A note on "Building Skeleton Models via 3D Medial Surface/Axis Thinning Algorithms"

2 authors:

Manuel G Forero
Universidad de Ibagué

Camilo Andres Murillo Ruiz
Universidad de Ibagué

A note on "Building Skeleton Models via 3D Medial Surface/Axis Thinning Algorithms"

Manuel G. Forero, Camilo A. Murillo
Universidad de Ibagué,
manuel.forero@unibague.edu.co
camiloandresmurilloruiz@gmail.com

Abstract

Skeletonization is one of the most-used techniques in object recognition and tracking; the one developed by Lee et al. being one of the most popular and used. However, this article shows that it does not always produce complete skeletons, finding the causes of the error and presenting a solution allowing to find complete skeletons in all situations.

1. Introduction

One of the most-used methods in the area of machine vision for object recognition is skeletonization. It seeks to obtain a continuous pattern, representing the original object with the least amount of data possible. The skeleton is the smallest representation of a set of lines formed by either a width of one pixel (2D) or a voxel (3D). To carry out this process, different methods are available. For 2D, one of the most-used methods consists of iteratively eliminating the pixels of the object that contact the background until the middle axis of the object is found. It should be one-pixel thick. In the 3D case, there are two types of skeletonization: the first one given by the central axis of the object, consisting of the minimum set and continuous voxels located equidistant from the edges of the object and the second one is defined by the surface half of the object, which includes the central axis.

Skeletonization is used in different applications, like computer animation, vision and pattern recognition, robotics, navigation, and medicine. 3D skeletonization is a complex technique that requires a high number of operations, which makes it computationally slow and difficult to process. Some methods are based on the 3D distance transform, which has a high computational load. While others are based on thinning, Palagyi et al. developed some methods, although computationally complex [1-5]. The technique created by Lee et al. is also based on thinning and it owes its popularity to its speed, compared to other techniques, because it divides the volume (given by the 26-neighbors around each voxel to be considered for elimination) in octants, making it more efficient [6]. This technique has been implemented for the Insight Segmentation and Registration Toolkit (ITK) [7], ImageJ [8], and Matlab [9] to obtain the 3D medial axis. An example is shown in Fig. 1.

(a) (b)

Figure 1. Example of skeletonization

The method scans the 3D image in 6 directions (Fig. 2) and, repetitively, removes the voxels that satisfy the following conditions:

I. It is a simple point, i.e., if it is deleted, the number of objects inside the 26-neighbors volume remain constant. This term has been used in thinning operations to preserve topological properties like the number of connected objects, cavities, and holes [10-12].
II. The Euler characteristic remains constant after the voxel deletion [13, 14].
III. The number of neighbors is greater or equal to one.

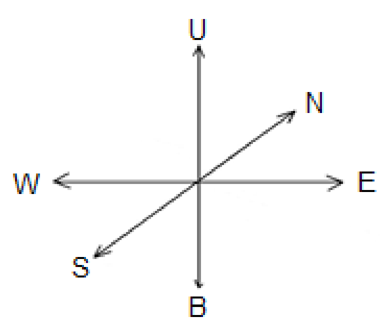
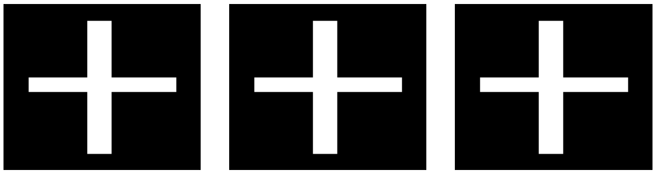


Figure 2. Scan addresses. Each letter represents the North, South, East, West, Up, and Bottom

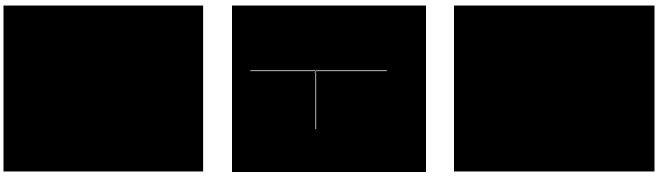
Although this algorithm works well in many cases, it produces significant errors in several cases (Fig. 3 and Fig.

978-1-4673-9461-1/15/$31.00 ©2015 IEEE

4). This paper describes error causes and presents the solution. It also includes an optimization in order to improve the algorithm.



(a)



(b)

Figure 3. (a) 3D image composed of three identical synthetic planes shown from top to bottom (b) Skeleton obtained by applying Lee's method.

