove pixels as shown at fig 3. White circle represents white pixel, black circle black pixel and the crossed squares means that on these pixels does not matter. Algorithm can be proposed with four sub-iterations. First and second conditions for pixel deletion are the same for all four sub-iterations:

1. $B(P1) > 1$
2. $C(P1) = l$
3. It is looked for pixel location where pixels match those in template T1.

In the next sub-iterations the only difference is that it is used next template T2 until T4. The algorithm repeats until there are no more pixels to delete.

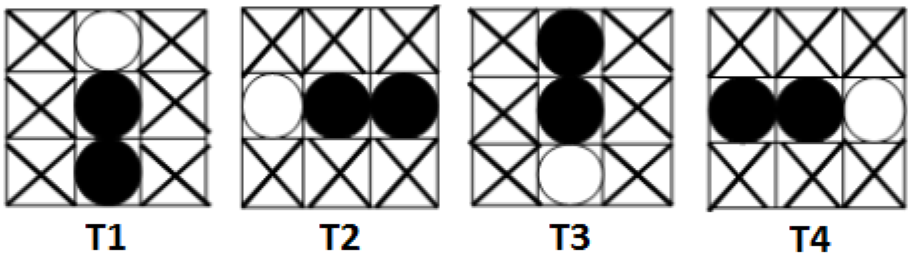


Fig. 3 Four templates used in the Stentiford thinning algorithm.

Arabic parallel thinning algorithm

Arabic thinning algorithm [192] uses four sub-iterations that are divided into 4 conditions. A dark point P1 is set for deletion if one of each sub-iteration is true. First and second conditions are the same for all sub-iterations:

1. $2 <= B(P1) <= 6$
2. $A(P1) = 1$

First sub-iteration

3. $P2 \wedge P4 \wedge P6 = 0$
4. $P4 \wedge P6 \wedge P8 = 0$

Second sub-iteration

3. $P2 \wedge P6 \wedge P8 = 0$
4. $P4 \wedge P6 \wedge P8 = 0$

Third sub-iteration

3. $P2 \wedge P4 \wedge P8 = 0$
4. $P2 \wedge P6 \wedge P8 = 0$

Fourth sub-iteration

3. $P2 \wedge P4 \wedge P6 = 0$
4. $P2 \wedge P4 \wedge P8 = 0$

Efficient Parallel Thinning Algorithm

Aparajeya and Sanyal proposed another parallel thinning algorithm [197]. There are 2 sub-iterations, the pixel deletion criterions are:

First sub-iteration

1. $!P4 \wedge P8 \wedge ((P2 \wedge !P5 \wedge !P6) \vee (!P2 \wedge !P3 \wedge P6))$ or
2. $P6 \wedge P7 \wedge P8 \wedge ((P9 \wedge P2 \wedge !P4) \vee (!P2 \wedge P4 \wedge P5))$ or
3. $!P2 \wedge !P3 \wedge !P4 \wedge (((!P5 \wedge !P6 \wedge P8) \wedge (P9 \vee P7)) \vee ((!P9 \wedge P6 \wedge !P8) \wedge (P7 \vee P5))) = 1$

Second sub-iteration

1. $P4 \wedge !P8 \wedge ((!P9 \wedge !P2 \wedge P6) \vee (P2 \wedge !P6 \wedge !P7))$ or
2. $P2 \wedge P3 \wedge P4 \wedge ((P5 \wedge P6 \wedge !P8) \vee (P9 \wedge !P6 \wedge P8))$ or
3. $!P6 \wedge !P7 \wedge !P8 \wedge (((!P9 \wedge !P2 \wedge P4) \wedge (P5 \vee P3)) \vee ((P2 \wedge !P4 \wedge !P5) \wedge (P3 \vee P9))) = 1$

This algorithm is the enhancement of the GH algorithm. The connectivity number is no longer computed. The algorithm provides good results if the input image is not very noisy.

Proposed thinning algorithm

The proposed algorithm has three passes where first two passes are identical with those in KWK algorithm. In the third pass the algorithm uses 5x5 neighborhood, table 3, and aims to delete corner pixels which do not form structure but are secondary thinning product of previous two passes.

