Order Number 9510688

Connectivity preserving transformation of digital images: Theory and applications

Ma, Cherng-Min, Ph.D.

City University of New York, 1994

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

Connectivity preserving transformation of digital images: theory and applications

by

Cherng-Min Ma

A dissertation submitted for the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy

The City University of New York

1994

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfication of the dissertation requirement for the degree of Doctor of Philosophy.

7/28/94 — Tat Yung Kong

Date — Chair of Examining Committee Prof. Tat Yung Kong

7/28/94

Date — Executive Officer Prof. Stanley Habib

Prof. Ted Brown (QC)

Prof. Robert A. Melter (LIU)

Supervisory Committee Prof. Paul Meyer (Lehman)

The City University of New York

This thesis is dedicated to my wife

Hsu-Ling

Acknowledgments

I wish to thank many people whose comments and suggestions are extremely important to this research:

Prof. T. Yung Kong for his careful supervision in the entire research period. He provided not only research ideas in the beginning of my Ph.D. study but also guidance and support when the research was in progress.

Prof. Ted Brown, Prof. Longin Latecki, Prof. Robert Melter, and Prof. Paul Meyer for continuous support and invaluable comments to this research. According to their suggestions, this thesis is revised to be more precise and more readable.

Prof. Milan Sonka and Mr. Gopal Sundaramoorthy for stimulating discussions about thinning algorithms and for providing real 3D medical images that show possible applications of this research.

Prof. Azriel Rosenfeld for suggesting revisions of the material in Chapter 3. Prof. Richard Hall and Prof. Christian Ronse for their ideas in 2D and 3D connectivity preservations. I learned a lot by reading their papers.

Prof. Stan Habib for patiently auditing my examination. Prof. Ali Ghozati for valuable discussions about minimization techniques of Boolean expressions. Mr. Peter Chen and Mr. Prashant Joshi for supporting productive systems at Queens College.

Finally, I want to express my special thanks to my parents and my wife for their support and encouragement.

C. M. M.

New York City

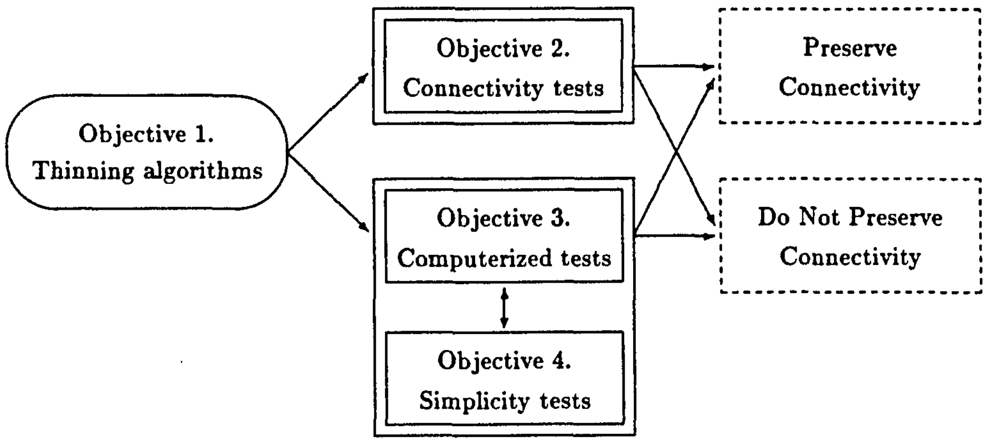
July, 1994

Preface

Pattern recognition is a major topic of image processing. Thinning, as a preprocessing operation of pattern recognition, is a process that erodes an object layer by layer until only a unit width skeleton is left. Since the skeleton is of unit width, it is easier to trace and hence is easier to recognize. A thinning algorithm must "preserve connectivity." This implies, for example, that it must never split or completely delete a connected object in an image, and must never create or eliminate a hole in an object. Normally, thinning algorithms only delete *simple* object points, i.e., points whose deletion does not change the connectivity structure of the original image. Many thinning algorithms for 2D images have been proposed. But this is not the case for 3D images.

Our first objective is to design two 3D fully parallel thinning algorithms – one for generating skeletons as "medial faces" and the other one for "medial lines." A multimedia team at M.I.T. is interested in the medial-face algorithm for analyzing actions of baseball players in 3D images (length $\times$ width $\times$ time). Such a 3D image can be obtained by stacking a sequence of 2D images of a moving object. The objective of that project is to save memory space and to analyze inappropriate actions of baseball players. The medial-line algorithm is applied to 3D medical images at the University of Iowa. The images (length $\times$ width $\times$ depth) show bronchial trees in human lungs. It is possible that our algorithms can be used eventually to save lives.

When a new thinning algorithm is designed, the designer should provide a proof showing that the algorithm is connectivity preserving. To provide such a proof, one needs to show that the thinning algorithm preserves the connectivity of every possible image. This is a difficult problem since there are infinitely many images. A number of papers by Rosenfeld, Ronse, Hall and others have proposed solutions for solving the problem for 2D images. Among these 2D results, Ronse's



results have the following contribution: for verifying the connectivity preservation of a 2D thinning algorithm, one only needs to check a finite number of configurations. Since we need to prove that our 3D algorithms preserve connectivity, we need some methods to solve such a problem for 3D images. Based on Ronse's 2D results, our second objective is to establish efficient connectivity preservation tests for 3D thinning algorithms such that whether a 3D thinning algorithm does or does not preserve connectivity can be determined by checking a finite number of configurations.

With Ronse's 2D tests and our 3D tests, one still needs to make an effort in writing and reading such proofs for 2D and 3D thinning algorithms to preserve connectivity. That process is time-consuming and error-prone. Since both Ronse's 2D results and our 3D results only need to check a finite number of configurations, we can use computers to give such proofs automatically. Hall proposed the first such 2D program. Kong established another 2D program. Hall's program takes about one hour to determine whether a 2D thinning algorithm preserves connectivity, and Kong's program takes about one minute for the same work. Kong's program needs a small memory space for storing a look-up table that was used for testing the simplicity of any object point. On the basis of Kong's program, our third objective is to establish computerized tests to verify the connectivity preservation of 3D thinning algorithms.

An important sub-test of the computerized tests is a test for testing the simplicity of any object point. Similar to Kong's 2D program, we use a look-up table to speed up the 3D computerized tests. The problem of this 3D look-up table is that the memory space of the complete look-up table is about 318 GBytes which is not feasible. Thus, our last objective is to establish a memory efficient algorithm for establishing such a 3D look-up table of a feasible size. That table should be portable, that is, it should be small enough to store in an ordinary floppy disk. The four objectives of this research can be summarized as shown in the above figure.

Contents

List of Figures

List of Tables

Chapter 1

Preliminary

This thesis studies the theory and applications related to connectivity preservation in 3D images. In this thesis, we establish necessary and sufficient conditions for 3D parallel thinning algorithms to preserve connectivity. We also implement these conditions. Such an implementation uses memory efficient lookup tables to speed-up the verification. Using above results, we design two fast 3D parallel thinning algorithms.

1.1 Domain

A *digital image* (or briefly, an *image*) can be obtained by digitizing a continuous image. In this thesis, we only consider 2D and 3D images. We assume any 2D image is embedded in $\mathbf{Z}^2$ and any 3D image is embedded in $\mathbf{Z}^3$ (i.e., each image is embedded in a regular-grid space). We do not consider images in irregular grids (see [73, 116]).

A gray-level image is an image in which each point is assigned a real number from 0 to 1 representing its gray level. Many authors have discussed operations in gray-level images (see [87, 93, 102, 104, 105, 106, 108, 130]). People are also interested in color images. The color of a point p of a color image could be obtained by mixture of the three basic colors: red, blue, and green. Thus, the color of p can be specified by finding the weight of each of the three basic colors (see [109]). For gray-level and color images, it's rather "fuzzy" to determine whether a point does or does not belong to an object (see [87, 102, 105, 106, 108]).

This thesis is based on *binary* images that can be obtained, for example, by applying a threshholding function (see [15, 17, 21, 86, 110, 128]) to gray-level images. Unless otherwise specified, we assume that all images discussed in this research are the resulting images after the application of noise-removing functions. Refer to [17, 24, 110] for noise-removing functions.

Pattern recognition is a major topic in image processing. For simplifying the process of pattern recognition, a simpler image while maintaining the same connectivity (or connectedness) characterization of the original image is necessary. Digital topology is a topic studying the connectivity properties (i.e., the *topology*) of digital images. Many papers have been published in this area (see [1, 13, 28, 29, 35, 42, 43, 44, 47, 49, 50, 51, 52, 53, 54, 55, 56, 60, 61, 75, 78, 79, 81, 82, 90, 92, 94, 95, 96, 97, 98, 100, 101, 103, 107, 111, 112, 119, 120, 122, 123]). Many operations are widely discussed for maintaining connectivity structures of binary images. We introduce some of them as follows.

Thinning This is a process similar to peeling an onion (see Figure 1.1). Its purpose is to transform every object in an input image to a unit-width skeleton in the output image. For example, a big, thick circle will be thinned into a big, thin circle where the thin circle is contained in the thick circle. See [27, 31, 33, 37, 77, 99, 118].

Shrinking This process is similar to thinning except that it generates thin and small skeletons. For example, a big, thick circle will be shrunk into a small, thin circle where the thin circle may not be contained in the thick circle. See [10, 62].

Surface tracking This process traces the boundary (or border) of each object of an image. For the 2D case, see [2, 72], for the 3D case, see [11, 12, 36, 127].

Shrinking to residue This is a process similar to shrinking except that each object is shrunk into a single point so that the number of object points in the output image is the number of objects in the input image. See [3, 121, 127].

Operations for higher dimensional images have also been discussed (see [45, 89, 126]). This thesis particularly emphasizes thinning which is a preprocessing operation of pattern recognition. A unit-width skeleton is easier to trace and hence, is easier to recognize. Generally, such a skeleton can be

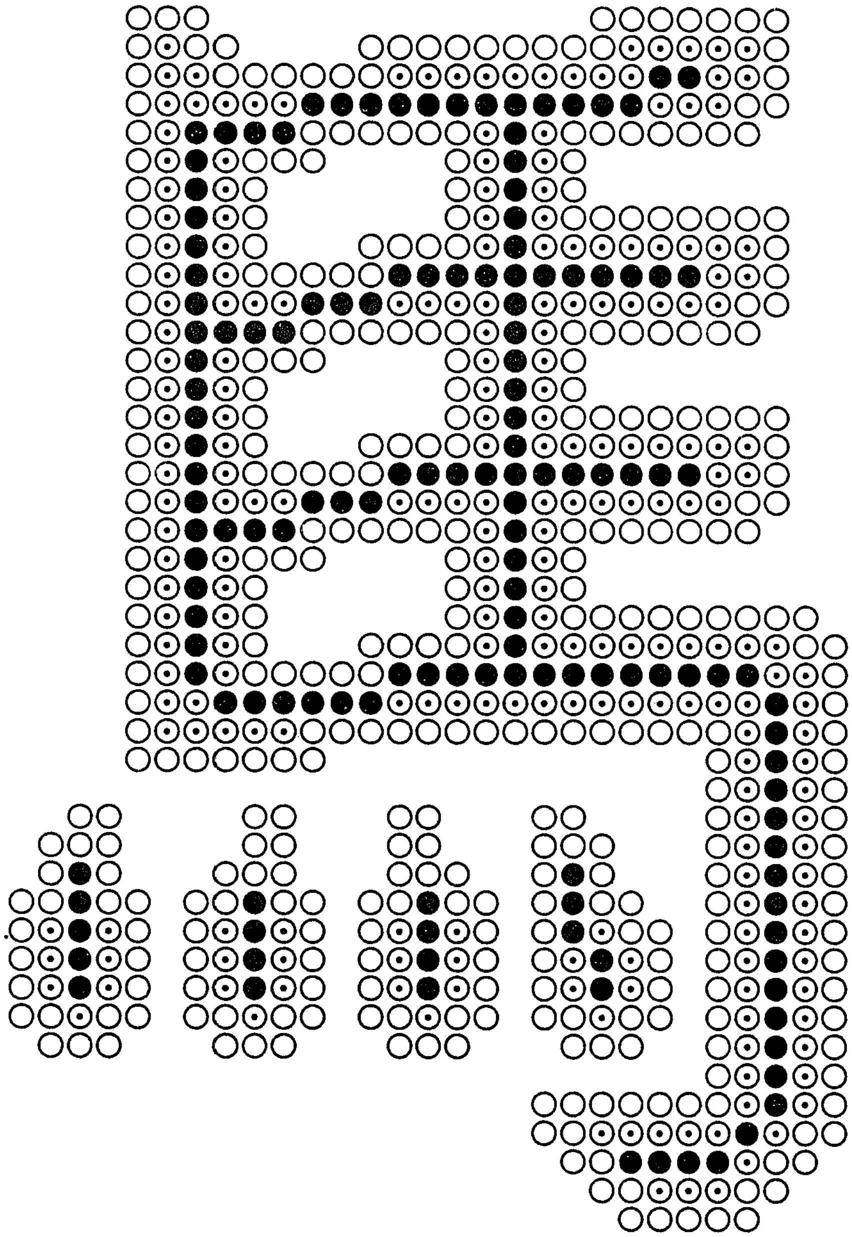


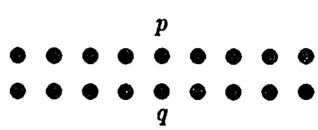
Figure 1.1: Every point marked ○ is a black point deleted in the first iteration of thinning. Every point marked ⊙ is a black point deleted in the second iteration of thinning. Every point marked ● is a black point in the skeleton.

obtained by reducing the original object layer by layer until the final result is generated. Many 2D thinning algorithms (see [5, 7, 14, 16, 18, 19, 20, 22, 23, 25, 26, 30, 33, 34, 37, 39, 40, 71, 72, 80, 85, 88, 91, 99, 113, 117, 118, 131]) and several 3D thinning algorithms (see [27, 31, 77, 124, 125]) have been proposed. When designing a thinning algorithm, one must show that for any input image to the algorithm, the output image has the same connectivity structure. If so, such a thinning algorithm is said to *preserve connectivity.*

1.2 Objectives of the thesis

A thinning algorithm is an iterative process. In each iteration, the algorithm is applied to *delete* object points (i.e., change object points to non-object points). An object point is called *simple* if the original image and the image after the point is deleted have the same connectivity structure.

A sequential thinning algorithm (see [7, 16, 113, 129]) deletes one object point in each iteration. Another kind of thinning algorithms is called the *fully* parallel thinning algorithms where the deletions are applied to check all object points in input images (see [37, 64, 67]). Thinning algorithm designers sometimes divide each iteration into several subiterations or divide the image space into several subfields. Thus there are *n-subiteration* or *n-subfield* parallel thinning algorithms where n is the number of subiterations or the number of subfields respectively. See [80, 99, 131] for 2D 4-subiteration thinning algorithms, see [30] for 2D 2-subfield thinning algorithm, and see [27, 77, 124] for 3D 6-subiteration thinning algorithms.



To prove a sequential thinning algorithm preserves connectivity, it is enough to show that any point deleted by the algorithm is simple. However, it is not so straightforward to show that a parallel thinning algorithm preserves connectivity since the parallel deletion of even two simple points may not preserve the connectivity of the original image. Consider the above figure where all unmarked points are non-object points, and points marked ● are object points. Both p and q are simple, but the deletion of both p and q changes the connectivity of the image.

A parallel thinning algorithm should be proved to preserve connectivity. To give such a proof, one must check all possible images. The difficulty is that since there are infinitely many images, we have to check infinitely many cases.

For 2D images, it is obvious that the skeleton of an elongated object is a thin line. Clearly, we do not want an object like "6" be thinned into a thin circle like "o". To guarantee that an "appropriate" skeleton can be obtained, each endpoint of a unit-width arc is preserved, that is, the endpoint of the bar of "6" should be preserved. For 3D images, it is not necessary that the skeleton of a 3D object must consist of thin lines. For an object like a square board, its skeleton could be a thin face rather than a thin line. Thus, for 3D thinning algorithms, we need to consider what kind of skeleton, thin lines or thin faces, is appropriate.

While a number of 2D connectivity preserving fully parallel thinning algorithms have been proposed, no such 3D algorithms have ever been proposed – all proposed 3D connectivity preserving parallel thinning algorithms are 6-subiteration algorithms. For any input image, one way to measure the *speed* of a parallel thinning algorithm is to count the number of iterations required to generate the output image. With an appropriate mesh computer, the speed of a fully parallel thinning algorithm is faster than any subiteration or any subfield parallel thinning algorithm. For example, for removing the outmost layer of an object in a 3D image, a fully parallel thinning algorithm needs only one application (one iteration), but a 6-subiteration parallel thinning algorithm needs six applications (six subiterations). It is of interest to establish 3D fully parallel thinning algorithms for generating thin faces and thin lines respectively.

How to prove that a 3D fully parallel thinning algorithm indeed preserve connectivity? Different methods for simplifying the proofs of 2D parallel thinning algorithms to preserve connectivity have been proposed by Rosenfeld in [99], Hall in [34], and Ronse in [94, 95]. Ronse's results showed that for proving a 2D parallel thinning algorithm preserves connectivity, only finitely many configurations need to be checked. Since the theory of 3D images is much more complicated than that of 2D images, not many 3D parallel thinning algorithms have been proposed. Compared with the 2D case, it is still difficult to provide a proof showing that a 3D parallel thinning algorithm preserves connectivity. Since there are no general solutions, it is of interest to establish efficient methods to solve this challenging problem.