Figure 3(a) shows a 3D synthetic image composed by three identical planes where Lee´s method works wrong.

2. Methodology

We did not find in the literature any mention regarding the errors produced by this technique. Therefore, it was necessary to profoundly study this method to determine under which conditions erroneous skeletons were produced.

To evaluate the technique, 132 synthetic 3D images recreating different conditions were created. In particular, images producing erroneous skeletons, as illustrated in Figures 3 and 4, were created. Results were evaluated by using the ITK, ImageJ and Matlab implementation of the Lee et al. method [7-9].

(a)

(b)

Figure 4. (a) 3D image composed of three synthetic images. (b) Skeleton obtained by applying Lee's method.

Figure 4 shows a result where Lee´s method given a result with several mistakes.

To save computational time, Lee divides the 26-neighbor volume into eight octants and represents the vertex of the 2x2x2 octants as a binary byte, where a bit is zero or one, depending on if it is white or black. Figure 5 (a) shows the weights given to each voxel. Thus, a table of 256 values is made to pre-evaluate all possible Euler characteristics. However, given that a voxel is considered for elimination, only when it is white, the analysis of this voxel can be avoided once it is found if it is white and only the other seven voxels inside (Fig. 5(b)) each octant are pre-evaluated to consider the Euler characteristics. Therefore, a table of only 128 values is now employed (Table 1).

Lee et al. Authors

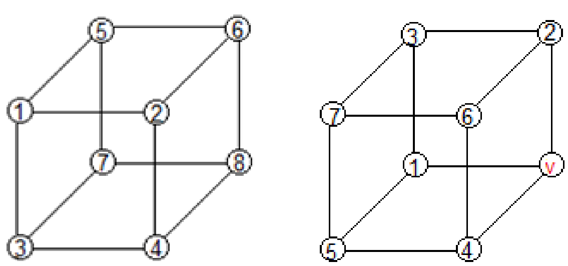


Figure 5. Voxel order. (a) Lee et al. and (b) Authors. Where v is the point to be eliminated. [2]

n	$8\delta G_{26}$	n	$8\delta G_{26}$	n	$8\delta G_{26}$	n	$8\delta G_{26}$	n	$8\delta G_{26}$
0	1	26	3	52	3	78	1	104	-1
1	-7	27	3	53	3	79	1	105	-1
2	-3	28	3	54	3	80	-1	106	-1
3	-3	29	3	55	3	81	-1	107	-1
4	-3	30	3	56	1	82	3	108	-1

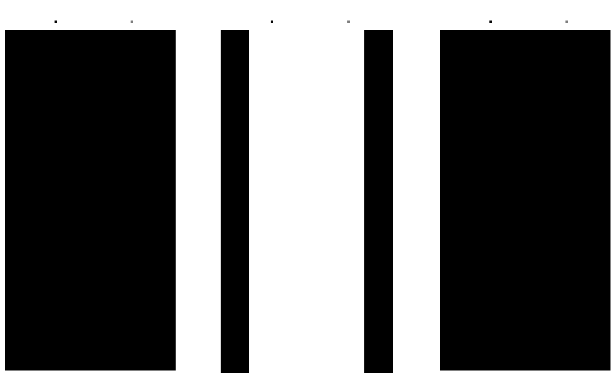
5	-3	31	3	57	1	83	3	109	-1
6	1	32	-1	58	1	84	-1	110	-1
7	1	33	-1	59	1	85	-1	111	-1
8	-1	34	-1	60	1	86	3	112	1
9	-1	35	-1	61	1	87	3	113	1
10	-1	36	3	62	1	88	1	114	1
11	-1	37	3	63	1	89	1	115	1
12	-1	38	3	64	-1	90	1	116	1
13	-1	39	3	65	-1	91	1	117	1
14	-1	40	1	66	3	92	1	118	1
15	-1	41	1	67	3	93	1	119	1
16	-3	42	1	68	-1	94	1	120	-1
17	-3	43	1	69	-1	95	1	121	-1
18	1	44	1	70	3	96	1	122	-1
19	1	45	1	71	3	97	1	123	-1
20	1	46	1	72	1	98	1	124	-1
21	1	47	1	73	1	99	1	125	-1
22	5	48	-1	74	1	100	1	126	-1
23	5	49	-1	75	1	101	1	127	-1
24	3	50	-1	76	1	102	1		
25	3	51	-1	77	1	103	1		

