A Survey of 3-D Thinning Algorithms

Paul C. K. Kwok and Vishwa Ranjan
Department of Computer Science
The University of Calgary
Calgary, Alberta, Canada T2N 1N4

Abstract

A number of 3-D thinning algorithms are examined. There are three approaches reported for the thinning of 3-D binary patterns based on the notions of path-connectivity, connectivity number, and topology. Their merits and demerits are described and suggestions for improvement have been included wherever possible. The algorithms published so far seem to be inadequate in one respect or the other. One common problem is that either a necessary or a sufficient (but not both) condition has been used to determine whether a voxel can be removed or not. A necessary but not sufficient condition leads to incorrect results while a sufficient but not necessary condition means that more voxels could have been removed in any iteration.

Keywords: 3-D thinning, skeleton, connectivity, erosion, interior voxel.

1 Introduction

Skeletonization (or thinning) of 3-D binary patterns can be performed iteratively by removing, in each iteration, voxels on the surface of an object until the object is reduced to a set of lines or curves of unit width[1]. In other words, skeleton should consist of just enough voxels necessary to maintain connectedness of the object. When considering the removal of a voxel from the pattern, the following should be observed:

[i] Topology is preserved i.e. the number of components, holes, and cavities in the object, and in the complement must be the same before and after the iteration.

[ii] Excessive erosion is prevented i.e. end-points of branches are detected as soon as possible, and retained, so that the length of protrusion in any direction representing a true feature of the object is not shortened excessively.

3-D pictures are not uncommon these days. They can be obtained by advanced techniques like computed tomography (CT), echocardiography, and magnetic resonance imaging (MRI). 2-D time varying images may also be treated as 3-D pictures where third dimension is time. Thinning can be used as a pre-processing step in object characterization and recognition. Some thinning algorithms are reversible and can be considered as a topological data-compression technique to be followed by other standard data-compression method to achieve very high compression ratios [15]. It will also be useful in efficient manipulation of solidly-modeled objects in computer animation. A thinning algorithm will be useful in space management and planning in robotics. It may have applications in volumetric modeling for mechanical engineering design, quantitative microscopy, analysis of grain microstructures in rocks and metals, and in crystallography [8].

The applications of thinning are limited by the fact that it is intended for solid binary objects. Unless boolean operators on objects are of significance, 3-D objects are more conveniently represented as a set of polygons. Skeletons are not unique. They are not well-defined for nearly spherical or cubical objects (objects with almost same length, breadth, and depth) but provide a good representation for elongated solids.

The paper assumes familiarity with the concepts like neighborhood, path-connectivity, and surface-connectivity in 3-D digital pictures. Detailed descriptions of these can be found in [4, 6, 9]. Some of these concepts are discussed briefly in Section 3. In this paper, 26-connectedness is assumed for the object and 6-connectedness for the complement.

2 Desirable Properties

Skeleton is a representation of the shape of the original pattern. From the number of iterations it took to skeletonize we should get an idea of length to thickness ratio of the object. Although, a thinning algorithm can be application specific, the following properties are desirable in a thinning algorithm:

[a] The algorithm should be isotropic i.e. [i] no matter what the configuration of the object in the 3-D digital space be, the final skeleton should almost be the same, transformed and rotated according to the initial configuration of the object, and [ii] if an iteration is divided into a number of sub-iterations, the order of sub-iterations should have a little influence on the shape of the skeleton.

---

[1] Some algorithms define skeleton as the set of medial surfaces.

[b] An iterative algorithm will usually reduce an object into a thin surface. If the skeleton is going to be a set of lines, further erosion should take place along the edge of the thin surface. An object may be a thin surface to begin with or it may be reduced to a surface after a number of iterations. This will ensure uniform erosion of objects and will be useful in applications like computer animation where an object shrinks step by step.

[c] The property of reconstructibility will be very useful in many applications. In this case the algorithm is reversible: we should be able to get back the original object from the skeletal voxels. One obvious advantage will be the reduction in memory space needed to store the essential structural information present in the pattern. It should be noted that reconstruction is not trivial for 3-D images, simply because [i] for a thinning algorithm, in general, many configurations can lead to same skeleton, and [ii] a voxel may be removed under many local configurations, hence, to get back this configuration would require encoding of a lot of information. Some reconstruction issues are discussed in [15].