The major contribution of Ronse's results is that for determining the connectivity preservation of 2D thinning algorithms only a finite number of configurations need to be checked. Since the number of such 2D configurations is finite, it is possible to use computers to establish connectivity preservation. Such computerized tests are helpful for algorithm designers to develop new and faster 2D thinning algorithms. Hall in [32] proposed the first such 2D computerized tests. Kong in [46] established another 2D computerized test which used a different approach. Both Hall's and Kong's programs are based on Ronse's 2D results. Our 3D connectivity preservation tests also depend on the following property: for determining the connectivity preservation of 3D thinning algorithms, only a finite number of configurations need to be checked. It would be of interest to construct 3D computerized tests to implement our 3D connectivity preservation tests for automatically verifying that a 3D parallel thinning algorithm preserves connectivity.

A major sub-test of Ronse's results checks whether a 2D thinning algorithm deletes only simple object points (in the 2D sense). 2D computerized tests by Hall and Kong both have such a sub-test. In our 3D connectivity preservation tests, such a sub-test (in the 3D sense) is still very important. A thinning algorithm can be stated as a set of deleting templates. For determining whether a thinning algorithm may delete non-simple object points, we need to verify all deleting templates of the algorithm. Normally, a point in any deleting template may be an object point, a non-object point or a don't-care point. An object point in the testing image with a neighborhood that matches any deleting template is deleted. In [115], a 3D two-value (object and non-object point) look-up table has been established. That table can be used to determine the simplicity of any object point in a 3D (binary) image. However, it is not enough to determine whether a deleting template of a 3D thinning algorithm can only delete simple points. Since a thinning algorithm may have deleting templates with three values, we need a 3D three-value simplicity test and such a test should be very efficient in both speed and memory aspects.

To achieve all these objectives, we:

1. develop two 3D fully parallel thinning algorithms for generating skeletons like thin lines and thin faces respectively;

2. apply the theory of digital topology to find 3D connectivity preservation tests such that whether

a 3D thinning algorithm preserves connectivity can be determined by checking a finite number of configurations;

3. provide 3D computerized tests which implement our 3D connectivity preservation tests for automatically investigating whether a 3D thinning algorithm preserves connectivity;

4. establish an efficient simplicity test for verifying whether a deleting template of a 3D thinning algorithm can only delete simple object points.

1.3 Significance of this thesis

The significance of the four objectives of this thesis can be summarized as follows.

Chapter 3 Our first objective was to design two 3D fully parallel thinning algorithms – one for generating skeletons as "medial faces" and the other one for "medial lines". Both of them are fully parallel and as a result they are faster than all existing 3D medial-face and medial-line thinning algorithms respectively. Possible applications include:

1. A multimedia team at M.I.T. is interested in the medial-face algorithm for analyzing actions of baseball players in 3D images (plane $\times$ time). Such a 3D image is a stack of a sequence of 2D images of a moving object. Their objective is to save memory space (by converting the resulting bitmap images to vector images) and to analyze inappropriate actions of baseball players.

2. The medial-line algorithm is applied to 3D medical images at the University of Iowa. The images (length $\times$ width $\times$ depth) show bronchial trees in human lungs. The resulting images after the application of our algorithm can be used eventually to save lives.

3. Both of them can be applied to pattern recognition. A simpler image with the same connectivity structure as the original image is easier to recognize. Several papers discussed the applications of thinning on Arabic, Chinese, and Japanese characters and English letters (see [63, 68, 84, 114]).

Chapter 4 and 5 Our second objective was to establish 3D connectivity preservation tests. Such 3D tests can be used in the following ways:

1. To verify the soundness of our 3D fully parallel thinning algorithms and other 3D thinning algorithms.
2. To provide a systematical method to prove the connectivity preservation of 3D thinning algorithms – such a method can replace current ad-hoc proving methods.
3. To help the designing of a new generation of 3D fully parallel thinning algorithms.

Chapter 6 The third objective was to implement our 3D connectivity preservation tests. The following tasks can be fulfilled:

1. Using heuristic methods, the 3D computerized tests can be used to verify the connectivity preservation of some 3D parallel thinning algorithms in about one minute on my 33 MHz 386 PC/AT.
2. The 3D computerized tests can be used to verify the connectivity preservation of either of our fully parallel thinning algorithms in about one hour.
3. The 3D computerized tests can be used to verify the connectivity preservation of any thinning algorithm with time complexity $O(n^4)$ where n is the number of deleting templates of the algorithm.

Chapter 7 The last objective was to construct small look-up tables which can be used to determine the simplicity of each deleting template of the algorithm. We use some techniques to reduce the size of the lookup table from 318 GBytes to 1.4 MBytes. Our small look-up tables can be stored in an ordinary floppy disk. The resulting look-up table is the first three-value table and can be applied to other connectivity preserving operations, for example, shrinking.

1.4 Results to date

In [53], digital images are classified into *conventional* images and *non-conventional* images where conventional images are used more frequently than non-conventional images. See next chapter for more details. Different methods have been proposed for determining whether a 2D parallel thinning algorithm preserves connectivity:

1. Rosenfeld in [99] provided clear examples of how to prove that a 2D 4-iteration parallel thinning algorithm preserves connectivity.
2. Hall in [34] extended Rosenfeld's proof methods to efficient tests for testing the connectivity preservation of 2D parallel thinning algorithms.
3. Ronse in [94, 95] proposed general solutions for determining the connectivity preservation of any 2D thinning algorithm.
4. Hall in [34] extended Ronse's results to the 2D hexagonal images (which are non-conventional images).

Ronse's results turn an infinite problem into a finite problem, that is, to show the connectivity preservation of a thinning algorithm, we check a finite number of patterns instead of checking an infinitely many images. For the 3D case, Hall in [32] proposed sufficient conditions for 3D parallel thinning algorithms to preserve connectivity. It can be shown that Hall's 3D results are subsets of our 3D connectivity preservation tests. Hall's results can be used to verify the connectivity preservation of 3D 2- and 4-subfield parallel thinning algorithms. However, Hall's results are not enough for the case of 3D fully parallel thinning algorithms.

Other approaches for achieving the same objective of thinning have also been proposed. For example, the erosion of mathematical morphology, contour tracing, etc. (see [6, 4, 8, 9, 38, 57, 41, 70, 83])

In [25, 71], thinning algorithms for a 2D hexagonal images have been proposed; in [31], a thinning algorithm for 3D non-conventional images was proposed. A number of 3D 6-subiteration thinning algorithms have been proposed (see [27, 77, 124, 125]). However, no connectivity preserving 3D fully parallel algorithm has ever been proposed. The reason for this poor situation is that the degree of complexity of 3D thinning is much higher than that of 2D thinning. Morgenthaler in [74] claimed a 3D fully parallel thinning algorithm. Unfortunately, his algorithm cannot guarantee to preserve connectivity for all input images.

The simplicity of a 2D object point can be verified by finding either the Hildtich number or the Rutovitz number (see [53, 114]). Another definition for a 2D object point to be simple is given by Rosenfeld in [99]. Definitions of the simplicity of a 3D object point have been proposed in

[69, 74, 115, 124]. These definitions need to use the concept of Euler characteristic and can be used for general cases; the other definition given in [69, 115] can be proved to be equivalent to a special case of the above general definition but does not need the concept of Euler characteristic.

For the computerized tests, Hall in [32] proposed the first such 2D tests which can determine the connectivity preservation of a 2D thinning algorithm in about one hour. Kong studies the same problem and proposed 2D computerized tests by another approach. Kong's program can determine whether a 2D thinning algorithm does or does not preserve connectivity in about one minute on 33 MHz 386 PC/AT. Kong's program used a look-up table for testing the simplicity of each deleting template of the testing algorithm. Since no general 3D connectivity preservation tests have been proposed, no such 3D computerized tests have been implemented.

1.5 The outline of this thesis

In Chapter 2, we give all basic definitions. We first introduce the concept of *adjacency*, and the adjacency relations between object points and non-object points. A very important and very fundamental concept, the *simplicity* of an object point, is introduced in this chapter. For parallel thinning algorithms, we need to define what it means for such algorithms to preserve connectivity. For this purpose, we use a concept of *simple sets* (of object points) suggested by Kong in [48]. If a parallel thinning algorithm only deletes simple sets, then it is said to *preserve connectivity.* We also briefly introduce the concept of the *continuous analog* of a digital image that is a continuous image converted from the digital image according to the adjacency relations.

In Chapter 3, two 3D fully parallel thinning algorithms were proposed, one for generating skeletons like medial lines and the other one for medial faces. A number of computer-generated images were given as examples. The medial-line algorithm is used at the University of Iowa. They also provided several 3D medical images showing the applications of the algorithm. For preserving geometry, this algorithm needs to preserve *tail points* which are, for example, endpoints of unit width arcs. The medial-face algorithm can be used to analyze the actions of baseball players. For preserving the geometry of the original image, we need a definition for preserving medial faces. A definition of *edge points* is given for this purpose. The medial-line algorithm has 37 deleting templates and the medial-face algorithm has 26 deleting templates. The proofs that these two algorithms preserve connectivity are given in Chapter 5 (after the 3D connectivity preservation tests are introduced in Chapter 4).

In Chapter 4, we establish 3D connectivity preservation tests. In this chapter, we use a very important concept, *minimal non-simple sets.* Such a set is non-simple but all its proper subsets are simple. This concept was first introduced by Ronse in [95] where he showed that a 2D minimal non-simple set is a subset of a 1×1 square (i.e., a unit square with four image corner points). We use the same concept to show that a minimal non-simple set for the 3D case is a subset of a $1 \times 1 \times 1$ cube (i.e., a unit cube with eight image corner points). Then we check all possible configurations of a $1 \times 1 \times 1$ cube. Since this chapter considers all possible adjacency relations of 3D conventional images, when considering simple points, we use the general definition in [53] which uses the concept of Euler characteristic.

In Chapter 5, the connectivity preservation tests were used to verify the connectivity preservation of existing 3D parallel thinning algorithms and to prove the connectivity preservation of our fully parallel thinning algorithms. We found out that the 6-subiteration parallel thinning algorithms by Gong and Bertrand [27], by Mukherjee, Das and Chatterji [77], and by Tsao and Fu [124] preserve connectivity. However, the 6-subiteration parallel thinning algorithms by Tsao and Fu [125], and by Morgenthaler [74] do not preserve connectivity. Hall in [32] proposed two sufficient conditions for 2- and 4-subfield parallel thinning algorithms to preserve connectivity. In this chapter, we proved that his results are special cases of our general 3D connectivity preservation tests. Hall in [32] claimed sufficient conditions that may be used to prove the connectivity preservation of 3D fully parallel thinning algorithms. We construct a thinning algorithm which passes his tests but cannot guarantee to preserve connectivity for all possible images.

In Chapter 6, the computerized tests are introduced. The computerized tests are used to automatically verify whether a given thinning algorithm preserves connectivity. An important fact of our 3D connectivity preservation tests is that whether a parallel thinning algorithm preserves connectivity can be determined by checking the configurations of a $1 \times 1 \times 1$ cube. Since we are dealing with parallel algorithms, we need to discuss the case when two or more corner object points of a $1 \times 1 \times 1$ cube are deleted, i.e., a thinning algorithm (with finitely many deleting templates) simultaneously delete these corner object points. A concept of *merging deleting templates* is introduced. Let n be the number of deleting templates of a thinning algorithm. The time complexity of Kong's 2D computerized tests is $O(n^2)$; the time complexity of our 3D computerized tests is $O(n^4)$. When n is big, $O(n^4)$ is not feasible. Some heuristic methods are used in our 3D computerized tests to speed up the test.

In Chapter 7, the simplicity test is introduced. This test is to test whether any deleting template of a thinning algorithm can delete non-simple object points, i.e., to determine the simplicity of a deleting template. A deleting template is a set of neighborhood configurations of an object point. If the neighborhood configuration of a certain object point is in this set, then the object point is deleted. Normally, a deleting template contains three kinds of points, object points, non-object points and don't-care points. A don't-care point can match with either an object point or a non-object point. Checking look up tables is a fast way to determine the simplicity of a deleting template. A complete look-up table needs $2,540 \times 10^{12}$ entries each of which corresponds to a distinct three-color neighborhood configuration and contains a logic value indicating the simplicity of that neighborhood configuration. If we use one bit for each entry, then we need about 318 GBytes for this table. Using the results of Kong and Saha et al. (see [115]), we establish a set of much smaller look-up tables (total 1.4 MBytes). These look-up tables are used in the 3D computerized tests for testing the simplicity of any deleting template of a given thinning algorithm.

Chapter 2

Background

In this chapter, we are going to introduce the basic definitions that will be used in this thesis. These definitions can also be found in [52, 53].

2.1 Adjacencies and connectivities

Adjacency is a fundamental relation between points of images. In this thesis, we are dealing with images embedded in $\mathbf{Z}^2$ or $\mathbf{Z}^3$. We first consider the adjacency relations of $\mathbf{Z}^2$.

Definition 2.1.1 *Let p and q be two distinct points of $\mathbf{Z}^2$ with coordinates (x_p, y_p) and (x_q, y_q) respectively. Then p and q are*

1. 4-adjacent *if* $|x_p - x_q| + |y_p - y_q| = 1$; *and*

2. 8-adjacent *if* $\max(|x_p - x_q|, |y_p - y_q|) = 1$ *and* $|x_p - x_q| + |y_p - y_q| \leq 2$.

See Figure 2.1 for the points which are 4- or 8-adjacent to a point p of a 2D image. Now we consider the adjacency relations between points of $\mathbf{Z}^3$.

Definition 2.1.2 *Let p and q be two distinct points of $\mathbf{Z}^3$ with coordinates (x_p, y_p, z_p) and (x_q, y_q, z_q) respectively. Then p and q are*

1. 6-adjacent *if* $|x_p - x_q| + |y_p - y_q| + |z_p - z_q| = 1$;

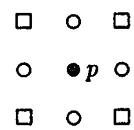


Figure 2.1: Every point marked ○ is 4-adjacent to p. Every point marked □ is diagonally adjacent to p. Every point that is 4- or diagonally adjacent to p is 8-adjacent to p.

2. 18-adjacent *if* $\max(|x_p - x_q|, |y_p - y_q|, |z_p - z_q|) = 1$ *and* $|x_p - x_q| + |y_p - y_q| + |z_p - z_q| \leq 2$*; and*

3. 26-adjacent *if* $\max(|x_p - x_q|, |y_p - y_q|, |z_p - z_q|) = 1$ *and* $|x_p - x_q| + |y_p - y_q| + |z_p - z_q| \leq 3$.

See Figure 2.2 for the points which are 6-, 18-, and 26-adjacent to a point p of a 3D image.

Definition 2.1.3 *Two sets, X and Y, of points of $\mathbf{Z}^2$ or $\mathbf{Z}^3$ are* k-adjacent *if some point of X is k-adjacent to some point of Y.*

The definitions of *neighbors* and *neighborhoods* are as follows.

Definition 2.1.4 *Let p and q be two distinct points of $\mathbf{Z}^2$ or $\mathbf{Z}^3$. Then p is a* k-neighbor *of q if they are k-adjacent where $k = 4$ or 8 of $\mathbf{Z}^2$ and 6, 18 or 26 of $\mathbf{Z}^3$.*

Definition 2.1.5 *Let p be a point of a $\mathbf{Z}^2$ or $\mathbf{Z}^3$ space. Then the* k-neighborhood *of p, denoted by $N_k(p)$, is the set of p and all p's k-neighbors. Let $N(p)$ be $N_8(p)$ in $\mathbf{Z}^2$ and $N_{26}(p)$ in $\mathbf{Z}^3$.*

The definitions of *paths* and *closed curves* are as follows.

Definition 2.1.6 *Let X be a set of points of $\mathbf{Z}^2$ or $\mathbf{Z}^3$. A* k-path *of X is a sequence $\langle x_1, x_2, \ldots, x_n \rangle$ such that each x_i is a point of X and x_j is k-adjacent to x_{j+1} for $1 \leq j < n$.*

Definition 2.1.7 *Let X be a set of points of $\mathbf{Z}^2$ or $\mathbf{Z}^3$. A* closed k-curve *of X is a k-path $\langle x_1, x_2, \ldots, x_n \rangle$ of X where $x_1 = x_n$.*

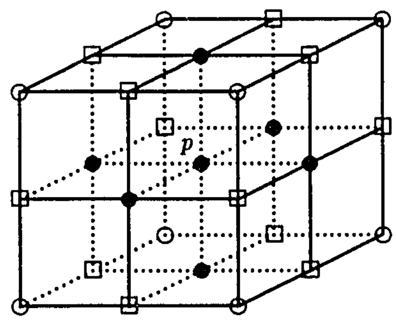


Figure 2.2: Every point marked ● (except p itself) is 6-adjacent to p, every point marked □ is diagonally adjacent to p, and every point marked ○ is diametrically adjacent to p. Every point that is 6- or diagonally adjacent to p is 18-adjacent to p, every point that is 18- or diametrically adjacent to p is 26-adjacent to p.