In the third pass there are four sub-iterations with conditions:

1. P11 = 0 Λ P2 = 1 Λ P4 = 1 Λ P15 = 0 Λ P7 = 0
2. P15 = 0 Λ P4 = 1 Λ P6 = 1 Λ P19 = 0 Λ P9 = 0
3. P19 = 0 Λ P6 = 1 Λ P8 = 1 Λ P23 = 0 Λ P3 = 0
4. P23 = 0 Λ P8 = 1 Λ P2 = 1 Λ P11 = 0 Λ P5 = 0

Table 3 Matrix 5x5 represents 8-neighboorhood of pixel P1 in proposed thinning algorithm

P26	P10	P11	P12	P13
P24	P9	P2	P3	P14
P23	P8	P1	P4	P15
P22	P7	P6	P5	P16
P21	P20	P19	P18	P17

The first sub-iteration deletes south west corners. The second sub-iteration deletes north west corners. The third sub-iteration deletes north east corners and the fourth sub-iteration deletes south east corners.

Method

The thinning algorithms are evaluated on a set of engineering drawings. As was mentioned before good thinning algorithm should maintain connectivity of skeleton, produce one pixel wide skeleton, be resistive to noise, and time-efficient. Based on these requirements five qualities are measured: connectivity, thinning ratio, thinness, sensitivity and execution time.

Connectivity

The original image and the skeleton in an ideal case should have the same amount of the discrete points and the end points. In reality thinned image would have more of these elements. More of these points in favor to the skeleton reflect more disconnection. The equation 1 [250] expresses that relationship.

$$C = \sum_{i=0}^{size} S(P_i) \qquad 1$$

$$S(P_i) = \begin{cases} 1 \; if \; B(P_i) < 2 \\ \quad else \; 0 \end{cases}$$

where size = width * height of the image in pixels.

Formula $B(P_i) < 2$ can be understood that pixel is a discrete point or an end point. The connectivity value that is closer to value of the original image denotes more quality skeleton.

Reduction rate

The reduction rate expresses the rate of pixel reduction of the original image expressed as a percentage. The value of this reduction can be computed by equation 2.

$$R = \left(1 - \frac{T_t}{T_o}\right) * 100\% \qquad 2$$

where T_t is the number of pixels in the thinned image,

T_o is the number of pixels in the original image.

Higher the reduction rate value means higher amount of pixels that were deleted from the original image.

Thinness

The thinness of a skeleton can be computed by the equation 3 inspired by the work of Zhou, Quek and Ng [250]. It expresses that the skeleton which is not one pixel wide contains triangles composed of three black pixels around pixel P_i.

$$T = 1 - \frac{\sum_{i=0}^{size} Th(P_i)}{[max(width, heigh) - 1]^2/4} \quad 3$$

$$Th(P_i) = (P_i * P_9 * P_2) + (P_i * P_9 * P_8) + (P_i * P_8 * P_7) + (P_i * P_7 * P_6) + (P_i * P_6 * P_5) + (P_i * P_5 * P_4) + (P_i * P_4 * P_3) + (P_i * P_3 * P_2)$$

The thinness value that is nearer to 1 suggests thinner skeleton.

Sensitivity

The noise that is presented in the image can cause a deformation of the skeleton. The pixels that remained after thinning process as a product of noise can be difficult to recognize comparing to normal pixels. The total number of the cross points that remained after thinning should be good indicator of sensitivity thus the resistance to the noise. The cross point can be defined as a point that connect three or more branches. The sensitivity can be measured by equation 4 in this paper.

$$S = \sum_{i=0}^{size} S(P_i) \quad 4$$

$$S(P_i) = \begin{cases} 1 \ if \ A(P_i) > 2 \\ \quad else \ 0 \end{cases}$$

Formula $A(P_i) > 2$ can be understood as the number of branches outgoing from the pixel P_i. Lower sensitivity value thus the lesser amount of cross points means better sensitivity of the algorithm and better skeleton.

Execution time

The execution time is the time duration of the thinning operation. The time is started to record with the beginning of the actual thinning process until the end of the process. One algorithm is executed several times and the average time is the execution time.

Results

The thinning algorithms were implemented in C# programming language. All tests were performed in Windows operating system under Oracle virtual machine with 4 GB dedicated memory on the processor Intel i5 4670K.

The table 1 represents the comparison of the execution time. MA, Hilditch and Efficient are the three fastest thinning algorithms. The Stentiford and proposed thinning algorithms are the two slowest one. All algorithms except Stentiford and proposed perform under one second.