[d] The algorithm should be robust and without any loopholes. There should be no scope for reasons like "such patterns are quite rare" because misinformation due to a loophole may be enormous, and consequences unacceptable depending upon the application.

[e] The skeleton should be immune to the noise in the digitization process.

3 Terminology

3.1 Neighborhood

A 3-D binary pattern can be described by a 3-D array of voxels (i.e. volume elements) $P = \{p(i,j,k) \mid 1 \leq i \leq L, 1 \leq j \leq M, 1 \leq k \leq N\}$ where L, M and N are positive integers and each point $p(i,j,k)$ has a value either 0 or 1. The objects, which form the foreground S of the pattern are represented by a set of "dark points" while the complement $C = P - S$ corresponds to the set of "white points". The $3 \times 3 \times 3$ neighborhood (referred to as l-window in this paper) of a point v are shown in Figure 1a. For convenience, six directions[2], N, S, E, W, T, and B are associated with the coordinate axes as shown in Figure 1b. The 6-neighbors of a voxel $v = p(i,j,k)$ are those points $\{p(l,m,n) \mid |l-i|+|m-j|+|n-k| = 1\}$ and the 26-neighbors are points $\{p(l,m,n) \mid \max(|l-i|, |m-j|, |n-k|) = 1\}$. A dark voxel which has at least one white 6-neighbor is called a boundary (or border or edge) point.

If deletion of a border point does not affect the topology in the l-window, it is called a simple (l-removable) point. The points that belong to the skeleton of an object are called the final points.

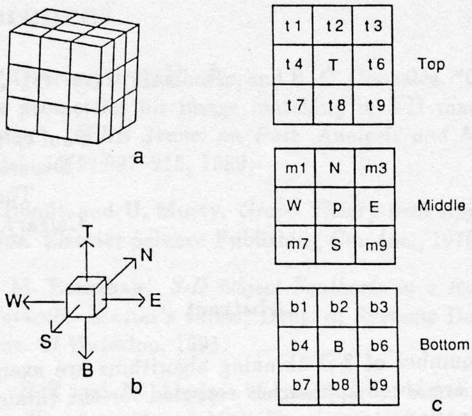

Figure 1: (a) $3 \times 3 \times 3$ neighborhood of a voxel or the l-window. (b) Six directions associated with a voxel. (c) Three-layer representation of the l-window.

3.2 Approaches

In many cases, thinning is an iterative process. Rosenfeld et al. [1] classified thinning algorithms as being sequential or parallel. A thinning algorithm is parallel if all the boundary points (of the objects) satisfying removal conditions are deleted simultaneously, and is sequential if the boundary points are processed and removed one after the other. With parallel algorithms, only result from the previous iteration affects the decision to remove a point in the current iteration, making it suitable for processing by parallel hardware such as a cellular array. A sequential algorithm, on the other hand, uses the results from previous passes plus the results obtained so far in the current pass to process the current voxel. In many cases, it is necessary to divide each iteration into a number of sub-iterations to process border voxels in specific directions, in order to preserve connectedness.

3.3 Topology

Rosenfeld and Morgenthaler have extended the concepts of 2-D digital geometry to three dimensions. They published a series of research reports on 3-D digital topology, curves and surfaces in 3-D digital pictures, and on thinning [4, 5, 6, 7]. Topological invariance implies that the number of components, holes, and cavities in the skeleton are the same as in the original object and that the basic shape is preserved. A localized version of this principle operates on the l-window of a voxel. The voxel is removable if its removal does not change the number of components and the number of holes in the l-window.

3.4 Path-connectivity

This is an extension of the concept of connectivity in 2-D digital pictures. If a voxel v and any of its 6-neighbors (or 18- or 26-neighbors) are dark, these two are said to be 6-connected (or 18- or 26-connected). If v_0 and v_m

---

[2] Acronyms for north, south, east, west, top and bottom respectively.

are two dark points that belong to the same object, there exists a path, which can be described as a chain of dark points $v_0, v_1, \ldots v_m$, with each pair of voxels, v_i and v_{i+1} ($i = 0, 1, \ldots m-1$), being neighbors of each other. If 6-neighbors (or 18- or 26-neighbors) are considered, v_0 and v_m are said to be 6-connected (or 18- or 26-connected).