Definition 2.1.8 *Let X be a set of points of $\mathbf{Z}^2$ or $\mathbf{Z}^3$, and let p and q be two distinct points of X. Then p and q are said to* k-connected *if there exists a k-path $\langle p = x_1, x_2, \ldots, x_n = q\rangle$ of X.*

Before moving to the next section, we introduce the following basic definitions.

Definition 2.1.9 *In $\mathbf{Z}^2$ and $\mathbf{Z}^3$, we define the following terms.*

1. *A* unit lattice edge *is a unit edge with both endpoints embedded in $\mathbf{Z}^2$ or $\mathbf{Z}^3$;*
2. *A* unit lattice square *is a unit square with all four corner points embedded in $\mathbf{Z}^2$ or $\mathbf{Z}^3$; and*
3. *A* unit lattice cube *is a unit cube with all eight corner points embedded in $\mathbf{Z}^3$.*

Definition 2.1.10 *Let p and q be two distinct points of $\mathbf{Z}^n$ where $n = 2$ or 3. Then p and q are*

1. diagonally adjacent *if $n = 2$ and they are 8- but not 4-adjacent, or $n = 3$ and they are 18- but not 6-adjacent; and*
2. diametrically adjacent *if $n = 3$ and they are 26- but not 18-adjacent.*

2.2 Binary digital images

In this section we define digital images and the adjacency relations used in digital images. The following definition is given by Kong et al. in [53].

Definition 2.2.1 *A* binary digital image *(or a* digital image, *or briefly an* image*) P is a quadruple* $(\mathbf{Z}^n, \beta, \omega, B)$ *where*

1. *$n = 2$ or 3;*
2. *each point of P is of either value 1 (a* black point*) or value 0 (a* white point*);*
3. *β is an adjacency used between two black points;*
4. *ω is an adjacency used between two white points and between one white point and one black point; and*
5. *$B \subseteq \mathbf{Z}^n$ is a finite set of all black points of P.*

Definition 2.2.2 *Let $P = (\mathbf{Z}^n, \beta, \omega, B)$ be an image where $n = 2$ or 3. If $n = 2$, P is also called a* 2D (β, ω) image*; if $n = 3$, P is called a* 3D (β, ω) image. *The pair (β, ω) is called the* adjacency pair *of P.*

Note that in this thesis we only consider images containing finitely many black points.

Definition 2.2.3 *Let P be a 2D or 3D (β, ω) image.*

1. *Two black points are* k-adjacent *if they are k-adjacent in the embedded space. If $k = \beta$, then they are simply called* adjacent.
2. *Two white points, or one black point and one white point are* k-adjacent *if they are k-adjacent of the embedded space. If $k = \omega$, then they are simply called* adjacent.

Let P be a 2D or 3D image without indicating the adjacency pair. We can still say that two black points or two white points of P are *adjacent*. This is because P must be equipped with a pair of adjacency relations (β, ω) and the statements "two black points of P are adjacent" and "two

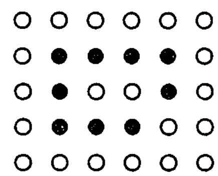


Figure 2.3: A "paradox" image which uses the (8,8) or (4,4) adjacency pair. How many white components are there? How many black closed curves are there?

white points of P are adjacent" just means that the two black points are β-adjacent and the two white points are ω-adjacent, respectively.

By "$p = 1$" we mean "p is a black point", and by "$p = 0$" we mean "p is a white point". The Jordan curve theorem is a very important property in digital topology. For the 2D case, the Jordan curve theorem shows that a closed curve (like the letter "o") separates the 2D space into two parts — one in the closed curve and the other one outside the closed curve. For 2D (8,8) images, consider Figure 2.3, since the black points are equipped with the 8-adjacency, the object is indeed a closed curve which by the Jordan curve theorem should separate the 2D space into two disconnected subsets. However, since the white points are also equipped with the 8-adjacency, the set of white points "in" the closed curve is 8-adjacent to the set of white points "out" of the closed curve. Thus, we cannot use the 8-adjacency for both black and white points of 2D images. If we apply the (4,4) adjacency pair to the same configuration in Figure 2.3, then the set of white points "in" the black object is not 4-adjacent to the set of white points "out" of the black object. However, the black object itself is not a closed curve so it cannot separate the 2D space into two parts. Thus, using the 4-adjacency for both black and white points is not appropriate either. The two generally used adjacency pairs of 2D orthogonal lattice grids are:

1. the (8,4) pair: use the 8-adjacency between black points, and the 4-adjacency between white points and between black points and white points; and

2. the (4,8) pair: use the 4-adjacency between black points, and the 8-adjacency between white points and between black points and white points.

As in the 2D case, while some pairs of adjacency relations are appropriate to be applied to 3D images, some other pairs of adjacency relations are not appropriate for 3D images. Following a similar argument as in the 2D case, we know that (26,26), (26,18), (18,26), (18,18), and (6,6) are not appropriate adjacency pairs for 3D images. The four generally used adjacency pairs of the 3D case are:

1. the (26,6) pair: use the 26-adjacency between black points, and the 6-adjacency between white points and between black points and white points;
2. the (18,6) pair: use the 18-adjacency between black points, and the 6-adjacency between white points and between black points and white points;
3. the (6,26) pair: use the 6-adjacency between black points, and the 26-adjacency between white points and between black points and white points; and
4. the (6,18) pair: use the 6-adjacency between black points, and the 18-adjacency between white points and between black points and white points.

The 2D (8,4) images are the most generally considered 2D images, and the 3D (26,6) images are the most generally considered 3D images. Images embedded in non-orthogonal grids were also considered. In [34], 2D (6,6) images were discussed, and in [31], 3D (12,12) images were discussed. See Figure 2.4 for the 6-neighborhood of 2D (6,6) images and the 12-neighborhood of 3D (12,12) images. We need the following notations in the following chapters.

Definition 2.2.4 *Let $P = (\mathbf{Z}^n, \beta, \omega, B)$ be a 2D or 3D image and let X be a set of points of P. We define the following terms.*

1. *Let $P - X$ be the image obtained from P by deleting all black points of X, that is, $P - X = (\mathbf{Z}^n, \beta, \omega, B - X)$.*
2. *Let $P \cap X$ be the image obtained from P by deleting all black points out of X, that is, $P \cap X = (\mathbf{Z}^n, \beta, \omega, B \cap X)$.*

Definition 2.2.5 *Let $P = (\mathbf{Z}^n, \beta, \omega, B)$ be a 2D or 3D image. Then the* complement *image of P, denoted by $\overline{P}$, is the image $(\mathbf{Z}^n, \omega, \beta, \mathbf{Z}^n - B)$.*

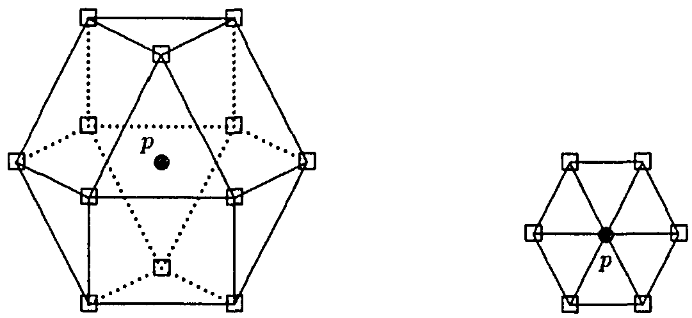


Figure 2.4: In (a), the 12-neighborhood of p of a 3D (12,12) image, in (b), the 6-neighborhood of p of a 2D (6,6) image.

In (β, ω) images, we define the concepts of *connection* and *components* as follows.

Definition 2.2.6 *Let X be a set of points of a 2D or 3D (β, ω) image P. Then a* black k-path *of X is a k-path consisting of black points of X. If $k = \beta$, then such a path is called a* black path *of X. A* white l-path *of X is an l-path consisting of white points of X. If $l = \omega$, then such a path is called a* white path *of X.*

Definition 2.2.7 *Let X be a set of points of a 2D or 3D (β, ω) image P. Then a* black closed k-curve *of X is a closed k-curve consisting of black points of X. If $k = \beta$, then such a curve is called a* black closed curve *of X. A* white closed l-curve *of X is a closed l-curve consisting of white points of X. If $l = \omega$, then such a curve is called a* white closed curve *of X.*

Definition 2.2.8 *Let X be a set of points of a 2D or 3D (β, ω) image P. Then two distinct black points, p and q, of X are called* k-connected *if there exists a black k-path $\langle p = x_1, x_2, \ldots, x_m = q \rangle$ of X. If $k = \beta$, then p and q are called* connected *in X. Two distinct white points, u and v, of X are called* l-connected *if there exists a white l-path $\langle u = y_1, y_2, \ldots, y_n = v \rangle$ of X. If $l = \omega$, then u and v called* connected *in X.*

Definition 2.2.9 *Let X be a set of points of a 2D or 3D (β, ω) image P. A* black k-component *of X is a maximal k-connected subset of the set of black points of X. If $k = \beta$, then such a black* *k-component is called a* black component *of X. Note that we also call a black component an* object. *A* white *l*-component *of X is a maximal l-connected subset of the set of white points of X. If* $l = \omega$, *then such a white k-component is called a* white component *of X. There is a unique white component called* background *of any image that is an infinite white component.*

Definition 2.2.10 *A* hole *of a 2D* (β, ω) *image is a finite white* $(\omega$-*)component. A* cavity *of a 3D* (β, ω) *image is a finite white* $(\omega$-*)component.*

For the 2D case, a typical hole is the inner white component surrounded by a black component like "o". For the 3D case, a typical cavity is the inner white component contained in a hollow ball. "Tunnels" are special 3D configurations. Although we know that the central hole of a donut is a "tunnel", it is surprisingly difficult to define a "tunnel". "Tunnels" will be discussed later.

Definition 2.2.11 *Let p be a black point of a 2D or 3D* (β, ω) *image P. Then p is called*

1. *a* border point *if p is* $(\omega$-*)adjacent to a white point;*

2. *an* isolated point *if p has no black* $(\beta$-*)neighbors;*

3. *an* end point *if p has exactly one black* $(\beta$-*)neighbor and P is a 2D image; and*

4. *a* tail point *if p has exactly one black* $(\beta$-*)neighbor and P is a 3D image.*

Definition 2.2.12 *In a 2D or 3D* (β, ω) *image, a set of black points is called* pairwise *k*-adjacent *if every pair of points in the set are k-adjacent. If* $k = \beta$, *then such a set of black points is called* pairwise adjacent. *A set of white points is called* pairwise *l*-adjacent *if every pair of points in the set are l-adjacent. If* $l = \omega$, *then such a set of white points is called* pairwise adjacent.

Definition 2.2.13 *The* deletion *of a black point p from a 2D or 3D image P is a process which converts p to a white point. The* deletion *of a set of black points from P is a process which converts every point of the set to a white point.*

2.3 Connectivity preserving deletion of points

Thinning is a connectivity preserving operation. A thinning algorithm is a skeletonizing operation where only the deletion is allowed to apply to the points of a given image. A skeleton must preserve

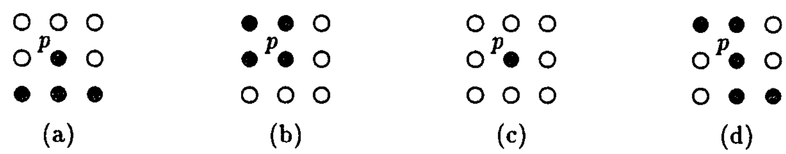


Figure 2.5: On 2D images, p in (a–b) are simple points; p in (c–d) are non-simple points.

the connectivity structure (the *topology*) of the original black component. A connectivity preserving thinning algorithm of 2D or 3D images must satisfy all the following conditions: (1) Never delete a black component completely. (2) Never merge two black components. (3) Never split one black component into two or more black components. (4) Never create a black component. (5) Never vanish a white component completely. (6) Never merge two white components. (7) Never split one white component into two or more white components. (8) Never create a white component.

Since thinning algorithms only allow black points to be deleted, we do not need to consider the case when black points are added to the image. Let P be a 2D image where D is a set of black points of P. Then the image $P - D$ preserves the connectivity of P if all of the following conditions hold:

- No black component of P is split into two black components of $P - D$.
- No black component of P is deleted completely.
- No two white components of P are merged into one white component of $P - D$.
- No white component of $P - D$ is created.

These conditions are *not* enough to show the deletion of a set of black points from a 3D image is connectivity preserving. Consider a donut-like object "o" of a 3D image. If it is turned into an object like "c" after the deletion of a portion of the object, then all above conditions are satisfied but the connectivity is not preserved. (In this case, a "tunnel" is deleted.) Although it is mathematically difficult to define the special configuration, "tunnels", of 3D images, the number of "tunnels" can be obtained without a definition of "tunnels" (see [52, 53]). More details will be introduced in Chapter 4.

Let's suppose the number of "tunnels" of a 3D image P is obtained and denoted by $\#_T(P)$. Define $\#_B(P)$ and $\#_C(P)$ to be the number of black components and cavities of P respectively. Then we have the following definition of the Euler characteristic of P, denoted by $\chi(P)$.

Definition 2.3.1 [52] *Let P be a 3D image. Then* $\chi(P) = \#_B(P) - \#_T(P) + \#_C(P)$.

An important concept, *simple points*, of 2D and 3D images can be defined as follows.

Definition 2.3.2 [53] *Let p be a black point of a 2D or 3D image P. Then p is a* simple point *of P if and only if all of the following conditions hold:*

1. *p is adjacent to only one black component in* $N(p) - \{p\}$*;*
2. *p is adjacent to only one white component in* $N(p)$*; and*
3. *for the 3D case;* $\chi(P \cap N(p)) = \chi(P \cap (N(p) - \{p\}))$.

It can be shown that the $P - \{p\}$ has the same connectivity as P if and only if p is a simple point of P (see [53]). In [69, 115], an alternative characterization of simple points of 3D (26,6) images was proposed. It can be shown that the alternative characterization is equivalent to the (26,6) case of the above general definition. The alternative characterization is useful since it does not need to consider the concept of Euler characteristic. We state the alternative characterization of 3D (26,6) simple points as follows.

Proposition 2.3.3 *Let p be a black point of a 3D* (26,6) *image P. Then p is a* simple point *of P if and only if all of the following conditions hold:*

1. *p is adjacent to only one black component in* $N(p) - \{p\}$*; and*
2. *p is adjacent to only one white component in* $N_{18}(p)$.

Some examples of simple and non-simple points of 2D (8,4) or (4,8) images, and of 3D (26,6), (18,6), (6,26), or (6,18) images are given in Figure 2.5 and Figure 2.6, respectively.

Sequential thinning algorithms can be found in [2, 7, 113]. If a sequential thinning algorithm can only delete simple points, then the algorithm is connectivity preserving. A parallel thinning algorithm can delete two or more points at the same time where each deleted point satisfies at least

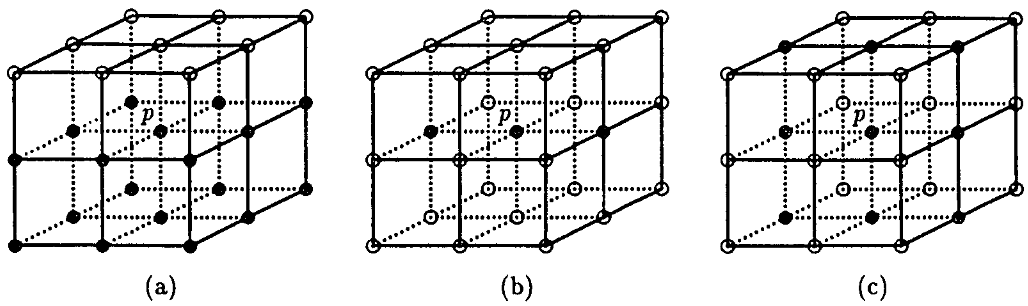


Figure 2.6: In 3D images, p is a simple point in (a) and p is non-simple points in (b–c).

one deleting template of the algorithm. A specific problem in parallel thinning is that the deletion of a set of simple points simultaneously might not guarantee preservation of connectivity.

Since a parallel thinning algorithm may delete more than one black point at the same time, we need a definition for the deletion of a set of black points to be connectivity preserving. The following definition was suggested by Kong.

Definition 2.3.4 *Let D be a set of black points of a 2D or 3D image P. Then*

1. *D is called a* simple sequence *of P if D can be ordered as a sequence in which every point is simple after all its predecessors in the sequence are deleted from P;*

2. *D is called a* simple set *of P if D can be ordered as a simple sequence of P; and*

3. *the deletion of D from P is said to* preserve connectivity *if D is a simple set of P.*

Note that the 2D version of this definition was proposed by Ronse in [94]. We still need a precise definition of what it means for a parallel thinning algorithm to preserve connectivity. This definition was suggested by Kong in [48].