Table 1 Comparison of execution time

Thinning algorithms	Execution time (s)
ZS	0,756835
LW	0,699355
MA	0,576294
KWK	0,838189
Hilditch	0,584245
GH	0,955331

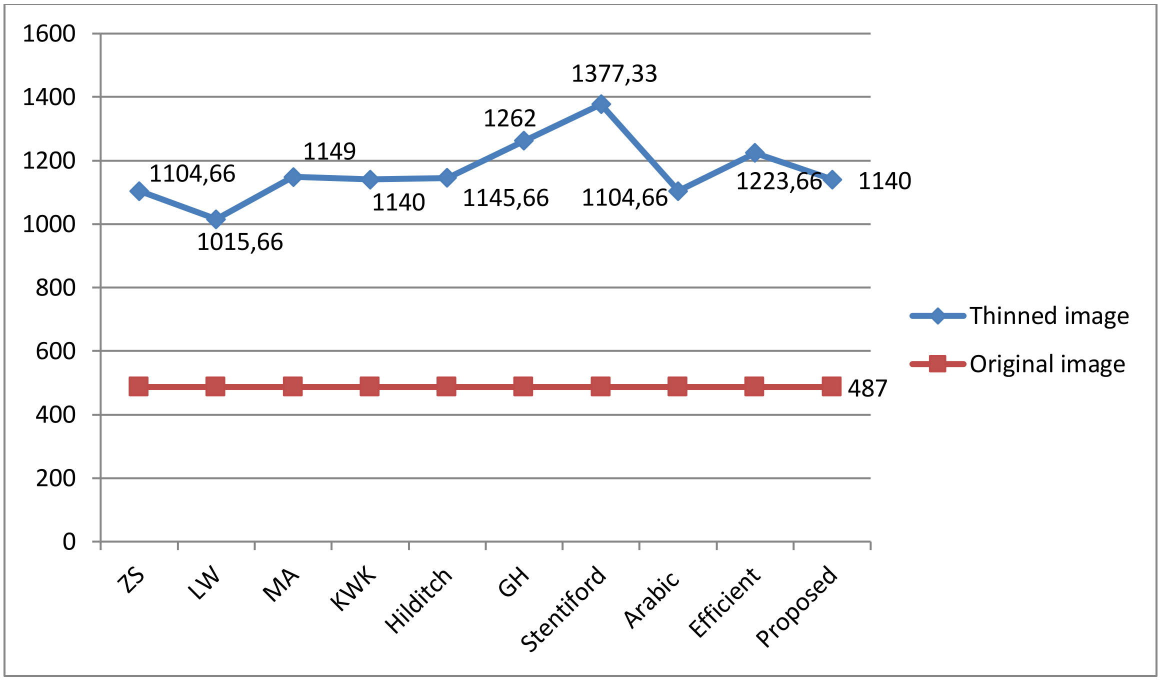
Stentiford	1,427934
Arabic	0,886065
Efficient	0,577033
Proposed	1,2088184

The table 2 represents the reduction rate and the thinness comparison of the thinning algorithms. The highest reduction rate is obtained by proposed algorithm with the third highest thinness value. The second highest reduction rate is attained by GH algorithm which also gets second highest thinness value. The KWK, Stentiford and Efficient algorithms get very good results in relation to reduction rate and thinness. Hilditch algorithm reaches the lowest reduction rate as well as the thinness value. The average reduction rate is 70,86% (69,48 – 71,78).