3.5 Connectivity Number

The concept of connectivity number rests on the notion of Euler number for 3-D digital pictures. Euler's formula says that there is a fixed relation among the numbers of vertices, edges and faces of a simple polyhedron: $V - E + F = 2$, where, V, E, and F denote the number of vertices, edges, and faces of the polyhedron respectively. More generally, when an object contains tunnels, Euler's formula has been extended to $V - E + F = 2 - 2P$, where, P denotes the number of tunnels and $2 - 2P$ is called the Euler characteristic of the surface. The connectivity number N of the objects in a 3-D digital picture is defined as the sum of Euler characteristics, $N = \sum_i (2 - 2P_i)$, over all polyhedrons.

3.6 Surface-connectivity

The connectivity number approach is linked to the topological approach by the concept of genus. The genus G of a 3-D pattern is defined as $O - H + C$, where O is the number of components, H is the number of holes, and C is the number of cavities. Essentially, genus and connectivity number represent the same criterion [8]. It can be shown that for simple polyhedra, connectivity number is twice the genus. These two approaches together are referred to as surface-connectivity approach. It should be noted that surface-connectivity is related to path-connectivity. It has been shown in [9] that preservation of surface-connectivity implies preservation of path-connectivity but the converse is not always true. In fact, the two approaches can yield same results if some constraints are applied.

4 Critical Problems

4.1 Simple Points

The main problem in thinning is the definition of a simple point. Different algorithms differ in their removal criterion (i.e. definition of simple points), definition of end-point, and the type of window they use in determining a simple point. Very few algorithms have succeeded in this objective. Some algorithms treated a necessary condition to be a sufficient condition [2, 13]. Some algorithms do not work because of the parallel nature of the algorithm [2, 9]. In some algorithms, the quality of skeleton was degraded by the fact that sufficiency of a condition does not imply necessity and hence some voxels which should have been removed were not removed, e.g. stage 1 of [9], [13], [14].

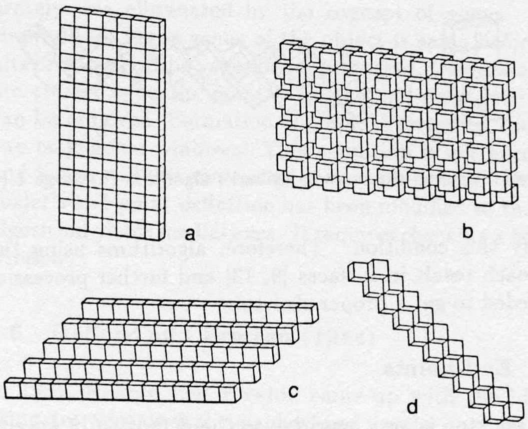

Figure 2: Examples of thin surfaces.

4.2 Thin Surfaces

Thin surfaces pose another difficulty in 3-D digital geometry. A single surface can have voxels with different type of six neighborhood. A few examples of surfaces are shown in Figure 2. The problem that parallel algorithms face in processing thin surfaces is that a whole surface can be reduced to a point in one iteration (for example, when the surface in Figure 2c is being processed). A sequential algorithm, when processing a surface may give one of the borders of the surface as the skeleton instead of the medial line, and that too in a single iteration[3]. Some algorithms stop as soon as the object has reduced to a surface [7, 13, 14]. Some algorithms have two stages. In the first stage, an object is thinned to a surface which is further thinned to its medial axis in the second stage [2, 9].

4.3 Holes

Another problem in the extension of 2-D digital geometry to 3-D is the concept of a hole (tunnel). The equivalent of a hole in 2-D is a cavity in 3-D. A 3-D hole may or may not change the number of components in the complement. The problem of detecting formation of a hole is essentially that of detecting a ring around the central voxel in the l-window. Alternatively, a 6-path between two white 6-neighbors (which lie on different sides of a surface) will not exist if removal of central voxel results in a hole. Both these problems are quite complex.

Two approaches have been adopted to prevent the making of a hole in the object.

1. The definition of genus can be used conversely to calculate the number of holes [7].
2. Checking for connectivity on two orthogonal planes prevents formation of a hole [10]. The proof of this fact has been included in the appendix.