Table 1. Table of Euler characteristic preservation

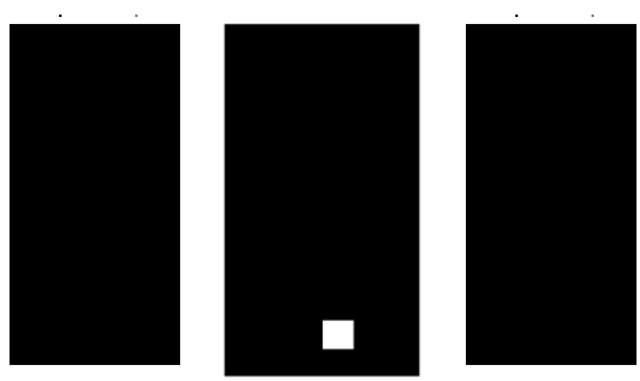
Where represents the modified octant and represents the Euler coefficient when a voxel is removed from this octant, for simplicity the entries in Table 1 have been multiplied by 8.

It was found that Lee's method fails when the initial width of the object in a given scan direction is even, as illustrated in Figures 3 and 4. The error is not easily seen because an object has different widths along its body.

To understand how the problem is produced, a 2D object in a 3D space is employed to describe how Lee's algorithm obtains the skeleton. This object is a 4x12x1 pixels rectangle oriented north-south in the X plane with black up and bottom layers, as shown in Figure 6(a).



(a)



(b)

Figure 6. Synthetic image created for analysis. (a) Original image and (b) result of the method.

Figure 6(b) shows Lee's result, where the skeleton consists of just two voxels, located at the bottom of the original rectangle. This is a wrong skeleton given that the one expected was a line one pixel in width going from top to bottom. The result is obtained in the following manner. First, the object is scanned, as illustrated in Figure 7, where numbers 1, 2, 3, or 4 describe the sequence followed to skeletonize the object. Initially, the border points of type North are evaluated, then the South, West, and East voxels, respectively, for each image. The whole set of candidate voxels are eliminated, given that they satisfied the three conditions previously established. Then, the 3D image is scanned again, but now from top to bottom. In this case, the second image of the stack is scanned, as shown in Figure 8. Now, the first white voxel (a) is deleted because it satisfies the three conditions. The following voxel (b) is also eliminated because it meets the three conditions. The remaining voxels will also be deleted in the same way. Only the last two voxels (c, d) are retained, just because the number of neighbors is lower than two (Condition 4 in Lee et al.).

In the ideal case, voxel b should not be retained, along with the voxels under b. However, because it is a single voxel, the number of neighbors is higher than one and, in particular, the Euler characteristic does not change, as shown in Figure 10, that is, before and after removing a candidate voxel, the Euler characteristic remains constant and, thus, it is removed. This situation corresponds to case 12 in Table 1 (equivalent to Case 49 in Lee's table). Therefore, the voxel is removed when it should be retained.