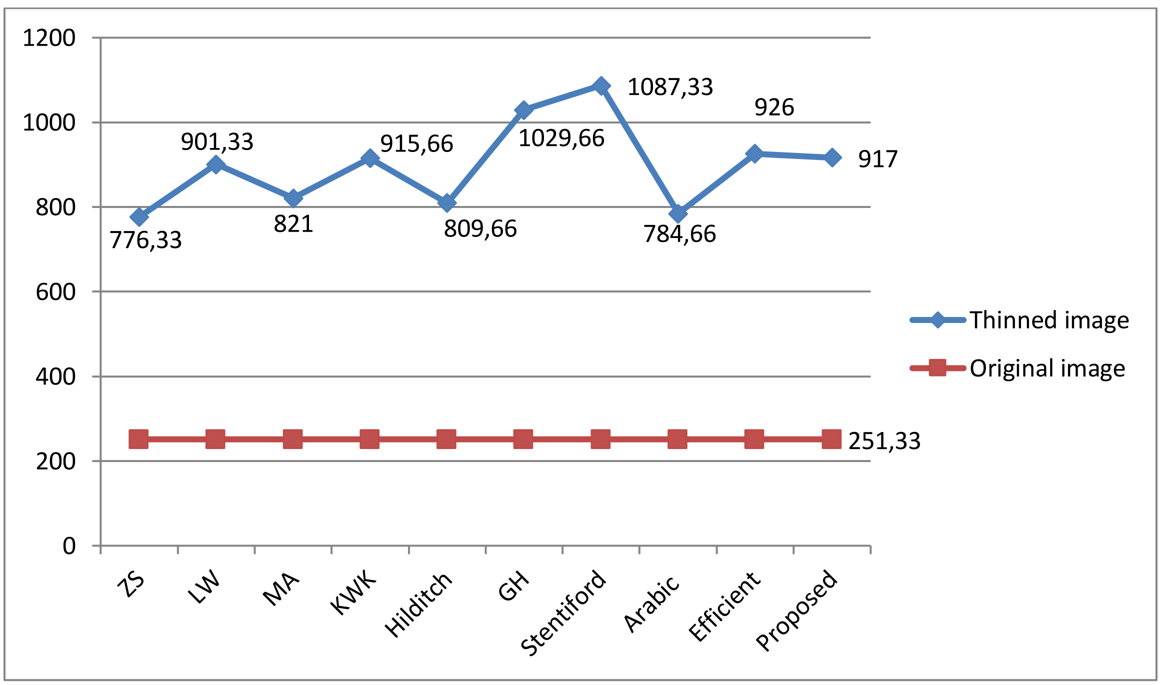
Table 2 Reduction rate and thinness comparison

Thinning algorithms	Original pixel count	Skeleton pixel count	Reduction rate (%)	Thinness
ZS	273389	80060	70,71	0,98730
LW	273389	82801	69,71	0,98639
MA	273389	79970	70,74	0,98690
KWK	273389	77704	71,57	0,99530
Hilditch	273389	83416	69,48	0,98222
GH	273389	77285	71,73	0,99730
Stentiford	273389	78959	71,11	0,99845
Arabic	273389	80157	70,68	0,98727
Efficient	273389	78921	71,13	0,99591
Proposed	273389	77134	71,78	0,99606

Graph 1 illustrates the connectivity comparison and graph 2 the sensitivity comparison of the thinning algorithms. It is important to notice that these results have to be judge in relation to the thinness and the reduction rate values. Stentiford algorithm has the best value of thinness and solid reduction rate but its connectivity value is the farthest from the original value and sensitivity value is the highest. These values express that there is more disconnection and more cross points after Stentiford thinning. On the contrary LW algorithm attains the closest connectivity value to original but second lowest reduction rate and thinness values. The sensitivity remains in the mid-range. It means that there is more connectivity at the expense of thinness. The Arabic thinning algorithm has solid connectivity and sensitivity results but the thinness is not so high. The most balanced algorithms in relation to thinness, connectivity and sensitivity are KWK and proposed algorithms.



Graph 1 Connectivity comparison. Lower value is better.



Graph 2 Sensitivity comparison. Lower value is better.

Based on the observation and the comparison of the thinned skeletons some findings can be concluded. Overall, the produced skeletons of engineering drawings are approximately same with small but important variations. ZW, LW, MA, Hilditch and Arabic thinning algorithms tend to produce double diagonal lines while KWK, GH, Stentiford, Efficient and proposed do not. KWK, MA, Arabic, Hilditch, proposed and ZW algorithms produce less spurs and hooks than other algorithms. ZW and Arabic thinning do not preserve the end points in some cases.

Conclusion

In this paper ten thinning algorithms were tested in order to examine their suitability for the further vectorization of the engineering drawings. In general all tested thinning algorithms were able to thin the engineering drawings to the level applicable for the vectorization. Since ZW, LW, MA, Hilditch and Arabic thinning algorithms remain double diagonal lines they appear to be less suitable for vectorization than KWK, GH, Stentiford, Efficient and proposed algorithms. From these five algorithms KWK and proposed produce less amount of the spurs and the hooks, have a better connectivity, the end points preservation and therefore they appear as the most suitable thinning algorithms among the others for the vectorization of the engineering drawings. The proposed algorithm was derived from KWK algorithm to reduce corners pixels so it even more softens the skeleton structure which is beneficial for the vectorization process. The drawback of the proposed algorithm is his slowness.