The second approach is simple to implement but is not a necessary condition. Not all voxels that are removable

---

[3] The final skeleton depends on how the end-points are defined but the inherent problems should be obvious from the example.

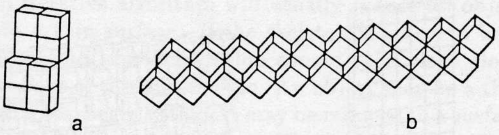

Figure 3: Counter-examples to Srihari's algorithm (a) Stage 1 (b) Stage 2.

satisfy this condition. Therefore, algorithms using this approach result in surfaces [9, 13] and further processing is needed to get a proper skeleton.

4.4 End-points

The skeleton is very sensitive to the definition of an end-point. Many algorithms define end-point as a dark voxel with at most one dark neighbor [2, 9, 13]. Obviously, parallel algorithms will fail to detect an end-point for two layered objects whereas a sequential algorithm may or may not yield an end-point depending on the order of sub-iterations. Morgenthaler [7] generalized the concept of end-points to take into account the difficulties arisen from the parallel nature of algorithms. All the points on an arc, curve, or surface are defined to be "ends". Tsao and Fu [10] improved on Morgenthaler's "ends" definition and used 5 × 5 × 5 window to define end-points to give medial axes as the skeleton.

5 Different Algorithms

In this section, we discuss the algorithms on 3-D thinning in the order they were published. The emphasis is on approach rather than on actual removal criterion. The papers should be referred for exact details on definition of simple and end-points. All the algorithms are parallel in nature.

5.1 Srihari et al. (1979)

Srihari et al. (1979) presented a 3-D thinning algorithm based on the concepts of path connectivity. This is the first reported attempt for thinning 3-D digital patterns. The algorithm is divided into two stages. In the first stage, the object is thinned into its medial surface, which is thinned to medial axes in the second stage. Removal condition just ensures that the number of components in the l-window will not change. This condition is far from being sufficient to preserve topology because of the following reasons:

- Holes are not checked for. The number of holes may increase.
- It is possible to construct patterns where this algorithm will fail. The objects can be disconnected or even removed completely because of the parallel nature of the algorithm.

The object in Figure 3a will be disconnected by stage 1 of Srihari's algorithm whereas whole object in Figure 3b will be deleted by stage 2 of the algorithm [9].

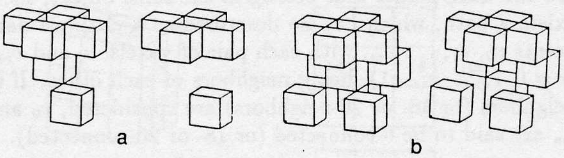

Figure 4: Examples to show that connectivity number is not a sufficient criterion to maintain connectedness.

5.2 Verbeek et al. (1980)

Verbeek et al. [3] described a local removal condition for 3-D thinning (1980) based on the idea of connectivity number N. A voxel v belonging to an object can be deleted when this deletion does not cause a change in N. This implies determining N before and after deletion of v. It is shown that the l-window of v can be considered as eight partially overlapping $2 \times 2 \times 2$ cubes (centered around each nodal point j of v) for which the contributions N_j to N are computed separately.

It is found that the removal criterion is a necessary but not sufficient condition to maintain connectedness. Figure 4 serves as an example in support of this fact. Because $V - E + F = 2 - 2P$, if increase in the number of components is same as the number of tunnels (holes) created, this number will be invariant! Hence, if we remove the central voxel in Figure 4, connectivity number remains same although connectivity is broken and hole(s) created. In Figure 4a removal of central voxel creates one hole and one new component in the l-window whereas in Figure 4b two holes and two new components are created. The criterion can be made sufficient by applying a constraint that the number of components in the l-window after removal should be one (number of components before removal is one in any case). It should be noted that although this condition is necessary and sufficient to preserve topology locally, it still cannot distinguish between the cases of formation of a hole, and breaking of connectivity.