Definition 2.3.5 *A 2D or 3D parallel thinning algorithm is said to* preserve connectivity *if it is only allowed to delete simple sets from a 2D or 3D image respectively.*

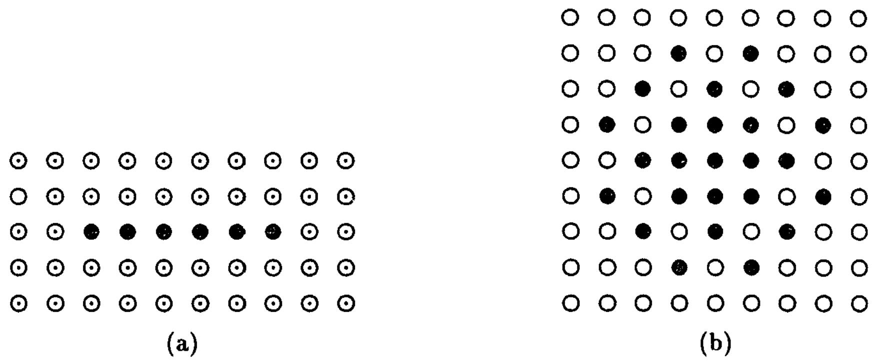


Figure 2.7: Possible results of thinning: (a) shows an "appropriate" skeleton of an object where a point marked ⊙ was originally a black point but was deleted by a thinning algorithm, (b) shows an object in which no black point can be deleted without changing the connectivity of the original image.

2.4 Thinning

Thinning is a process for generating a skeleton of each object of an image. The parallel thinning is an operation like "pealing an onion". Each time the outmost layer of an object is removed. A parallel thinning algorithm terminates when no points can be deleted (see Figure 1.1). Intuitively, for each object of a 2D or 3D image, an "ideal" thinning algorithm should generate a skeleton satisfying all the following properties. The third property is discussed in the previous section. We discuss the other three properties in this section.

- The *skeleton* must be as thin as possible.
- The *skeleton* must be as medial as possible.
- The *skeleton* must preserve connectivity.
- The *skeleton* must preserve geometry.

For example, see Figure 2.7.(a) for a 5×10 black component of a 2D (8,4) image, the resulting black object is appropriate to be taken as a skeleton of the original object. (Since such a skeleton satisfies all above four properties). However, for the object in Figure 2.7.(b), no black points can be deleted without changing the connectivity of the original image. Intuitively, the object in Figure 2.7.(b) is not "of unit width", but it is indeed as thin as possible (since no points can be deleted).

Another requirement of thinning is that the skeleton should be as medial as possible in the object containing it. For example, in Figure 2.7.(a), the skeleton is "as medial as possible" in the original object. However, for an elongated object whose width is an even number, what is an appropriate skeleton? Consider the two objects shown in Figure 2.8. Although both skeletons are "as thin as possible", we prefer the skeleton shown in Figure 2.8.(b).

Thinning algorithms should preserve "geometry" as well. This implies that the skeleton generated by a thinning algorithm must "look like" the original object. For example, an object like "d" cannot be thinned into a object like "o" (although the connectivity is preserved). To preserve the geometrical properties of a 2D image, an endpoint of a unit width arc cannot be deleted.

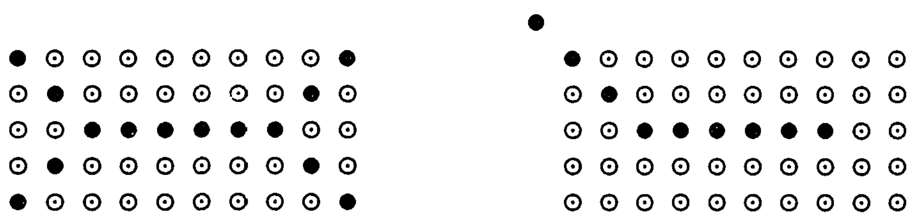
It is rather vague when we say a skeleton is "as medial as possible" or "as thin as possible". This is also true when we say that a skeleton "preserves geometry" of the original object. The two skeletons shown in Figure 2.8 can be taken as examples since both of them satisfy all the three properties. Consider the skeleton shown in Figure 2.9.(a). It can be obtained by the medial-line transformation. However, it is hard to tell whether such a skeleton is better than the skeleton shown in Figure 2.7.(a). Normally, the decision is made by practioners for different applications. A small branch of an object might also change the final skeleton significantly (compare the skeletons in Figure 2.7.(a) and in Figure 2.9.(a)). Normally, a very small branch can be taken as a noise of the image. However, it is difficult to tell whether a branch of a certain length is or is not a noise of the image.

2.5 Continuous analogs

In the previous sections, we considered the 2D images embedded in $\mathbf{Z}^2$. For this combinatorial approach, a pair of adjacency relations are employed. Using such relations of 2D images, some well-defined properties, such as the adjacency tree, the Jordan curve theory, the minimal 4- and 8-circles, etc., are discussed (see [96, 97]). Many 2D thinning algorithms [30, 37, 118] are based on

(a) (b)

Figure 2.8: Which skeleton is "appropriate" for an $n \times 4$ object? A point marked $\odot$ is originally a black point but deleted by a thinning algorithm.



(a) (b)

Figure 2.9: Is the skeleton in (a) "appropriate" for an $n \times 5$ object? Is the skeleton in (b) "appropriate" for an object with a small branch? A point marked $\odot$ is originally a black point but deleted by a thinning algorithm.

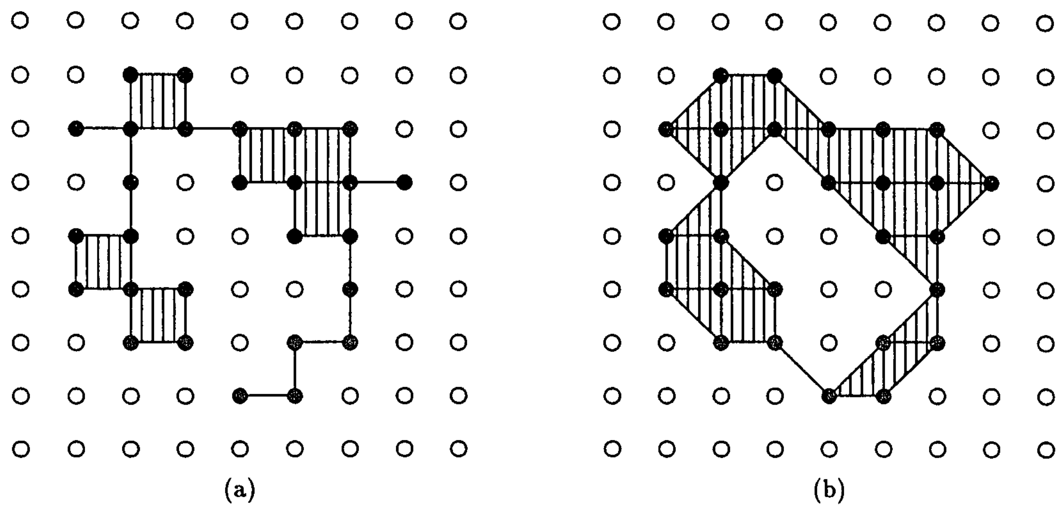


Figure 2.10: Examples of continuous analogs. In (a), we use the (4,8) adjacency pair, and in (b), we use the (8,4) adjacency pair.

this approach. Generally, this approach is enough for 2D thinning.

Some concepts of 3D digital topology are also based on this approach (see [103]) where 6-, 18- and 26-adjacencies are the most widely used adjacency relations. Well-defined properties, for example the Jordan surface theory, were discussed. For the 3D case, the combinatorial approach is not very efficient.

Consider a point p of a 2D or a 3D image. Some very important properties of thinning are based on the local neighborhood of p where such a neighborhood contains 8 points surrounding p for the 2D case and 26 points surrounding p for the 3D case. Thus, there are only $2^8 = 256$ different black and white configurations of such a local neighborhood on 2D images, but there are $2^{26} = 67,108,864$ different black and white configurations of such a neighborhood in 3D images.

Another approach was proposed for the continuous analog of a digital image (see [52, 53]). The concept of this approach is to apply the properties of the continuous topology to digital images (see [47, 54, 75]). Let P be a 2D or 3D image. We briefly introduce the concept of the *continuous analog* of P, denoted by $C(P)$, and the process for constructing $C(P)$ as follows. (Refer to [52, Section 5] for the details of point-adding and line-connecting rules.)

1. Connect every pair of 6-adjacent points of P by a unit-length line segment.

2. Add one diagonal line in every unit lattice square of P based on the black/white configuration of the square and the (β, ω) adjacency pair of P. Each unit lattice square is uniquely subdivided into two equilateral triangles.

3. Add a centroid point in each unit lattice cube of P, and connect it to all corners of the cube. The color of each centroid point is either black or white based on the black/white configuration of the cube and the (β, ω) adjacency pair of P. Each unit lattice cube is subdivided into twelve equilateral tetrahedra.

4. If the two endpoints of a line segment added by the rules shown above are both black points (one of which could be a centroid point), then call the line segment a *black edge*; otherwise call it a *white edge.*

5. If all three corners of a triangle are black points (one of which could be a centroid point) where edges of the triangle are added by the rules shown above, then call the triangle a *black triangle*; otherwise call it a *white triangle.*

6. If all four corners of a tetrahedron are black points (one of which is a centroid point) where edges of the tetrahedron are added by the rules shown above, then call the tetrahedron a *black tetrahedron*; otherwise call it a *white tetrahedron.*

Let $C(P)$ be the union of all black points, black edges, black triangles and black tetrahedra that are obtained by above procedures on P. Then $C(P)$ is called the *continuous analog* of P where $C(P)$ is a continuous image. Figure 2.10 showed two continuous analogs of the same set of points of a 2D image where we use the (4,8) adjacency pair in (a), and the (8,4) adjacency pair in (b). In (a), there are only one black component and only one white component in the image; in (b), there are only one black component, but two white components in the image.

Chapter 3

3D fully parallel thinning algorithms

A thinning algorithm should preserve connectivity, i.e., any object and its skeleton should maintain the same connectivity structure. Since 3D (26,6) images are the most widely discussed 3D images, in this chapter, we propose two fully parallel thinning algorithms for 3D (26,6) images – one for generating skeletons like "medial faces", and the other one for generating skeletons like "medial lines". The medial-line algorithm is also introduced in a joint paper with Dr. Milan Sonka (see [67]).

Since in this chapter we only consider (26,6) images, we use Proposition 2.3.3 to characterize simple points in such 3D images. The two thinning algorithms also demonstrate possible ways for designing a new generation of 3D parallel thinning algorithms. Note that in all figures of this thesis, a point marked ● is a black point and a point marked ○ is a white point. Furthermore, from now on, unless otherwise specified, when we say "thinning algorithm" we mean "parallel thinning algorithm".

The orientations of the x-, y- and z-axis are shown in Figure 3.1. Suppose p is a point in a 3D image. Let $e(p)$, $w(p)$, $n(p)$, $s(p)$, $u(p)$ and $d(p)$ be the east, west, north, south, up, and down neighbors of p, respectively (see Figure 3.1). Furthermore, let $nu(p)$, $nd(p)$, $ne(p)$, $nw(p)$, $su(p)$, $sd(p)$, $se(p)$, $sw(p)$, $wu(p)$, $wd(p)$, $eu(p)$, and $ed(p)$ be the north-up, north-down, north-east, north-west, south-up, south-down, south-east, south-west, west-up, west-down, east-up, east-down

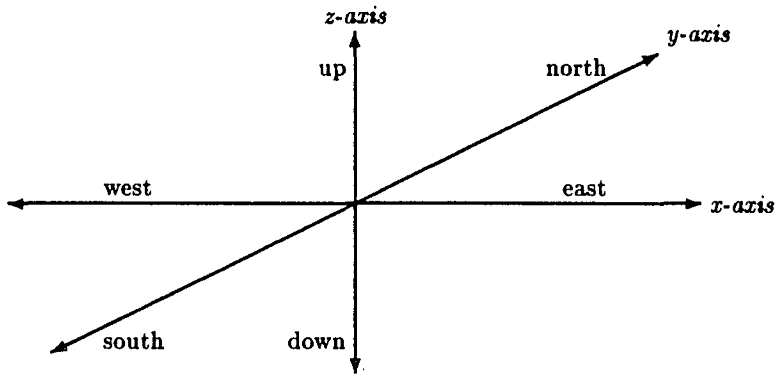


Figure 3.1: The x-, y- and z-axis, and the arrangement of each orientation.

neighbors of p, respectively.

3.1 Existing algorithms

Generally, thinning algorithms are applied iteratively to remove the outmost layer of every black component of an image. Generally, there are three classes of parallel thinning algorithms:

n-subiteration parallel thinning algorithms There are n subiterations in each iteration. This term specifically refers to the algorithms where only a certain kind of border points are considered for deletion in each subiteration. For 3D images, since there are six kinds of border points, normally we consider the case of six subiterations. A number of 6-subiteration parallel thinning algorithms have been proposed. We will discuss some of them later.

n-subfield parallel thinning algorithms Each image is subdivided into n subfields and the thinning algorithm is alternatively applied to all black points in each subfield. Hence, there are actually n subiterations in each iteration. Two possible such 3D algorithms are:

- 2-subfield parallel thinning algorithms where two points of a 3D image are in the same subfield if and only if they are connected by a 26-path in which every two consecutive points are diametrically adjacent; and

- 4-subfield parallel thinning algorithms where two points of a 3D image are in the same subfield if and only if they are connected by an 18-path in which every two consecutive points are diagonally adjacent.

Fully parallel thinning algorithms Algorithms from this group are applied parallelly to all black points of the given image. Hence, there is only one application of the thinning algorithm in each iteration. Intuitively, a fully parallel thinning algorithm is "faster" than any algorithm described above. (Practically, "faster" means that the number of applications of the algorithm for generating an output image is smaller).

For the 2D case, since there are only four kinds of border points, Rosenfeld in [99] proposed a famous 4-subiteration thinning algorithm that is to alternatively delete all north, east, south and west border simple non-end points. Morgenthaler in [74] proposed a definition of "end-points" for the 3D case and studied the problem for constructing 3D 6-subiteration and fully parallel thinning algorithms. Tsao and Fu in [125] studied the same problem of 3D 6-subiteration thinning algorithms by using their own definition of 3D "end points".

Tsao and Fu in [124] proposed a two-step 6-subiteration parallel thinning algorithm for 3D (26,6) images. After the first step, each black component of the original image is converted into a medial face. After the second step, each medial face is converted into a medial line. Suppose p is the origin of $\mathbf{Z}^3$. Then $N_x(p)$ is the set of all points in $N(p)$ with x-coordinates being 0. Define $N_y(p)$ and $N_z(p)$ similarly. The first step of their algorithm can be proved to preserve connectivity by showing the correctness of the following proposition. We omit the second step of their algorithm. The proof of the following proposition is in Chapter 5.

Proposition 3.1.1 *The deletion of all north border black points p in 3D $(26,6)$ images preserves connectivity if all the following conditions hold:*

1. *p is simple;*

2. *p is adjacent to only one black component in $N_x(p)$ and $N_z(p)$;*

3. *p is adjacent to at least two black points in $N_x(p)$ and $N_z(p)$; and*

4. *p is adjacent to at least two black points in $N(p)$.*

Gong and Bertrand in [27] proposed another 6-subiteration parallel thinning algorithm for 3D (26,6) images. The output of this algorithm is an image containing a set of medial faces. Let a *two-cube* be the union of two unit lattice cubes sharing a common unit lattice square. Theie algorithm can be proved to preserve connectivity by showing the correctness of the following proposition. The proof of the following proposition is in Chapter 5.

Proposition 3.1.2 *The deletion of all north border points p in 3D* $(26,6)$ *images preserves connectivity if all the following conditions hold:*

1. $s(p) = 1$;
2. *p is adjacent to only one black component in* $N_x(p)$ *and* $N_z(p)$*; and*
3. *p is adjacent to only one black component in every two-cube containing* $\{p, s(p), n(p)\}$.

In [77], a 6-subiteration parallel thinning algorithm called *MESPTA* for 3D (26,6) images was proposed that is extended from a 2D thinning algorithm called SPTA (see [80]). This algorithm also generates an image containing a set of medial faces. In their algorithm, two "slant" 2D 3×3 neighborhoods of p were used. Generally, these "slant" 2D neighborhoods can be obtained by 45 degree rotations of $N_x(p)$ and $N_z(p)$ respectively according to the y-axis. Let $N_1(p)$ be such a "slant" 2D neighborhood of p where $N_1(p)$ contains p, $n(p)$, $s(p)$ and their east-up and west-down neighbors. Let $N_2(p)$ be another "slant" 2D neighborhood of p where $N_2(p)$ contains p, $n(p)$, $s(p)$ and their east-down and west-up neighbors. The algorithm *MESPTA* can be stated as follows. The proof of the following proposition is in Chapter 5.

Proposition 3.1.3 *The deletion of all north border points p on 3D images preserves connectivity if all following conditions hold:*

1. $s(p) = 1$;
2. *p is adjacent to only one black component in* $N_x(p)$ *and* $N_z(p)$*; and*
3. *p is adjacent to only one black component in* $N_1(p)$ *and* $N_2(p)$.

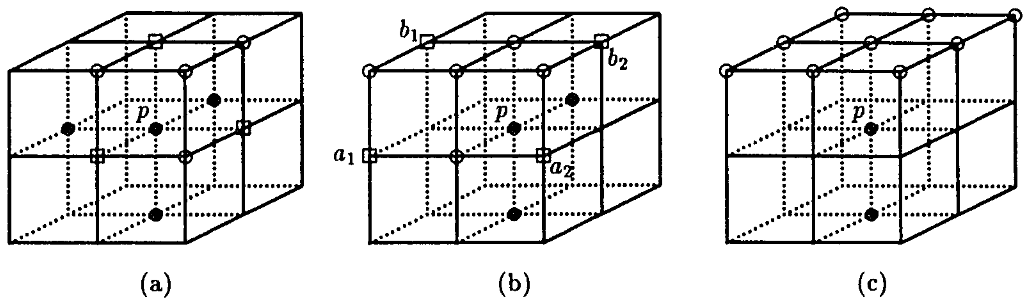


Figure 3.2: Three basic configurations of our medial-face parallel thinning algorithm. In (a), at least one point marked □ is a white point, in (b) at least one point in $\{a_1, b_1\}$ and at least one point in $\{a_2, b_2\}$ are white points, and in (a–c), every unmarked point is either a black point or a white point.