The best two algorithms for vectorization are KWK and proposed because of its most balanced results. Although whichever the algorithm would be utilize to thin the engineering drawings in order to vectorize them after the vectorization certain processes have to be used for recover lost connectivity and for adjusting geometry e.g. the gap filling and the polygonal approximation. The vectorization process is simpler if there are fewer pixels on skeleton. In case of deletion some pixels which should not be deleted, the gap filling should be used after the vectorization and the polygonal approximation for ablation the aberrations caused by the thinning.

Acknowledgement

This publication is the result of implementation of the project: "UNIVERSITY SCIENTIFIC PARK: CAMPUS MTF STU - CAMBO" (ITMS: 26220220179) supported by the Research & Development Operational Program funded by the EFRR.

References

[1] Kumar, H. Kaur, P. A Comparative Study of Iterative Thinning Algorithms for BMP Images / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (5) , 2011, 2375-2379

[97] R.D.T. Janssen and A.M. Vossepoel, Adaptive Vectorization of Line Drawing Images, Computer Vision and Image Understanding, vol. 65, no. 1, pp. 38-56, 1997.

[119] Kasturi R, Bow ST, El-Masri W, Shah J, Gattiker JR, Mokate UB (1990) A system for interpretation of line drawings. IEEE Trans Pattern Anal Mach Intell 12(10):978–992

[151] E. Bodansky and M. Pilouk, “Using Local Deviations of Vectorization to Enhance the Performance of Raster-to-Vector Conversion Systems,” Int’l J. Document Analysis and Recognition, vol. 3, no. 2, pp. 67-72, Dec. 2000.

[152] V. Nagasamy and N.A. Langrana, “Engineering Drawing Processing and Vectorization System,” Computer Vision, Graphics and Image Processing, vol. 49, no. 3, pp. 379-397, 1990.

[63] Lam, L., Lee, S.W., Suen, C.Y, 1992. Thinning Methodologies- A Comprehensive Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 14(9), 869-885

[65] Tombre, K., Tabbone., S., 2000. Vectorization in Graphics Recognition: To Thin or not to Thin, International Conference on Pattern Recognition (ICPR’00)- 2, Barcelona, Spain

[120] Niblack CW, Gibbons PB, Capson DW (1992) Generating skeletons and centerlines from the distance transform. CVGIP: Graph Models Image Process 54(5):420–437

[121] Kolesnikov AN, Belekhov VV, Chalenko IO (1996) Vectorization of raster images. Pattern Recognit Image Anal 6(4):786–794
[100] X. Hilaire and K. Tombre, “Robust and Accurate Vectorization of Line Drawings,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, no. 6, pp. 890-904, June 2006.
[112] Llados, J., Rusinol, M. Handbook of Document Image Processing and Recognition. Springer-Verlag London. Editors: Doermann, D., Tombre, K. kapitola Graphics Recognition Techniques. Pages 489-521. 2014. ISBN 978-0-85729-858-4. DOI 10.1007/978-0-85729-859-1
[170] Lakshmi, J. K., Punithavalli, M. A Survey on Skeletons in Digital Image Processing. Proceeding ICDIP '09 Proceedings of the International Conference on Digital Image Processing, pages 260-269, IEEE Computer Society Washington, DC, USA, 2009. ISBN: 978-0-7695-3565-4, doi 10.1109/ICDIP.2009.21
[94] J.Y. Chiang et al., A New Algorithm for Line Image Vectorization, Pattern Recognition, vol. 31, no. 10, pp. 1541-1549, 1998
[96] S. J. Tue and John Y. Chiang, The archiving of line drawing images, Proc. SPIE, SPIE’s Photonics East’95 Sympo.Conf.on Digital Image Storage and Archiving Systems, Philadelphia, PA, USA, Vol. 2606, pp. 145–156 (1995)
[107] Girija Dharmaraj. Algorithms for Automatic Vectorization of Scanned Maps. THE UNIVERSITY OF CALGARY. DEPARTMENT OF GEOMATICS ENGINEERING. CALGARY, ALBERTA. JULY, 2005
[64] Song, J., Su, F., Tai, C.L., Cai, S., 2002c. An Object-Oriented Progressive-Simplification-Based Vectorization System For Engineering Drawings: Model, Algorithm, and Performance, IEEE Transactions on Pattern Analysis and Machine Intelligence 24(8), 1048-1060
[61] Xu, X. W., Bai, Y. B., 2000. Computerising Scanned Engineering Documents, Computers In Industry 42(2000), 59-71
[162] Jia Liang. A Research on Guided Thinning Algorithm and Its Implementation by Using C#. International Journal of Image Processing (IJIP), Volume (5) : Issue (5) : 2011 580. 2011
[179] Gulshan Goyal, Maitreyee Dutta, Akshay Girdhar, “A Parallel Thinning Algorithm for Numeral Pattern Images in BMP Format”, International Journal of Advanced Engineering & Application, Jan. 2010, pp. 197-202
[180] Jagna, A. Some algorithms for image thinning using spatial domain processing. Jawaharlal Nehru Technological University, India, 2012. URI: http://hdl.handle.net/10603/3466
[181] Subashini P., Jansi S. Optimal Thinning Algorithm for detection of FCD in MRI Images. International Journal of Scientific & Engineering Research Volume 2, Issue 9, September-2011 1 ISSN 2229-5518
[182] Devi, H.K.A., (2006). Thinning: A Preprocessing Technique for an OCR System for the Brahmi Script. Ancient Asia. 1, pp.167–172. DOI: http://doi.org/10.5334/aa.06114
[183] Zhang, Y. Y.; Wang P. S. P. A modified parallel thinning algorithm. Pattern Recognition, 1988., 9th International Conference on Year: 1988 Pages: 1023 - 1025 vol.2, DOI: 10.1109/ICPR.1988.28429
[185] Lu, H.E., Wang, P.S.P. A Comment on “A Fast Parallel Algorithm for Thinning Digital Patterns”, comm. AMC, vol.29, no. 3, pp.239-242. 1986
[186] Zhang, T. Y., Suen, C. Y. A fast parallel algorithm for thinning digital patterns, Comm. ACM, vol. 27, no. 3, pp. 236-239. 1984
[187] Kwon, J. S, Woong J., Kang, E. K. An Enhanced Thinning Algorithm Using Parallel Processing, IEEE, pp.no.752-755. 2001