5.3 Tsao and Fu (1981)

Tsao and Fu [9] modified the algorithm presented in [2] and tried to make it complete by imposing some extra constraints as the conditions in [2] were not sufficient to prevent the making of a hole. The imposed condition is sufficient but not necessary and hence results in a surface after stage 1. They proved that their algorithm will preserve 26-connectedness of the object. They based their claim on the correctness of their criterion in the l-window. In their algorithm, the difficulty is not with the removal criterion but with the parallel nature of the algorithm. Figure 5 serves as a counter-example to their algorithm. All the voxels in Figure 5e will be removed if stage 1 of their algorithm is executed in parallel. However, their algorithm should work if points are examined sequentially or if the end-point definition is modified. Also, the algorithm will give undesired results for thin surfaces as it fails to

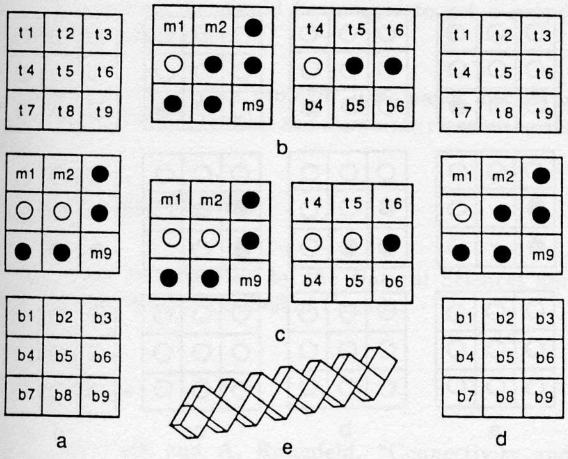

Figure 5: Counter-example to Tsao and Fu's algorithm; lettered boxes denote white voxel. (a) l-window of p. (b) Planes $z = k$ and $y = j$ before removal of p. (c) Planes $z = k$ and $y = j$ after removal of p. (d) l-window after removal of p. (e) The plane being processed.

distinguish between interior voxels and border voxels in a surface. For some surfaces thinning can be six times faster than for a parallelopiped.

5.4 Morgenthaler (1981)

Morgenthaler (1981) gave the most general condition for determining a simple point [7]. The algorithm is motivated by all three previous attempts. A point is simple in an l-window if its removal does not change the number of components and holes in the object and the complement. What this means is that four numbers have to be computed. There is no simple way of computing the number of holes. The number of components, number of cavities and the number of holes are linked by the concept of genus. It is used conversely to find the number of holes. A detailed description can be found in [7].

There is no suggestion as to how these numbers can be calculated easily. It is shown that l-window is not sufficient for isotropic parallel thinning operations. Some of the conditions in [7] can easily be improved. For example, number of holes in the object before removal (in l-window) will be nil. So, we have to just check that there is no hole after removal. Similarly, the number of components before removal is one. So, we have to check that the number of components after removal will be one. The algorithm takes care of the difficulties associated with the parallel nature of the algorithm by generalizing the definition of an end-point. All points of an arc, curve, and surface are defined to be "ends". Hence, thin surfaces are not processed at all.

5.5 Tsao and Fu (1982)

Tsao and Fu (1982) improved on Morgenthaler's simple and end-point definitions to get a proper thinning algorithm [10]. The problem of computing four numbers separately was eliminated by the concept of genus. They showed that if the genus of the object is same before and after removal of the voxel being processed, then if no holes are created, and the point is not an end-point, the voxel can be removed. Formation of a hole is tested by checking two orthogonal windows. This condition is not necessary and hence will not remove all removable points. Morgenthaler's end-point definition has been modified so that the algorithm yields medial axes. It requires checking a $5 \times 5 \times 5$ window.

5.6 Hafford and Preston (1984)

In 1984, Hafford and Preston came up with the idea of using tetradecahedral neighborhood in the face-centered cubic tessellation. This is an extension of Golay's 2-D subfield concept using hexagonal tesellation. It allows use of a localized parallel operator for determining simple points in three dimensions using six subfields. It makes use of a table of action corresponding to each of the 2^{12} cases. It suffers from the formation of undesired rings in the skeleton. It is not clear from the published work how the algorithm will preserve the original topology of the object as a whole. The problems due to the parallel nature of the algorithm and excessive erosion of thin surfaces in an iteration are not taken care of.