3.2 The medial-face thinning algorithm

Our first algorithm is for generating images containing skeletons as medial faces. To generate such skeletons, we need to preserve *edge points* which are defined as follows.

Definition 3.2.1 *Let p be a black point of a 3D (26,6) image. Then p is called an* edge point *if p has exactly one black neighbor in $N(p)$, $N_x(p)$, $N_y(p)$ or $N_z(p)$; p is called a* non-edge point *if it is not an edge point.*

For establishing the proof of the connectivity preservation of our medial-face algorithm, we need the following preserving condition.

Rule 3.2.2 *Suppose all four corners of a unit lattice square are to be deleted. Then the corner with the smallest sum of coordinates is preserved if and only if it is non-simple after the other three corners are deleted.*

Consider the configurations shown in Figure 3.2. Let $\mathcal{O}_f$ be the set of all rotations and reflections of all three configurations. For each element T of $\mathcal{O}_f$, a black point p is said to *satisfy* T if all of the following conditions are satisfied:

1. $N(p)$ matches T (where a don't-care point in T matches either a black point or a white point of $N(p)$);
2. p is a non-edge point;
3. if p is a north border point in T, then $s(s(p)) = 1$;
4. if p is an east border point in T, then $w(w(p)) = 1$; and
5. if p is an up border point in T, then $d(d(p)) = 1$.

A black point p is said to *satisfy* $\mathcal{O}_f$ if p satisfies any element in $\mathcal{O}_f$. We now introduce our medial-face algorithm.

Algorithm 3.2.3

repeat

parallel delete every black point that satisfies $\mathcal{O}_f$ *and is not preserved by Rule 3.2.2;*

until *no points are deleted;*

Algorithm 3.2.3 terminates when no black points can be deleted. Since we assume all input images contain finitely many black points, this algorithm will eventually terminate. Different shapes of objects have been tested by this parallel algorithm. Here we present only four examples (see Figure 3.3 to 3.6).

3.3 The medial-line thinning algorithm

The deletion conditions

Clearly, this algorithm is for generating skeletons like medial-lines. Let $\mathcal{O}_l$ be the set of all rotations and reflections of the four configurations shown in Figure 3.7.(a–d), i.e., $\mathcal{O}_l$ is a set of $3 \times 3 \times 3$ configurations. Then for each element T of $\mathcal{O}_l$, a black point p is said to *satisfy* T if all of the following conditions are satisfied, and a black point p is said to *satisfy* $\mathcal{O}_l$ if p satisfies any element in $\mathcal{O}_l$.

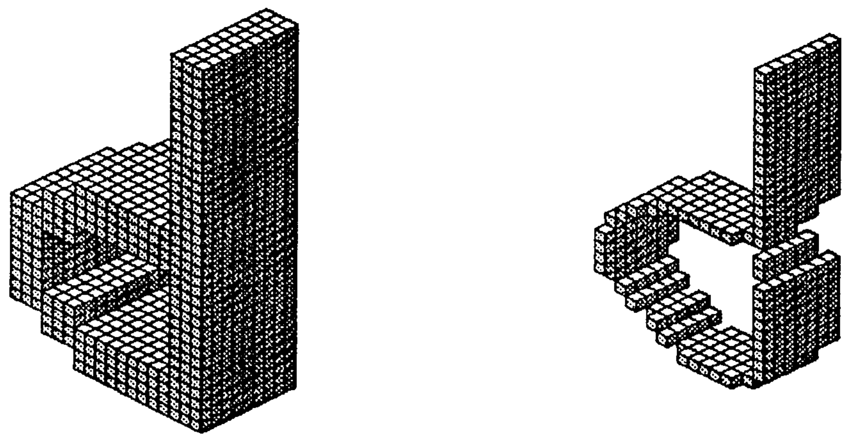


Figure 3.3: A letter "d" and its skeleton where each small cube is a black point in the image.

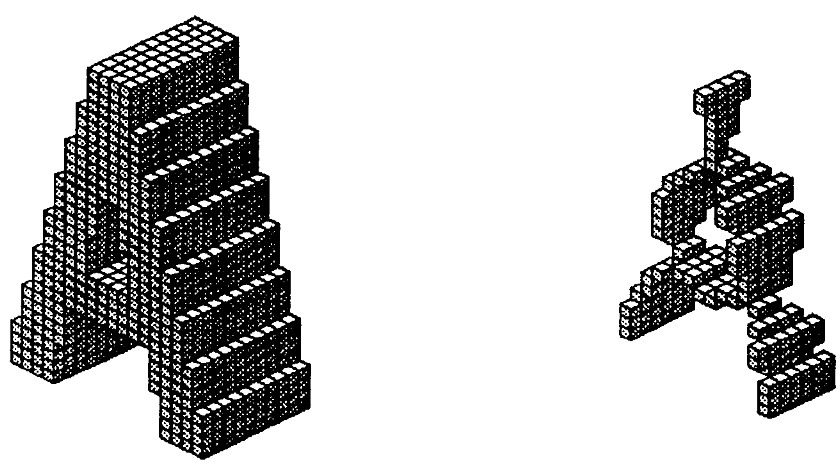


Figure 3.4: A letter "A" and its skeleton where each small cube is a black point in the image.

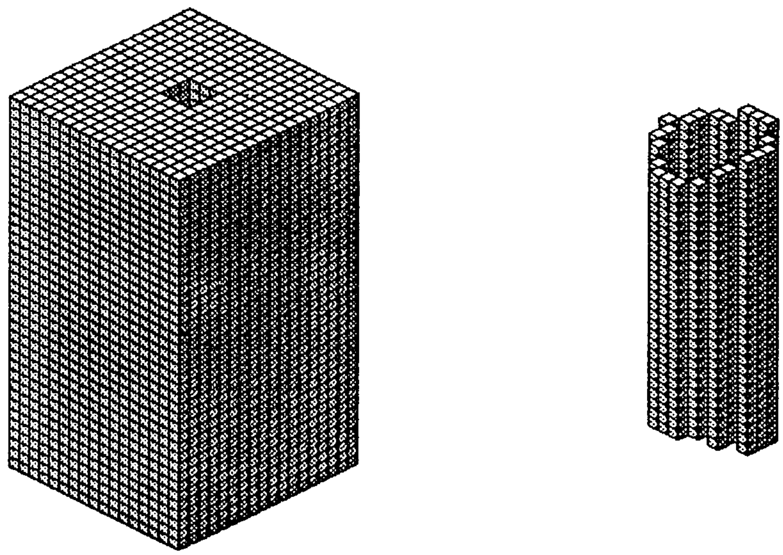


Figure 3.5: An orthogonal tube and its skeleton where each small cube is a black point in the image.

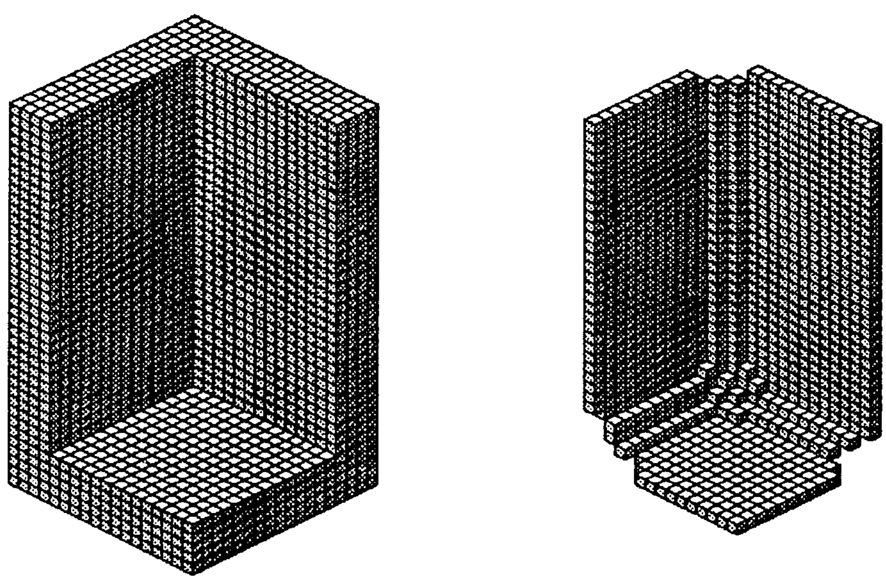


Figure 3.6: An orthogonal corner and its skeleton where each small cube is a black point in the image.

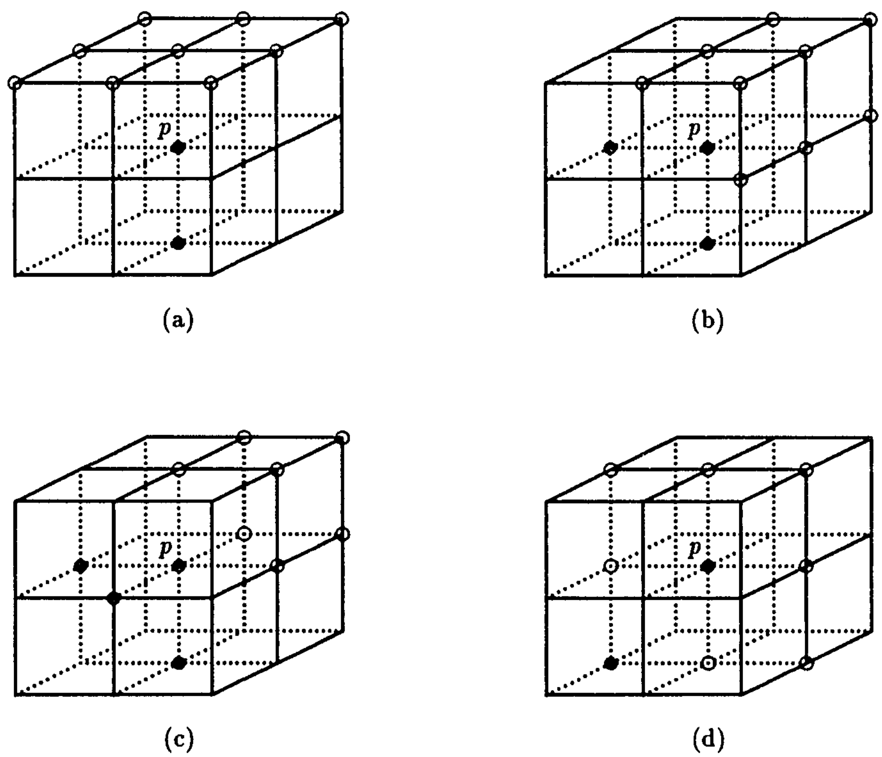


Figure 3.7: Four basic configurations of our parallel thinning algorithm where an unmarked point is a "don't-care" point which can be either black or white. For (d), p must be simple.

Rule 3.3.1 *Let p be a black point and T be an element of $\mathcal{O}_l$. The following are three conditions:*

1. *$N(p)$ matches T;*

2. *if T is isomorphic to a template core shown in Figure 3.7(a–c), all following conditions must be satisfied:*

   (a) *if p is a north border point in T, then $s(s(p)) = 1$;*

   (b) *if p is an east border point in T, then $w(w(p)) = 1$;*

   (c) *if p is an up border point in T, then $d(d(p)) = 1$;*

3. *if T is isomorphic to the template core shown in Figure 3.7.(d), all the following conditions must be satisfied:*

   (a) *p is a simple point in $N(p)$;*

   (b) *if $q = su(p)$ is a 1 in T, then at least one of the points, $s(s(p))$, $s(q)$, $su(q)$, $u(q)$ and $u(u(p))$ is a 1;*

   (c) *if $q = sd(p)$ is a 1 in T, then at least one of the points, $s(s(p))$, $s(q)$, $sd(q)$, $d(q)$ and $d(d(p))$ is a 1;*

   (d) *if $q = se(p)$ is a 1 in T, then at least one of the points, $s(s(p))$, $s(q)$, $se(q)$, $e(q)$ and $e(e(p))$ is a 1;*

   (e) *if $q = sw(p)$ is a 1 in T, then at least one of the points, $s(s(p))$, $s(q)$, $sw(q)$, $w(q)$ and $w(w(p))$ is a 1;*

   (f) *if $q = wu(p)$ is a 1 in T, then at least one of the points, $w(w(p))$, $w(q)$, $w(u(q))$, $u(q)$ and $u(u(p))$ is a 1; and*

   (g) *if $q = wd(p)$ is a 1 in T, then at least one of the points, $w(w(p))$, $w(q)$, $wd(q)$, $d(q)$ and $d(d(p))$ is a 1.*

Rule 3.3.1.(2) was suggested by Holt et al.'s 2D parallel thinning algorithm in [37]. Based on Rule 3.3.1, each element in $\mathcal{O}_l$ is extended to a deleting template. Let *Class A* be the union of all deleting templates of our algorithm that are generated from the configuration shown in Figure 3.7.(a). Define *Classes B, C,* and *D* similarly. In Figures 3.14 to 3.17, some deleting templates of our algorithm are presented. With all rotations and reflections of the four configurations in Figure 3.7.(a–d), there are six templates in Class A, twelve templates in Class B, eight templates in Class C and twelve templates in Class D.

The preserving conditions

As mentioned in previous chapters, geometry preservation is a major concern of thinning algorithms. To preserve geometry, a black point with only one black 26-neighbor should not be deleted. For our medial-line algorithm, we need to preserve some black points which are 26-adjacent to two distinct black points. The following definition is about the preserving conditions of our algorithm. Recall that a black point with only one black neighbor is called a *tail point.*

Definition 3.3.2 *Let p be a black point. Then p is called a* near-tail point *if any of the following conditions hold:*

1. *p is 26-adjacent to exactly two black points which are either $s(p)$ and $e(p)$, or $s(p)$ and $u(p)$ but not both;*
2. *p is 26-adjacent to exactly two black points which are either $n(p)$ and $w(p)$, or $u(p)$ and $w(p)$ but not both;*
3. *p is 26-adjacent to exactly two black points which are either $n(p)$ and $d(p)$, or $e(p)$ and $d(p)$ but not both.*

A black point is called a line-end point *if it is either a tail point or a near-tail point; otherwise it is called a* non-line-end point.

The medial-line algorithm

A black point is deleted by a thinning algorithm if it satisfies at least one deleting template but is not preserved by any preserving condition of the algorithm. Our thinning algorithm 3.3.3 has the traditional structure of parallel thinning algorithms. In each iteration of this algorithm, all non-line-end black points satisfying at least one of the deletion templates are deleted. For each elongated black component, the output of this algorithm is a medial line. This algorithm terminates when no black points can be deleted. Since we assume all input images contain finitely many black points, this algorithm will eventually terminate.

Algorithm 3.3.3

repeat

delete (in parallel) every non-line-end black point which satisfies $\mathcal{O}_l$*;*

until *no point can be deleted;*

The proof of the connectivity preservation of this algorithm will be given later in Chapter 5. For "smooth" 3D objects, the skeletons generated by this algorithm are reasonably good. Different shapes of objects have been tested by this parallel algorithm. Here, only four cases are presented (see Figures 3.8 to 3.11).

3.4 The applications

The section emphasizes the medial-line algorithm which has been recentely applied by Dr. Sonka at the University of Iowa for analyzing 3D medical images. These 3D image data obtained from computer tomography (CT), magnetic resonance (MR), or positron emission tomography (PET) frequently visualize 3D objects consisting of elongated mutually interconnected pieces like vascular trees in brain, heart, lungs, or pulmonary airway trees.

In these applications, skeletons may be used to identify 3D midline of vessels or airways, determine branch points, and identify the tree structure. The resulting tree structure may identify the tree topology across time and can be used in progression/regression studies for quantitative comparisons. More specifically, our thinning algorithm was developed with pulmonary airway tree thinning in lungs and the following steps were developed:

1. Convert an input image P into a *well composed image* P_1 where the definition of such a concept is introduced later.

2. Apply dilation of mathematical morphology in P_1 and generate a "smoother" image P_2.

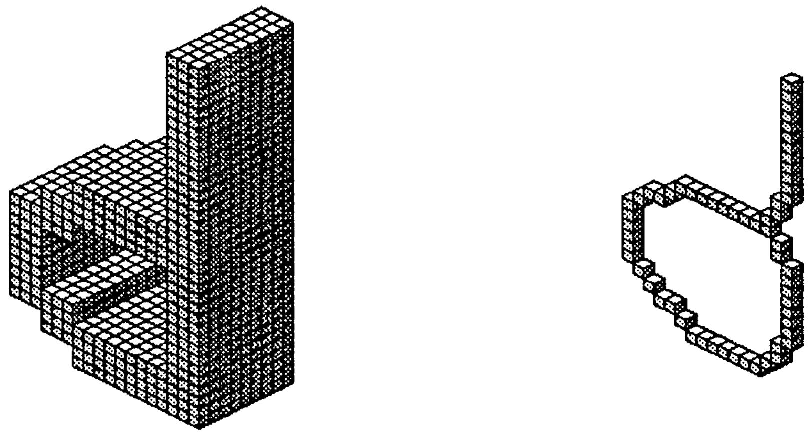


Figure 3.8: A letter "d" and its skeleton where each small cube is a black point in the image.