[188] E. Hastings. A Survey of Thinning Methodologies, Pattern analysis and Machine Intelligence, IEEE Transactions, vol. 4, Issue 9, 1992, pp. 869-885.
[189] Widiarti, A., R. “Comparing Hilditch, Rosenfeld, Zhang-Suen, and Nagendraprasad -Wang-Gupta Thinning”, International Scholarly and Scientific Research & Innovation, Vol:5, 2011
[190*] Thulasi, S., S. Jyothi, R., L. A Novel Image Thinning Method for Signature Recognition System. Department of Computer Science and Engineering, College of Engineering Karunagappally. 2016
[192] Naccache, N., J. Shinghal, R. An investigation into the skeletonization approach of Hilditch. Journal Pattern Recognition archive Volume 17 Issue 3, 1984 Pages 279-284. doi>10.1016/0031-3203(84)90077-3
[191] Guo, Z. Hall, R., W. Parallel thinning with two-subiteration algorithms, Comm. ACM, vol. 32, no. 3, pp. 359-373. 1989
[192] Tellache, M., Sid Ahmed, M., A., Abaza B. Thinning Algorithms for Arabic OCR. Communications, Computers and Signal Processing, 1993., IEEE Pacific Rim Conference on, Volume: 1, 19-21 May 1993 Pages: 248-251 vol. 1
[193] Lopich, A., Dudek, P. "Hardware implementation of skeletonization algorithm for parallel asynchronous image processing", journal of signal processing system, July 2009.
[194] Stentiford, F., W., M., Mortimer, R., G. Some New Heuristics for Thinning Binary Handprinted Characters for OCR, R.G. 1 IEEE Transactions on Systems, Man, and Cybernetics, SMC-13 (1983), p. 81
[195] Abu-Ain, W. et al. and col. Skeletonization Algorithm for Binary Images. The 4th International Conference on Electrical Engineering and Informatics
(ICEEI 2013)
[250] Zhou, R., W., Quek, C., Ng, G., S. A novel single-pass thinning algorithm and an effective set of performance criteria. Pattern Recognition Letters 16 (1995) 1267-1275
[197] Aparajeya, P., Sanyal, S. AN EFFICIENT PARALLEL THINNING ALGORITHM USING ONE AND TWOSUB-ITERATIONS.12th IASTED International Conference on Computer Graphics and Imaging (CGIM-2011)