5.7 Mukherjee et al. (1989)

Mukherjee et al. [13] (1989) gave an extended safe point thinning algorithm (ESPTA) based on 2-D SPTA by Naccache and Shinghal [12]. The authors claim that the 18- (6-) connectivity in 3-D can be ensured through the enforcement of 8- (4-) connectivity in two 2-D orthogonal planes. It is shown that ESPTA preserves 18-connectedness of objects. However, in order for the object and the complement to be well behaved mathematically, 6- and 26-connectedness provide a neat complementary relationship in the sense that if 6-connectedness is being considered for the object, 26-connectedness should be considered for the complement and vice versa. For the case of 18-connectedness there is no such relationship at all between the object and the complement.

The basic idea is to apply the boolean equations for 2-D SPTA to two of the three 3×3 orthogonal planes. These two conditions are then combined to get the boolean equation representing the deletion criterion. The algorithm checks for 2-D connectivity on two orthogonal planes. No check is performed on the third orthogonal plane. This is therefore not a sufficient condition in the l-window to maintain connectedness, and the number of components. Figure 6 shows an example where this algorithm will fail. The algorithm will preserve 18-connectedness if some constraints are applied on the third orthogonal plane. The patterns are thinned to a surface biased towards a certain direction because in SPTA voxels once marked as safe cannot be removed in any successive iteration. Moreover, their claim of 100% reconstructibility is not correct because SPTA is not exactly reconstructible.

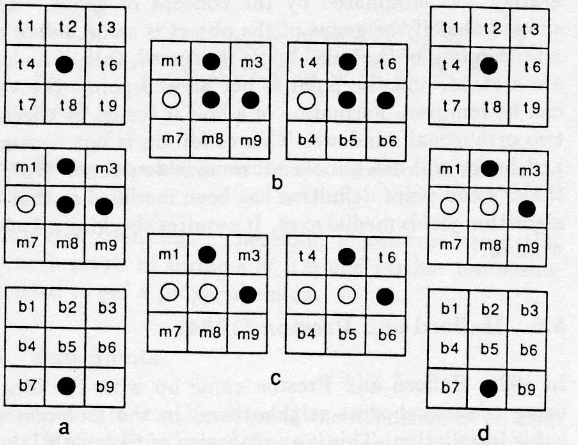

Figure 6: Counter-example to ESPTA; lettered boxes denote white voxel. (a) l-window of p. (b) Two orthogonal planes on which p is same type of border point. (c) Same planes as (b) after removal of p. (d) Formation of two 18-connected components in the l-window after removal of p.

5.8 Gong and Bertrand (1990)

Gong and Bertrand (1990) improved on Morgenthaler's characterization of simple points. They also defined new topological predicates which characterize simple points and are simple to calculate. They showed that their predicates include the predicates in [7]. The conditions for a point to be simple are sufficient but not necessary. Hence the algorithm results in a surface which is very difficult to foresee. In fact, it removes less points compared to [7] but has the advantage of simplicity. The difficulties associated with the parallel nature and thin surfaces are still there.

6 Conclusion

In 2-D, a parallel algorithm without sub-iterations fails for two-pixel wide lines. However, a local removal condition together with four sub-iterations is a sufficient condition to guarantee connectedness globally. A sequential algorithm without sub-iterations will preserve connectedness but a two-pixel wide line may reduce to a point[4]. A sequential algorithm with sub-iterations will again work but may yield spurious end points.

In 3-D, a parallel algorithm without sub-iterations will fail for two-voxel thick lines and planes. Unlike in 2-D, a local removal condition along with six sub-iterations is not a sufficient condition to preserve global connectedness in 3-D. Consider the simple configuration in the window shown in Figure 4. Both x and y may be processed in the E sub-iteration, and are l-removable individually (because their removal will not result in disconnectivity, nor will it form a hole in the respective l-windows), but obviously, the object is disconnected if both the voxels are removed. So far, it was assumed that the preservation of topology in the l-window preserves the topology globally. A sequential algorithm without sub-iterations will preserve connectedness but as in 2-D case will be inadequate in processing two-voxel thick configurations. A sequential algorithm with sub-iterations will also preserve connectedness but may result in multilayer erosion from a unit voxel thick surface. It may lead to a skeleton with many branches and spurious end-points.