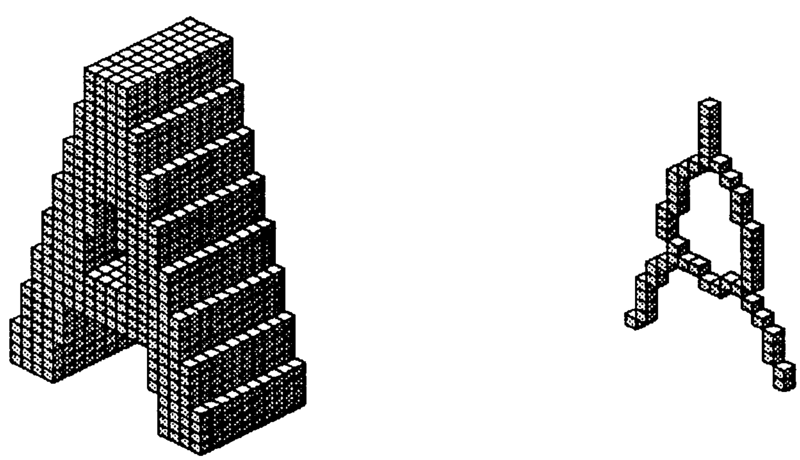


Figure 3.9: A letter "A" and its skeleton where each small cube is a black point in the image.

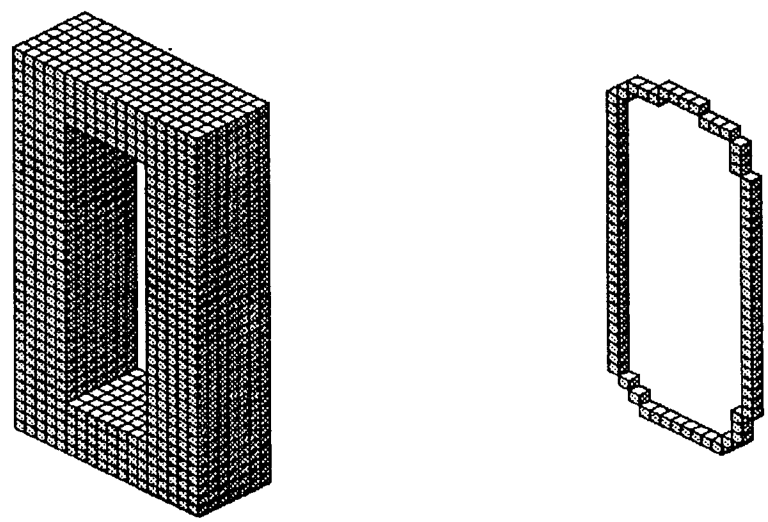


Figure 3.10: A frame and its skeleton where each small cube is a black point in the image.

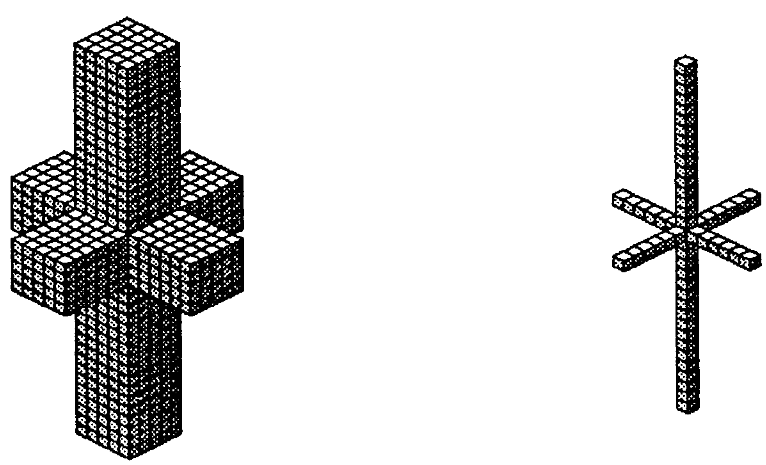


Figure 3.11: A three-way cross and its skeleton where each small cube is a black point in the image.

3. Apply a revised version of fully parallel thinning algorithm in P_2 to obtain the final image P_3.

We now introduce these steps.

Well-composed images

The purpose of thinning is to generate an image in which every black component is of unit width. However, this cannot be guaranteed in real images. For example, consider the 2D object in Figure 2.7.(b). It can be shown that there are no simple points (in the 2D (8,4) sense, see [53]) in this object. Hence, no points can be deleted. However the object does not seem to be a skeleton of unit width. Generally, we don't want this problem to occur in real applications. The concept of 2D *well-composed images* is introduced in [58] to overcome this problem. Its definition for the 3D (26,6) case is as follows.

Definition 3.4.1 *A 3D* $(26,6)$ *image P is called* well composed *if both of the following conditions hold:*

1. *each unit lattice square of P contains at most one black 6-component; and*

2. *each unit lattice cube of P contains at most one black 6-component.*

A black component (or an object) in a well-composed image is called a *well-composed black component* (or a *well-composed object*). A 3D well-composed (26,6) image does not contain any black 26-component which is not a black 6-component. Since every black 26-component and every white 6-component are both 6-connected, the number of special white/black configurations is reduced. In [58], efficient rules were introduced for adding black points in unit lattice squares of 2D (8,4) images of some special configurations. (Analogous rules for 3D (26,6) images can be obtained similarly.) For example, the left object in the following diagram is not well-composed, but the right one is.

1		
	1	
		1
		1
	1	
1	1	1
1	1	1

1		
1	1	
	1	1
		1
	1	1
1	1	1
1	1	1

Note that all computer generated 3D objects shown in Figure 3.3–3.6, and in Figure 3.8–3.11 are well-composed. The two real 3D medical objects shown in Figure 3.12.(a) and Figure 3.13.(a) are also well-composed. However, their skeletons shown in Figure 3.12.(b) and Figure 3.13.(b) are not well-composed.

Dilation

Consider the 2D black component on the left of the following diagram. There is a two-point white component in the bottom and a small branch at the up-left corner; the middle concavity is a "valley".

1			
	1	1	
1	1	1	1
1	1	1	1
			1
1	1	1	1
1	1	1	1
1	1		1
1	1		1
1	1	1	1
1	1	1	1

		2					
	2	2	2	2			
2	2	1	2	2	2		
	2	2	1	1	2	2	
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
	2	2	2	2	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	2	1	2	2
2	2	1	1	2	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
	2	2	2	2	2	2	
		2	2	2	2		

For the 3D case, the middle part of a donut is a "tunnel"; the concept of a "valley" is analogous to the 2D case. We can use dilation of mathematical morphology to fill small valleys and small holes and, hence, to obtain a "smoother" image. In our application, the input images are dilated in two or three iterations using a $3 \times 3 \times 3$ structure element. The origin of the structure element is its central point, all 6-neighbors of the central point are black. For example, suppose we are using a 2D structure element with the origin being its central point and all 4-neighbors of the central point being black points. After two parallel applications of this structure element, the above original 2D object is changed to the component on the right of the above diagram where each point marked 2 is a black point dilated in the two applications.

The revised parallel thinning algorithm

For the purpose of generating "better" skeletons in 3D images, we revise the structure of Algorithm 3.3.3 and generate the following Algorithm 3.4.2. In each iteration of the following algorithm, we first mark every black point which is 26-adjacent to a white point, then apply Algorithm 3.3.3 to the set of all marked points. This algorithm terminates when no black point can be deleted. Since we assume that all input images contain finitely many black points, this algorithm will eventually terminate. In our experiments, the revised algorithm better preserves geometry properties than Algorithm 3.3.3, i.e., a skeleton generated by the revised algorithm has a structure that looks closer to the original black component.

Algorithm 3.4.2

```
repeat
    mark every black point which is 26-adjacent to a white point;
    repeat
        delete (in parallel) every non-line-end marked black point which satisfies O_l;
    until no point can be deleted;
until no marked point can be deleted;
```

Two examples are presented which show that the revised algorithm works reasonably well (see

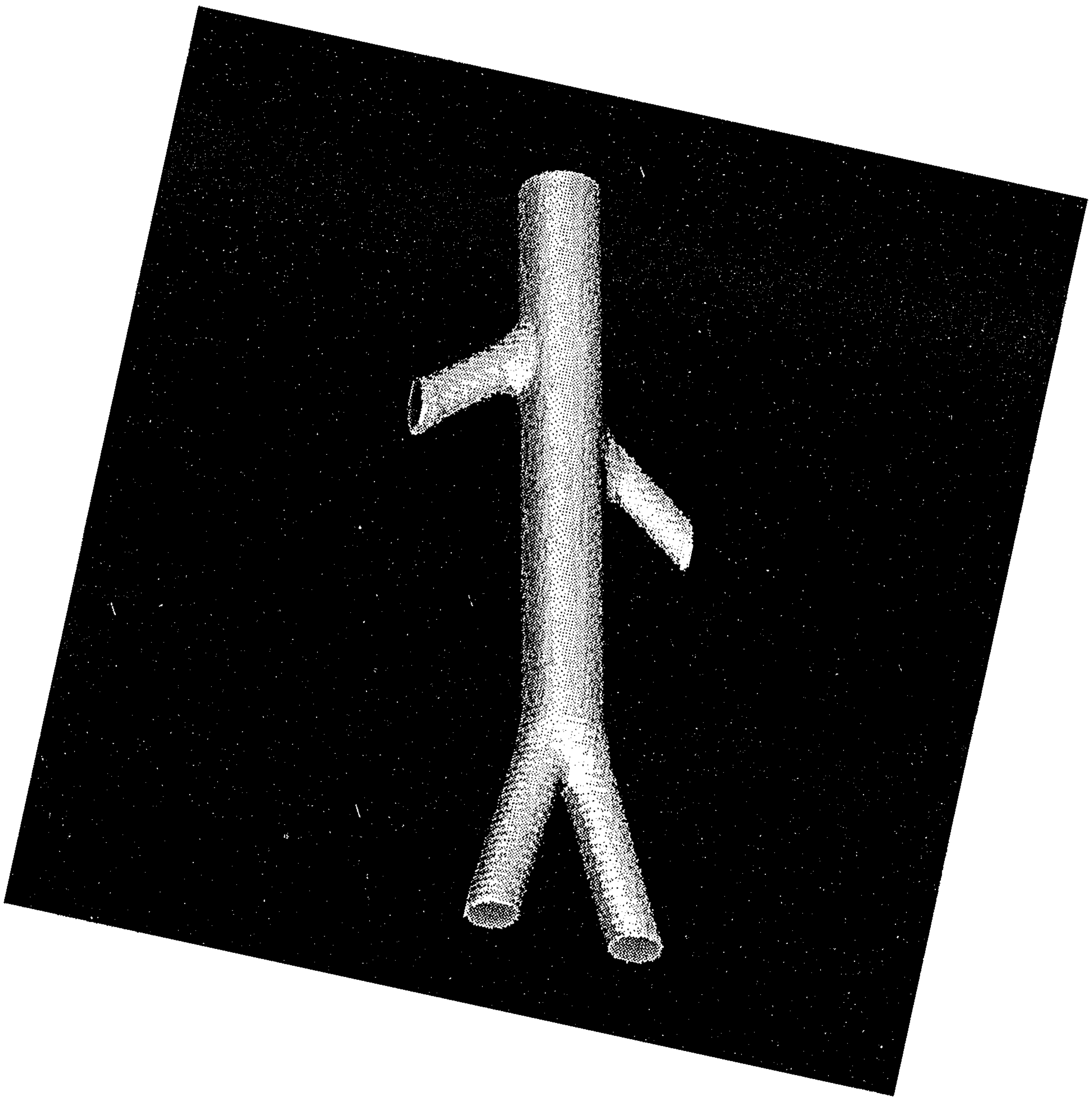
Figures 3.8 to 3.11). Each object in Figure 3.8 to 3.11 is connected. They may not seem to be so because the resolution of the print-out is not high enough.

3.5 Discussion

The proofs that our algorithms preserve connectivity are given in Chapter 5. The revised medial-line algorithm is somewhat similar to the erosion of mathematical morphology. That is, we "erode" the image A by a deleting template B with the origin being the testing point of the template. However, unlike the ordinary erosion, a black point is deleted when it should be eroded in the sense of the ordinary erosion and is a non-line-end point satisfying at least one of the deleting templates.

3D 6-subiteration parallel thinning algorithms have been proposed in [27, 77, 124, 125]. All of them need six applications in each iteration (where the deletion is applied first to all north border points, then on all east border points, and so on). Our medial-line algorithm is the first connectivity preserving fully parallel thinning algorithm for generating skeletons like medial lines. Our medial-face algorithm is also the first fully parallel thinning algorithm for generating skeletons like medial faces. A fully parallel thinning algorithm on an appropriate mesh computer system needs only one application in each iteration.

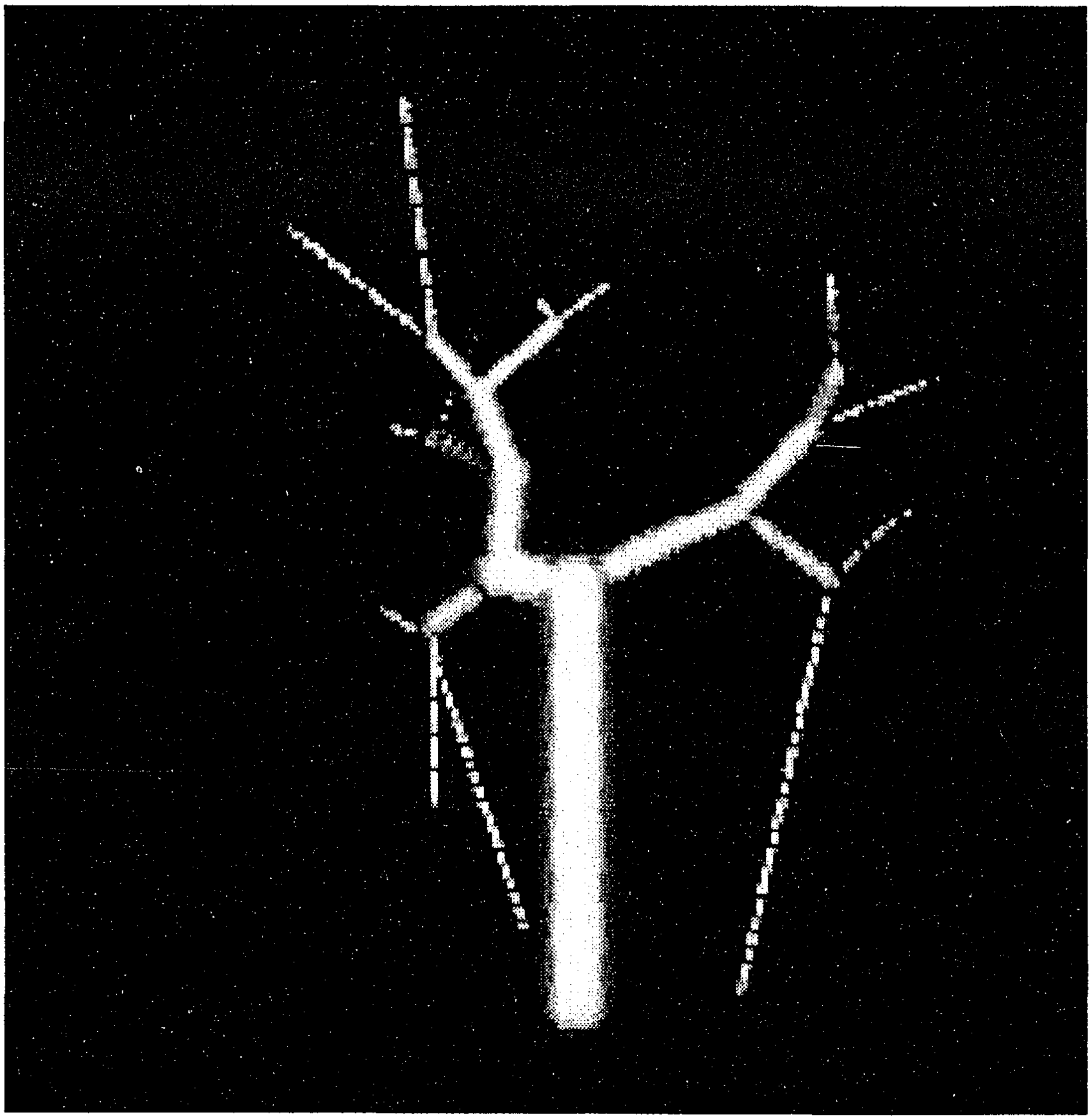
Intuitively, it is not difficult to see that for a set of randomly picked 3D input images, a fully parallel thinning algorithm will generate the output images "faster" than any 6-subiteration, or any 2- or 4-subfield parallel thinning algorithm. Since both of our algorithms, Algorithm 3.2.3 and 3.3.3 are fully parallel, we feel confident to claim that our algorithms are "faster" than all existing medial-face and medial-line connectivity preserving thinning algorithms, respectively.