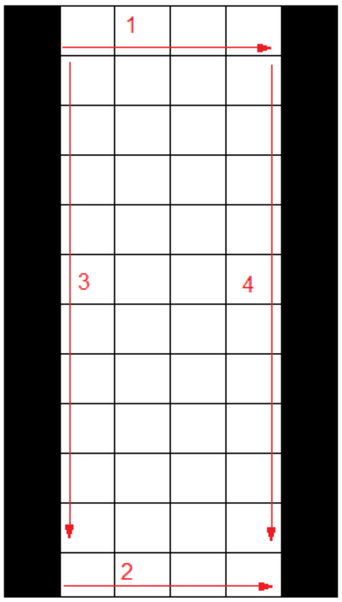


Figure 7. Scanning the edges of the image

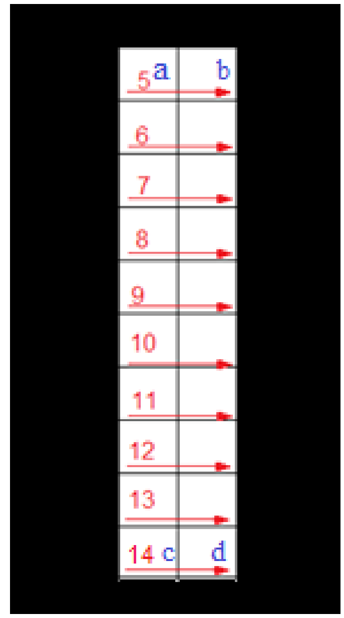


Figure 8. Second image scanning

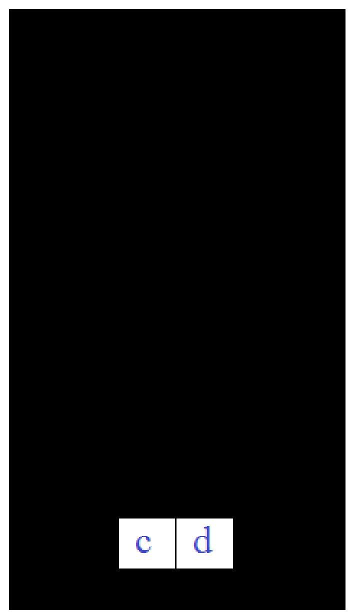


Figure 9. Final result of the method

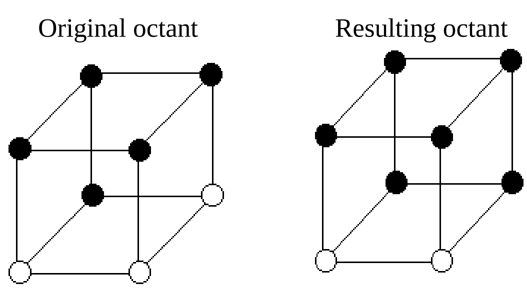


Figure 10. Case 12, which produces an error in the skeleton when presented

This problem is not easily observed, given that most objects have variations in width and the thickness is odd; thereby, the problem disappears due to the condition of elderly neighbors.

This problem is due to the way the scan is done. In the example, the problem appeared when the image was scanned row by row. To solve this problem, scanning is done plane by plane, following the six directions suggested by Lee et al., but inside each plane scanning is done from the outside in opposite directions, where each edge is given by the intersection of the plane and the planes perpendicular to such. Thus, the scanning can be done by following the sequence: NE,NW,NU,NB,SE,SW,SU,SB,EN,ES,EU,EB,WN,WS, WU,WB,UN,US,UE,UW,BN,BS,BE,BW.

3. Results

Table 2 table shows the results obtained with four synthetic images, using the Lee et al. method and the modified one. The whole set of images is composed of three identical layers.