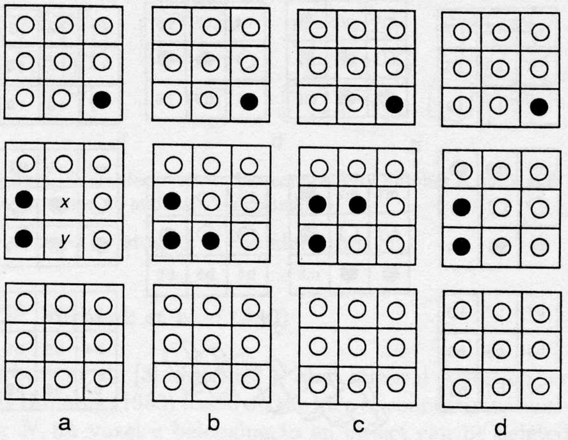

Figure 7: Example configuration which shows that preservation of connectivity in l-window does not imply preservation of connectivity globally. (a) Initial configuration: both x and y are dark. E is towards the right edge of the paper. (b) Configuration if only x is removed. (c) Configuration if only y is removed. (d) Configuration if x and y both are removed.

Considering the general difficulties associated with the approaches so far, we suggest the following:

1. Directionality should be incorporated in the processing of an individual voxel, i.e. if it is a multiple type of border voxel it should be processed in exactly one direction. This will prevent algorithms from processing multiple-type border points in the wrong sub-iteration. For example top layered voxels in Figure 2c should not be processed in the south sub-iteration although they are south-type border points also. This will take care of multilayer erosion in thin surfaces. Some algorithms have taken care of this partly [2, 9, 14]. Incorporating directionality also helps in simplifying the end-point condition.
2. A distinction should be made between border and interior voxels on a thin surface. Voxels interior to a surface should be detected and they should not be processed in any subsequent sub-iteration.
3. Thinning of 3-D digital patterns requires a tremendous amount of processing. Therefore, a parallel algorithm is a suitable choice. On the other hand, definition of simple points and end-points is not easy in parallel algorithms. So, a sequential algorithm is the alternative. Neither parallel nor sequential approach alone is sufficient for the kind of algorithm desired—

---

[4] Sequential algorithms, in general, preserve connectedness but suffer from the inability to process two pixel (voxel in 3-D) thick configurations—a problem commonly referred to as *ducting*.

both approaches should be merged to get a proper thinning algorithm.

3-D thinning is a difficult problem. This paper intends to clarify some of its difficulties and stimulate research in the field.

Acknowledgements

This work is supported by the Natural Sciences and Engineering Research Council of Canada.

References

[1] C. M. Park and A. Rosenfeld, "Connectivity and genus in three dimensions", TR-156, Computer Science Center, University of Maryland, College Park, MD, May 1971.

[2] S. N. Srihari, J. K. Udupa, and M. M. Yau, "Understanding the bin of parts", *Proceedings of the IEEE Conference on Decision and Control*, pp. 44-49, 1979.

[3] S. Lobregt, P. W. Verbeek, and F. C. A. Groen, "Three-dimensional skeletonization: principle and algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, no.1, pp. 75-77, Jan. 1980.

[4] A. Rosenfeld, "Three-dimensional digital topology", TR-936, Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD, Sept. 1980.

[5] D. G. Morgenthaler and A. Rosenfeld, "Surfaces in three-dimensional digital images", TR-940, Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD, Sept. 1980.

[6] A. Rosenfeld and D. G. Morgenthaler, "Some properties of digital curves and surfaces", TR-942, Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD, Sept. 1980.

[7] D. G. Morgenthaler, "Three-dimensional simple points: serial erosion, parallel thinning, and skeletonization", TR-1005, Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD, Feb. 1981.

[8] S. N. Srihari, "Representation of three-dimensional digital images", *ACM Computing Surveys*, vol. 13, pp. 399-424, Dec. 1981.

[9] Y. F. Tsao and K. S. Fu, "A parallel thinning algorithm for 3-D pictures", *Computer Vision, Graphics and Image Processing*, vol. 17, pp. 317-330, 1981.

[10] Y. F. Tsao and K. S. Fu, "A 3-D parallel skeletonwise algorithm", *Proc. IEEE Conference on Pattern Recognition and Image Processing*, pp. 678-683, 1982.

[11] K. J. Hafford and K. Preston Jr., "Three-dimensional skeletonization of elongated solids", *Computer Vision, Graphics and Image Processing*, vol. 27, pp. 78-91, 1984.