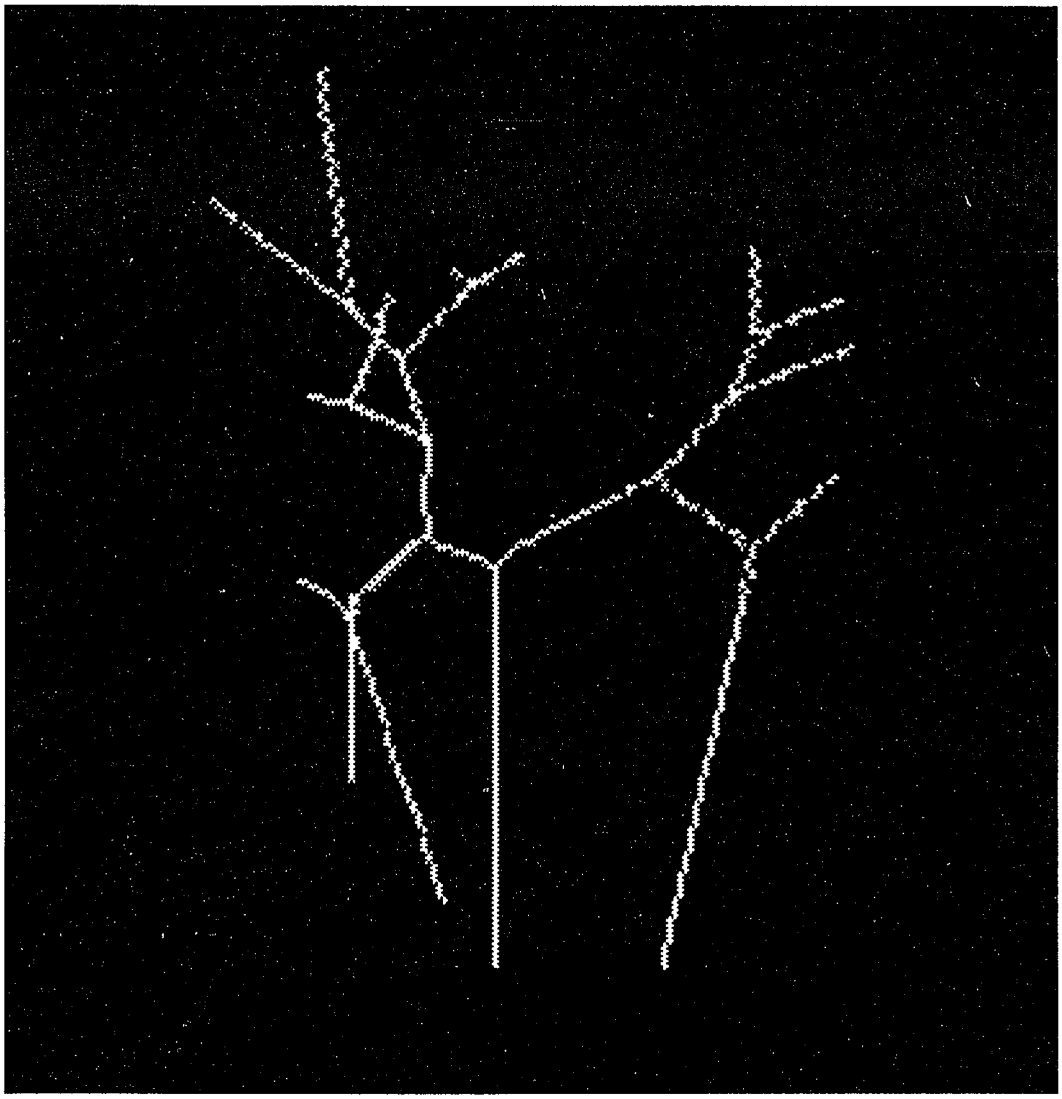
(a)

(b)

Figure 3.12: A 3D image containing a stick with four branches in (a) and its skeleton in (b).



(a)



(b)

Figure 3.13: A 3D image containing a tree-structure object in (a) and its skeleton in (b).

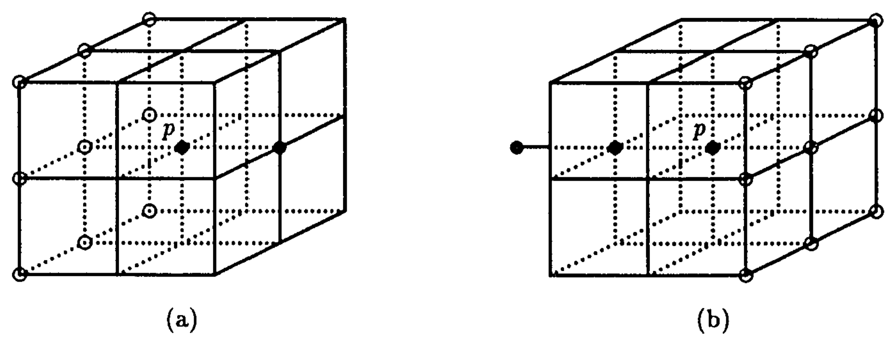


Figure 3.14: There is a Class A template in (a) for deleting west border points, and there is another Class A template in (b) for deleting east border points.

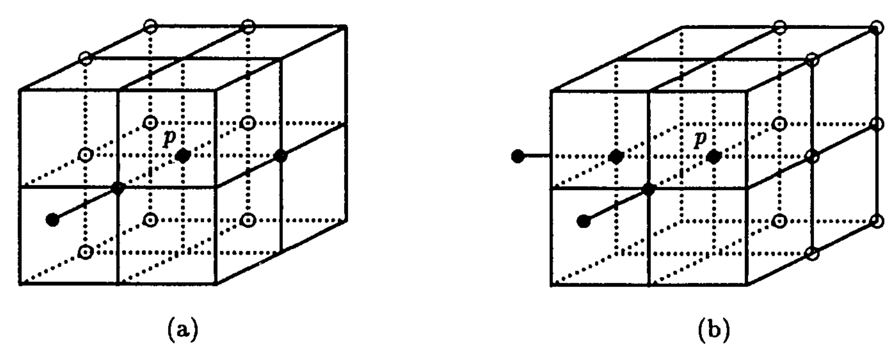
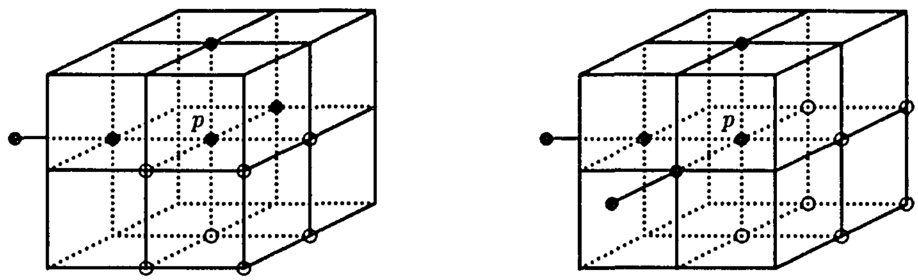
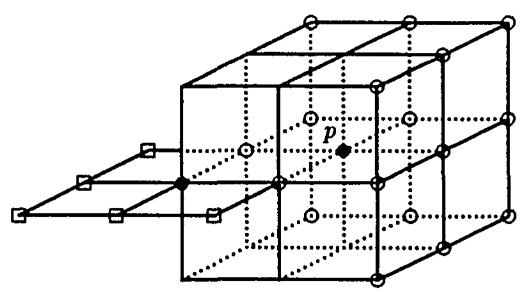


Figure 3.15: There is a Class B template in (a) for deleting north-west border points, and there is another Class B template in (b) for deleting north-east border points.



(a) (b)

Figure 3.16: There is a Class C template in (a) for deleting south-east-down border points, and there is another Class C template in (b) for deleting north-east-down border points.



(a) (b)

Figure 3.17: There is a Class D template in (a) for deleting south-west border points, and there is another Class D template in (b) for deleting north-east border points where at least one point marked □ is a black point.

Chapter 4

3D connectivity preservation tests

Many parallel thinning algorithms (eg., [30, 37, 118] for the 2D case, and [124, 125] for the 3D case) have been proposed and proved to preserve connectivity. Generally, it is not very easy to prove that a parallel thinning algorithm is connectivity preserving. To give such a proof, one needs to test whether the algorithm is connectivity preserving for all possible images. Since there are infinitely many possible input images, it could be difficult to perform such a test. The purpose of the chapter is to solve this problem by finding sufficient conditions such that to show a parallel thinning algorithm preserves connectivity in 3D images, we only need to check a finitely many configurations.

4.1 Main Theorems

Ronse in [95] proposed sufficient conditions for parallel thinning algorithms of 2D (8,4) and (4,8) images to be connectivity preserving. Ronse's results can be described as follows.

Theorem 4.1.1 *Let $\mathcal{A}$ be a parallel thinning algorithm for 2D (β,ω) images where $(\beta,\omega) = (8,4)$ or $(4,8)$. Then $\mathcal{A}$ is connectivity preserving if, for all (β,ω) images P, all of the following conditions hold when $\mathcal{A}$ is applied to P:*

1. *$\mathcal{A}$ only deletes simple points of P;*

2. *for any two ω-adjacent black points, p and q, of P that are deleted by $\mathcal{A}$, $\{p,q\}$ is simple; and*

3. *for the* $(8,4)$ *case,* $\mathcal{A}$ *never deletes any black component of* P *contained in a unit lattice square.* □

By Ronse's results, for determining whether a parallel thinning algorithm preserves connectivity on 2D images we only need to check a finite number of configurations (since only the configurations of a unit lattice square need to be checked, see [94]). Rosenfeld in [99] showed sufficient conditions for parallel thinning algorithms to preserve connectivity on 2D (8,4) images. Hall in [34] extended Rosenfeld's results to more general results on 2D (8,4) images. Hall's results are equivalent to Ronse's results for the 2D (8,4) case. For 3D images, the same topic was studied by Hall in [32] for the (26,6) case, and completely solved by Ma in [65, 66] for all 3D (26,6), (18,6), (6,26) and (6,18) cases. The results in [65, 66] can be stated as follows.

Theorem 4.1.2 *Let* $\mathcal{A}$ *be a parallel thinning algorithm for 3D* $(26,6)$ *images. Then* $\mathcal{A}$ *is connectivity preserving if, for all* $(26,6)$ *images* P*, both of the following conditions hold when* $\mathcal{A}$ *is applied to* P*:*

1. *Every set of black points of* P *that is contained in a unit lattice square and is deleted by* $\mathcal{A}$ *is a simple set of* P*.*

2. $\mathcal{A}$ *does not delete any black component of* P *that is contained in a unit lattice cube.*

Theorem 4.1.3 *Let* $\mathcal{A}$ *be a parallel thinning algorithm for 3D* $(18,6)$ *images. Then* $\mathcal{A}$ *is connectivity preserving if, for all* $(18,6)$ *images* P*, all of the following conditions hold when* $\mathcal{A}$ *is applied to* P*:*

1. *Every set of black points of* P *that is contained in a unit lattice square and is deleted by* $\mathcal{A}$ *is a simple set of* P*.*

2. *Every set of black points of* P *which lies in a unit lattice cube and meets at most three of the four pairs diametrically opposite corners of the cube, and which is deleted by* $\mathcal{A}$*, is a simple set of* P*.*

3. $\mathcal{A}$ *never deletes any black component consisting of four points of* P *where every pair of them are adjacent.*

Theorem 4.1.4 *Let* $\mathcal{A}$ *be a parallel thinning algorithm for 3D* $(6,26)$ *images. Then* $\mathcal{A}$ *is connectivity preserving if, for all* $(6,26)$ *images* P*, both of the following conditions hold when* $\mathcal{A}$ *is applied to* P*:*

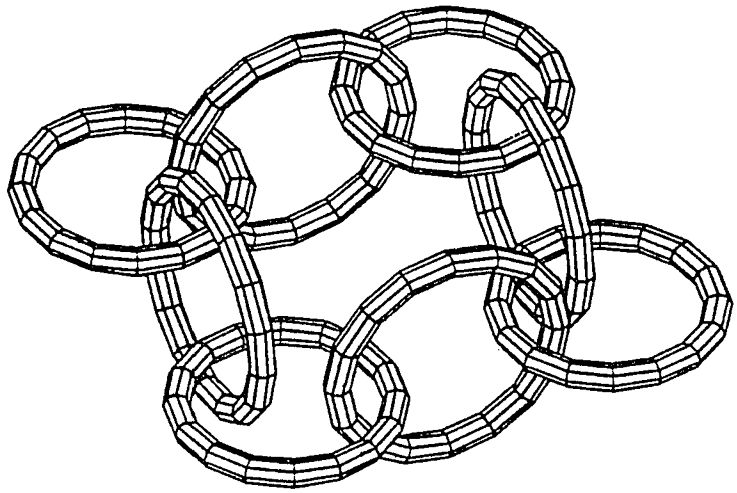


Figure 4.1: Eight interlinked donuts in a 3D image. There are only eight "tunnels" in this image.

1. *Every set of black points of P that contains at most two points and is deleted by $\mathcal{A}$ is a simple set of P.*

2. *Every set of black points of P that is a subset of corners of a $(\sqrt{2}, \sqrt{2}, \sqrt{2})$ triangle and is deleted by $\mathcal{A}$ is a simple set of P.*

Theorem 4.1.5 *Let $\mathcal{A}$ be a parallel thinning algorithm for 3D $(6, 18)$ images. Then $\mathcal{A}$ is connectivity preserving if, for all $(6, 18)$ images P, both of the following conditions hold when $\mathcal{A}$ is applied to P:*

1. *Every set of black points of P that contains at most three points and is deleted by $\mathcal{A}$ is a simple set of P.*

2. *Every set of black points of P that is a subset of corners of a $1 \times \sqrt{2}$ parallelogram and is deleted by $\mathcal{A}$ is a simple set of P.*

4.2 Tunnels and Euler Characteristics

The theory of 3D thinning is much more complicated than that of 2D thinning. This is partly because "tunnels" occur in 3D images (see Figure 4.1). As mentioned above, we will not define "tunnels" in 3D images. Alternatively, we show how to compute $\chi(P)$ of a 3D (β, ω) image P that is equal to the number of black components of P plus the number of cavities of P minus the number of "tunnels" of P. The concept of the continuous analog of P, denoted $C(P)$, was introduced in Chapter 2. It is shown in [52] that $\chi(P) = \chi(C(P))$ where $\chi(C(P)) =$ (no. of black points in $C(P)$) $-$ (no. of black edges in $C(P)$) $+$ (no. of black triangles in $C(P)$) $-$ (no. of black tetrahedra in $C(P)$).

Let K be a unit lattice cube of a 3D image P. We now introduce another approach for computing $\chi(P)$ which assigns a value to K according to its black/white configuration and the adjacency pair of P. Such a value is denoted by $\chi(P;K)$ and can be obtained as follows (see [52, Section 7]). Let K_0 be the set of all corners of K, let K_1 be the set of all edges of K and let K_2 be the set of all six faces of K. Define

$$\chi(P;K) = \chi(P \cap K) - \chi(C(P) \cap K_2)/2 - \chi(C(P) \cap K_1)/4 - \chi(C(P) \cap K_0)/8$$

If all eight corners of K are white points, then $\chi(P;K) = 0$. Note that $\chi(P;K)$ can be computed independently from the black and white configuration out of K. The values of $\chi(P;K)$ for different kinds of 3D images are given in Table 4.1. Then $\chi(P)$ is just the sum of $\chi(P;K)$ for all K of P containing at least one black point. We write this precisely as follows. Let $\mathcal{K}(P)$ be the set of all unit lattice cubes of a 3D image P each of that contains at least one black point of P. Then the following equation holds:

$$\chi(P) = \textstyle\sum_{K \in \mathcal{K}(P)} \chi(P;K)$$

It is not difficult to find $\#_B(P)$ and $\#_C(P)$. Thus, after $\chi(P)$ is obtained, $\#_T(P)$ can be computed accordingly.

Proposition 4.2.1 *Let P_1, P_1', P_2 and P_2' be (β, ω) images where p is the* key point *in each of them. Suppose all of the following hold:*

1. *P_1 and P_1' differ at, and only at p (i.e., p is black in one image, and white in the other);*
2. *P_2 and P_2' differ at, and only at p; and*
3. *the configuration of $N(p)$ is the same in P_1 and P_2.*

Then $\chi(P_1) = \chi(P_1')$ if and only if $\chi(P_2) = \chi(P_2')$.