Original image	Lee et al. result	Modified method result

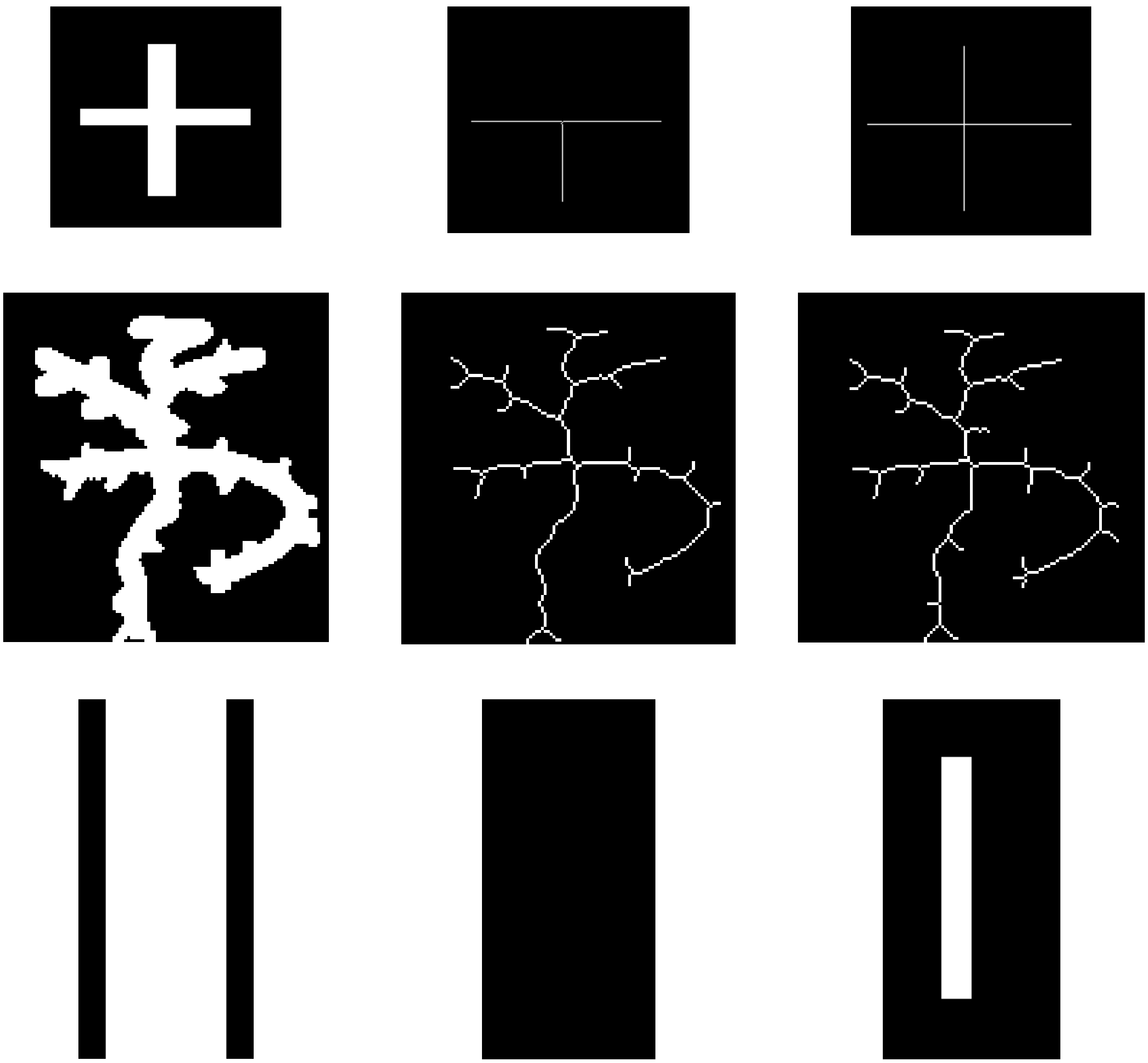


Table 2. Results

As it can be seen, the modified method produces more accurate skeletons, solving the problem found in the original method, while keeping the same skeletons when the original method was right.

Conclusions

The causes of errors produced by irregularities in the skeletons obtained with the method by Lee et al. were explained. It was found that although the basic conditions of the method are right, error occurred because of the way images are scanned. A solution was presented, which allows solving the problem. A new table of Euler characteristics was also given, reducing the size of the original table from 256 to 128 values, allowing to reduce the memory used.

References

[1] Palágyi K., “Topology Preserving Parallel 3D”, Institute of Informatics, University of Szeged, Szeged, Hungary, 2012.

[2] Palágyi K. and Németh G., “Fully Parallel 3D Thinning Algorithms Based on Sufficient Conditions for Topology Preservation”, University of Szeged, Szeged, Hungary, 2009.

[3] Palágyi K., “A 3D fully parallel surface-thinning algorithm”, Theoretical Computer Science Vol 406, 119–135, 2008.

[4] Palágyi K., “A 3-subiteration 3D thinning algorithm for extracting medial surfaces”, Pattern Recognition Letters Vol 23, 663–675, 2002.

[5] Palágyi K., “Topology Preserving Parallel 3D”, Institute of Informatics, University of Szeged, Szeged, Hungary, 2012.

[6] Lee T. C., R. L. Kashyap, “Building skeletons models via 3D medial surface/axis thinning algorithms”, Knowledge based laboratory, Vol 56, pp. 462-478, 1994.

[7] Implementation of the method of Lee et al. to ITK, by Hanno Homann, http://www.insight-journal.org/browse/publication/181.

[8] Implementation of the method of Lee et al. to ImageJ, by Ignacio Arganda, http://fiji.sc/Skeletonize3D#General_Description.

[9] Implementation of the method of Lee et al. to Matlab, by Philip Kollmannsberger, http://www.mathworks.com/matlabcentral/fileexchange/43400-skeleton3d.

[10] Skeletonize 3D [Online]. Available in: (http://imagejdocu.tudor.lu/doku.php?id=plugin:morphology:skeletonize3d:start), Nov 2011, Oct 2014.

[11] D. G. Morgenthaler, “Three Dimensional Simple Points: Serial Erosion, Parallel Thinning, and Skeletonization”, Technical Report TR-1005, Computer Vision Laboratory, Computer Science Center, University of Maryland, 1981.

[13] S. Lobregt, W. Verbeek, and F. C. A. Groen, “Three dimensional skeletonization: Principle and algorithm”, IEEE Trans, Pattern Anal. March Intell. Vol 2, 75-77, 1989.

[14] Gisela kettle, Computer Analysis of Images and Patterns Lecture Notes in Computer Science, simple point in 2D and 3D binary images. Vol 2756, 57-64, 2003.

[15] D. G. Morgenthaler, “Three Dimensional Digital Topology: The genus, Technical Report” TR-980, Computer Vision Laboratory, Computer Science Center, University of Maryland, 1980.

[16] T. Y. Kong and A. Rosenfeld, Digital topology: Introduction and survey, Comput. Vision Graphics Image Process.Vol 48(3). 357-393, 1989.