[12] N. J. Naccache and R. Shinghal, "SPTA: a proposed algorithm for thinning binary patterns", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 14, pp. 409-418, May 1984.

[13] J. Mukherjee, B. N. Chatterjee, and P. P. Das, "Thinning of 3-D images using the Safe Point Thinning Algorithm (SPTA)", *Pattern Recognition Letters*, vol. 10, pp. 167-173, Sept. 1989.

[14] W. Gong and G. Bertrand, "A simple parallel 3D thinning algorithm", *Proceedings of the 10th International Conference on Pattern Recognition*, pp. 188-190, 1990.

[15] P. C. K. Kwok and V. Ranjan, "3-D thinning and reconstruction", (submitted to) *6th International Conference on Image Analysis and Processing 1991*.

Appendix

Consider a voxel $v(x=i, y=j, z=k)$; x is EW axis, y is NS axis and z is TB axis. Consider 26-connectedness for the object and 6-connectedness for the complement. Then given that:

- one of the six neighbors of v is white i.e. v is a border voxel, and
- removal of v preserves 8-connectedness on two other orthogonal planes i.e. if v is E or W border point, 8-connectedness is preserved on planes $y=j$ and $z=k$,

we can prove the following proposition.

Proposition 1 *Removal of v cannot make a hole in the l-window.*

Proof. Consider the window shown in Figure 7a. A ○ means a white voxel. Let us say that $m4$ is white to begin with (i.e. v is at least E type border point). A hole is a 6-path in the complement surrounded by a ring which may have its axis parallel to any one of the three axes x, y, or z. We have three cases corresponding to these:

1. $m4-p-t5$ equivalent to $m4-p-b5$
2. $m4-p-m6$
3. $m4-p-m2$ equivalent to $m4-p-m8$

We do not have to consider any other combination as $t5-p-b5$ or $m6-p-m2$ because these are equivalent to cases 2 and 3 respectively.

C1: Consider $m4-p-t5$. Two orthogonal planes on which 8-connectivity is preserved are shown in Figure 7b. The ring axis is parallel to z-axis. Since, $m4$ is white, ring can only be completed if $t4$ is dark but if $t4$ is dark removal of p will disconnect $t4$ from $t6$ or $m6$ or both (one of these should be dark to complete

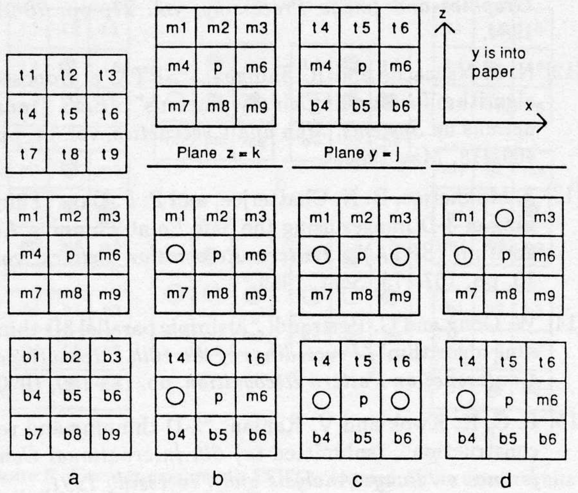

Figure 8: Figure for proof of Proposition 1. (a) Top, middle, and bottom layers in l-window of p. (b), (c), and (d): Configuration of planes $z = k$ and $y = j$ for cases 1, 2, and 3 respectively.

the ring).

C2: Consider $m4 - p - m6$. Two orthogonal planes on which 8-connectivity is preserved are shown in Figure 7c. The ring axis is parallel to x-axis in this case. Now because one of $t4$, $t5$, $t6$ and one of $b4$, $b5$, $b6$ should be dark to complete the ring, connectivity cannot be preserved between these two dark voxels in the second window.

C3: Consider $m4 - p - m2$. Two orthogonal planes on which 8-connectivity is maintained are shown in Figure 7d. The ring axis is parallel to y-axis this time which means $m1$ should be dark. But removal of p will disconnect it with $m3$ or $m6$ or both (one of these should be dark for completion of the ring).

Hence, under given conditions, removal of v cannot make a hole in the l-window.

Q.E.D.