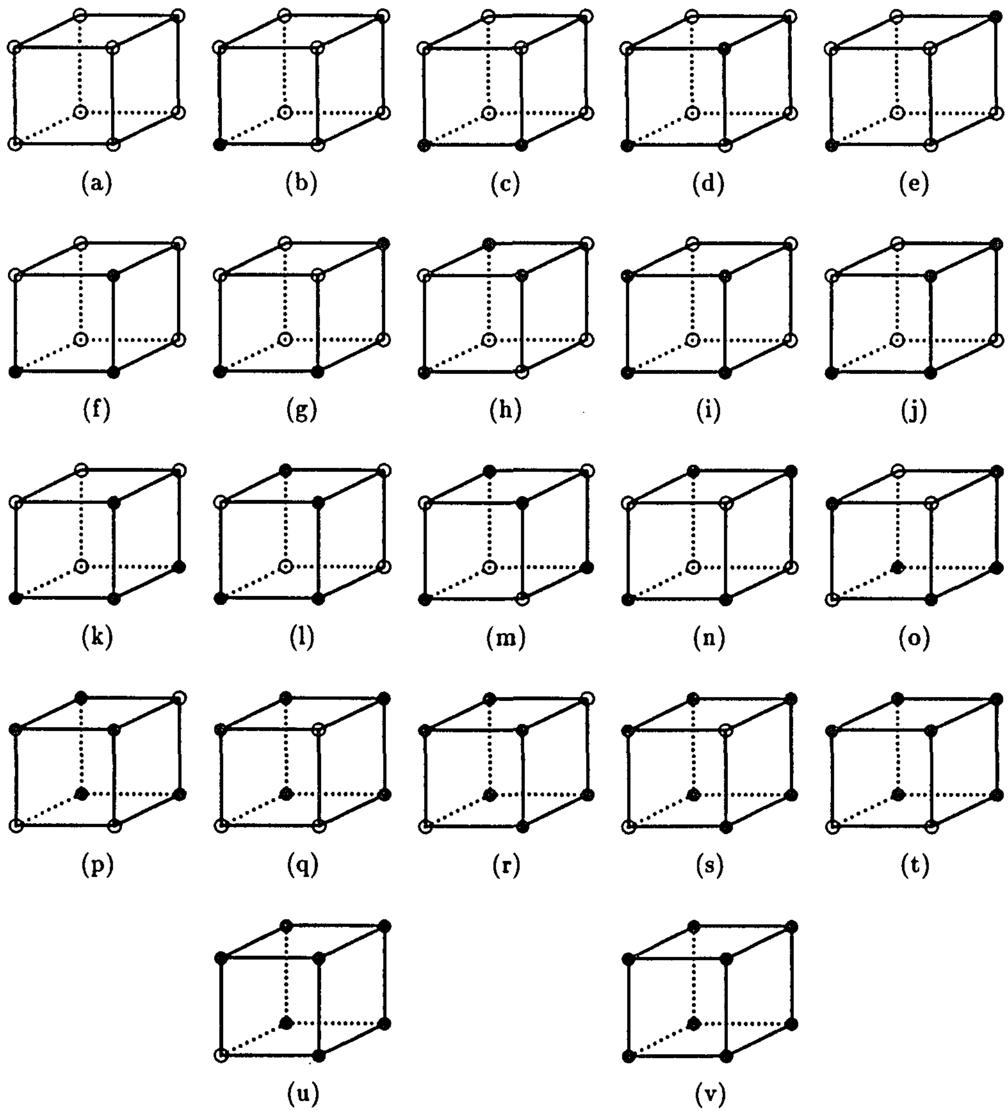


Figure 4.2: The twenty-two different configurations (up to their rotations and reflections) of a unit lattice cube.

Proof. Let $\mathcal{K}(P)$ be the set of all unit lattice cubes each of which contains some black points of a (β,ω) image P. Let $\mathcal{K}_p(P) \subseteq \mathcal{K}(P)$ be the set of all unit lattice cubes in $\mathcal{K}(P)$ each of that contains p (p could be either black or white). Let K be a unit lattice cube in P. If K contains some black points, then K belongs to either $\mathcal{K}_p(P)$ or $\mathcal{K}(P)-\mathcal{K}_p(P)$ but not both. Since the Euler characteristic of an image can be computed by checking all its unit lattice cubes, we have

$$\begin{aligned}
\chi(P_1) &= \textstyle\sum_{K\in\mathcal{K}_p(P_1)}\chi(P_1;K)+\sum_{K\in\mathcal{K}(P_1)-\mathcal{K}_p(P_1)}\chi(P_1;K)\\
\chi(P_1') &= \textstyle\sum_{K\in\mathcal{K}_p(P_1')}\chi(P_1';K)+\sum_{K\in\mathcal{K}(P_1')-\mathcal{K}_p(P_1')}\chi(P_1';K)\\
\chi(P_2) &= \textstyle\sum_{K\in\mathcal{K}_p(P_2)}\chi(P_2;K)+\sum_{K\in\mathcal{K}(P_2)-\mathcal{K}_p(P_2)}\chi(P_2;K)\\
\chi(P_2') &= \textstyle\sum_{K\in\mathcal{K}_p(P_2')}\chi(P_2';K)+\sum_{K\in\mathcal{K}(P_2')-\mathcal{K}_p(P_2')}\chi(P_2';K)
\end{aligned}$$

By (1–2), for P_1 and P_1', and for P_2 and P_2', the only difference is p, that is, the black and white configurations of all unit lattice cubes which does not contain p are the same in P_1 and P_1', and in P_2 and P_2' . Thus,

$$\begin{aligned}
\textstyle\sum_{K\in\mathcal{K}(P_1)-\mathcal{K}_p(P_1)}\chi(P_1;K) &= \textstyle\sum_{K\in\mathcal{K}(P_1')-\mathcal{K}_p(P_1')}\chi(P_1';K)\\
\textstyle\sum_{K\in\mathcal{K}(P_2)-\mathcal{K}_p(P_2)}\chi(P_2;K) &= \textstyle\sum_{K\in\mathcal{K}(P_2')-\mathcal{K}_p(P_2')}\chi(P_2';K)
\end{aligned}$$

Since every unit lattice cube containing p is in $N(p)$, by (3), the black and white configuration of $N(p)$ is the same in P_1 and P_2. By (1–3), the black and white configuration of $N(p)$ is the same in P_1' and P_2'. Thus,

$$\begin{aligned}
\textstyle\sum_{K\in\mathcal{K}_p(P_1)}\chi(P_1;K) &= \textstyle\sum_{K\in\mathcal{K}_p(P_2)}\chi(P_2;K)\\
\textstyle\sum_{K\in\mathcal{K}_p(P_1')}\chi(P_1';K) &= \textstyle\sum_{K\in\mathcal{K}_p(P_2')}\chi(P_2';K)
\end{aligned}$$

Hence, $\chi(P_1)=\chi(P_1')$ if and only if $\chi(P_2)=\chi(P_2')$. □

Let p, q be two points of a 3D image. Let $K(p,q)=N(p)\cap N(q)$. If p is diametrically adjacent to q, then $K(p,q)$ is a single unit lattice cube containing $\{p,q\}$; if p is diagonally adjacent to q, then $K(p,q)$ is the union of two unit lattice cubes sharing a common unit lattice square which contains $\{p,q\}$; if p is 6-adjacent to q, then $K(p,q)$ is the union of four unit lattice cubes sharing a common unit edge with endpoints p and q. If p is not 26-adjacent to q, then $K(p,q)=\emptyset$; if $p=q$, then $K(p,q)=N(p)$. By Proposition 4.2.1, we have the following corollary.

Type	$\#_B(P \cap K)$				$\#_T(P \cap K)$				$\chi(P \cap K)$				$\chi(P;K)$			
in Figure 4.2	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
a	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	1	1	1	1	0	0	0	0	1	1	1	1	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
c	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
d	1	1	2	2	0	0	0	0	1	1	2	2	$-\frac{1}{4}$	$-\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
e	1	2	2	2	0	0	0	0	1	2	2	2	$-\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
f	1	1	1	1	0	0	0	0	1	1	1	1	$-\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$
g	1	1	2	2	0	0	0	0	1	1	2	2	$-\frac{3}{8}$	$-\frac{3}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
h	1	1	3	3	0	0	0	0	1	1	3	3	$-\frac{1}{8}$	$-\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$
i	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
j	1	1	1	1	0	0	0	0	1	1	1	1	$-\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$
k	1	1	1	1	0	0	0	0	1	1	1	1	$-\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$
l	1	1	2	2	0	0	0	0	1	1	2	2	0	0	0	0
m	1	1	4	4	0	0	0	0	1	1	4	4	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
n	1	1	2	2	0	0	0	0	1	1	2	2	0	0	0	0
o	1	1	2	2	0	0	0	0	1	1	2	2	$\frac{3}{8}$	$\frac{3}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$
p	1	1	1	1	0	0	0	0	1	1	1	1	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{3}{8}$	$-\frac{3}{8}$
q	1	1	1	1	0	0	0	0	1	1	1	1	$-\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$
r	1	1	1	1	0	0	1	0	1	1	0	1	$\frac{1}{4}$	$\frac{1}{4}$	$-\frac{3}{4}$	$\frac{1}{4}$
s	1	1	1	1	0	0	0	0	1	1	1	1	$\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$
t	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
u	1	1	1	1	0	0	0	0	1	1	1	1	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{6}$	$\frac{1}{8}$
v	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

Table 4.1: The Euler characteristics computing table of a unit lattice cube K on an image P where A = (26,6), B = (18,6), C = (6,26) and D = (6,18). $\#_C(P \cap K) = 0$ for all possible types of K.

Corollary 4.2.2 *Let p, q be two 26-adjacent black points of a 3D image $P = (\mathbf{Z}^3, \beta, \omega, B)$. Then*

1. $\chi(P) = \chi(P - \{p\})$ *if and only if* $\chi(P \cap N(p)) = \chi(P \cap (N(p) - \{p\}))$;

2. $\chi(P \cap N(p)) = \chi(P \cap (N(p) - \{q\}))$ *if and only if* $\chi(P \cap K(p,q)) = \chi(P \cap (K(p,q) - \{q\}))$; *and*

3. $\chi(P \cap (N(p) - \{p\})) = \chi(P \cap (N(p) - \{p,q\}))$ *if and only if* $\chi(P \cap (K(p,q) - \{p\})) = \chi(P \cap (K(p,q) - \{p,q\}))$.

Proof. Apply Proposition 4.2.1 on (1) where p is the key point, and on (2–3) where q is the key point. □

Let K be a unit lattice cube of P. Then $P \cap K$ contains at most eight black points which are embedded in a unit lattice cube. Thus, it is not difficult to compute $\chi(P \cap K)$ as described above. Since K contains only eight points, $\#_B(P \cap K)$ can be obtained easily. Clearly, K does not contain any cavity, that is, $\#_C(P \cap K) = 0$. It then implies that $\#_T(P \cap K)$ can be computed according to Definition 2.3.1, that is,

$$\#_T(P \cap K) = \#_B(P \cap K) - \chi(P \cap K)$$

Since there are only twenty-two different configurations of a unit lattice cube K (see Figure 4.2), we can get $\chi(P \cap K)$, $\#_B(P \cap K)$ and $\#_T(P \cap K)$ for each of these configurations (see Table 4.1). Now we introduce the following lemma.

Lemma 4.2.3 *Let P be a 3D $(26,6)$ or $(18,6)$ image. Then all of the following hold.*

1. $\chi(P \cap S) = \#_B(P \cap S)$ *where S is a unit lattice square;*

2. $\chi(P \cap K) = \#_B(P \cap K)$ *where K is a unit lattice cube; and*

3. $\chi(P \cap K_T) = \#_B(P \cap K_T)$ *where K_T is the union of two unit lattice cubes sharing a common unit lattice square.*

Sketched Proof. Since every possible black point is adjacent to the background, there are no cavities in each of the following images, $P \cap S$, $P \cap K$ and $P \cap K_T$. It is not difficult to see that there is no "room" for creating tunnels in each of the three images. Thus, (1–3) holds. □

4.3 The minimality of 3D MNS sets

Clearly every non-simple set of black points contains a subset which is a *minimal non-simple set.* We refer to such a set simply as an *MNS set* which is non-simple but any of its proper subset is simple. (Ronse in [95] call such a set an "MND set" which stands for a "minimal non-deletable set".)

If no MNS set is deleted by a parallel thinning algorithm, then the algorithm is connectivity preserving. It would be helpful if the number of configurations of MNS sets was finite (since we can check all possible configurations of MNS sets). Ronse in [95] showed that for the 2D case, every MNS set is a subset of a unit lattice square. An important 3D analogous result is that every MNS set in 3D (26,6), (18,6), (6,26), and (6,18) images is contained in a unit lattice cube (see [48, 65, 66]). For proving such a result, we need the following proposition which was proved by Kong in [48] for 3D (26,6) and (6,26) cases. The same results can be established for 3D (18,6) and (6,18) cases similarly (see [48]).

Proposition 4.3.1 [48] *Suppose D and $D \cup \{p\}$ are simple sets of black points of a 3D image P where $p \notin D$. Then p is a simple point of $P - D$.* □

By Proposition 4.3.1, it is not difficult to see that an MNS set X has an important property that is a nonempty proper subset Y of X is also an MNS set in the image after $X - Y$ is deleted. We prove this property as follows.

Proposition 4.3.2 *Let X be an MNS set of an image P where X is the union of two disjoint sets, X_1 and X_2, and $X_2 \neq \emptyset$. Then X_2 is an MNS set of $P - X_1$.*

Proof. Let D be a proper subset of X_2 and be ordered in any sequence $\langle d_1, d_2, \ldots, d_k\rangle$ where D consists of k distinct elements and $d_i \neq d_j$ if $i \neq j$. Then since both $X_1 \cup \{d_1, d_2, \ldots, d_{i-1}\}$ and $X_1 \cup \{d_1, d_2, \ldots, d_{i-1}, d_i\}$ are proper subsets of X, for $1 \leq i \leq k$, by Proposition 4.3.1, d_i is a simple point after $X_1 \cup \{d_1, d_2, \ldots, d_{i-1}\}$ is deleted. Thus, $\langle d_1, d_2, \ldots, d_k\rangle$ is a simple sequence after X_1 is deleted which implies that D is simple after X_1 is deleted. Since D is arbitrarily chosen, all proper subsets of X_2 are simple after X_1 is deleted. To show X_2 is an MNS set after X_1 is deleted, we need to show that X_2 is non-simple after X_1 is deleted. Pick any point $p \in X_2$. Then $X_2 - \{p\}$ is a proper subset of X_2 and hence is simple after X_1 is deleted. By Proposition 4.3.1, if X_2 is simple after X_1

is deleted, then p is simple after $X-\{p\}$ is deleted which contradicts the fact that X is an MNS set. □

Then we have the following corollary.

Corollary 4.3.3 *Let X be a set of black points an image P where X is the union of two disjoint sets, X_1 and X_2, and $X_2 \neq \emptyset$. If X_2 is not an MNS set in $P - X_1$, then X is not an MNS set in P.* □

By Proposition 4.3.1, we show the following proposition which is a 3D analog of Ronse's 2D results in [95, Theorem 3.5]. The proof of the following proposition is based on the definition of MNS set.

Proposition 4.3.4 *Every 3D MNS set is contained in a unit lattice cube.*

Proof. Let X be a 3D MNS set. It is enough to show that any two points p, q of X are 26-adjacent. Since X is an MNS set, $X - \{p, q\}$ and $X - \{q\}$ are simple. Consider the image in which $X - \{p, q\}$ is deleted. Plainly, p is simple in that image with $q = 1$ but non-simple with $q = 0$. Then, by Definition 2.3.2, q is in $N(p)$ which implies that p, q are 26-adjacent. □

By Proposition 4.3.4, to find all possible configurations of MNS sets in 3D images, we only need to check all black and white configurations of a unit lattice cube. By checking Figure 4.2, there are only 22 different configurations of a unit lattice cube (up to their rotations and reflections).

4.4 Some 3D properties

This section will show some useful properties which will be used to verify whether a given set of black points is an MNS set in 3D images.

Lemma 4.4.1 *Let X be an MNS set of a 3D image P where p, $q \in X$ are two adjacent black points. Suppose all points in $K(p,q) - X$ are white points. Then X is a component.*

Proof. By Proposition 4.3.4, X is contained in a unit lattice cube in $K(p,q)$. Suppose X is not a component. Then X is adjacent to a black point s where $s \notin X$. Since X is an MNS set, every proper subset of X is simple. Thus, the deletion of $X - \{p,q\}$ cannot disconnect s and $\{p,q\}$ in P (otherwise, $X - \{p,q\}$ is non-simple). WLOG let s be adjacent to p. Since all points in $K(p,q) - X$ are white points, $s \notin K(p,q)$. Then after $X - \{p,q\}$ is deleted, p is adjacent to two black components, $\{q\}$ and the one containing s, in $N(p)$. Hence, by Definition 2.3.2.(1), p is non-simple after $X - \{p,q\}$ is deleted. But since $\{p,q\}$ is an MNS set after $X - \{p,q\}$ is deleted, p is simple after $X - \{p,q\}$ is deleted, a contradiction. Thus, X is a component. □

Corollary 4.4.2 *Let X be an MNS set of a 3D image P. Suppose $D \subseteq X$ is a component of $P - (X - D)$. Then X is a component of P.*

Proof. Suppose X is not a component of P. Then X is adjacent to a black point s where $s \notin X$. But since D is a component after $X - D$ is deleted, s is not connected to D in $P - (X - D)$. Thus, the deletion of $X - D$ disconnects s and D and hence $X - D$ is non-simple which contradicts the fact that X is an MNS set. □

Lemma 4.4.3 *Let p and q be two β- but not 6-adjacent black points of a $(\beta,6)$ image P where p is simple and $\beta = 26$ or 18. Suppose $\#_B(P \cap K(p,q)) = \#_B(P \cap (K(p,q) - \{q\})) = \#_B(P \cap (K(p,q) - \{p\})) = \#_B(P \cap (K(p,q) - \{p,q\}))$. Then p is simple in P after q is deleted.*

Proof. Since P is a (26,6) or (18,6) image, referring to [53, Section 9], we only need to show that p satisfies Definition 2.3.2.(1) and (3) after q is deleted for the following three cases:

Case 1. $\beta = 18$, and p is diagonally adjacent to q.

Case 2. $\beta = 26$, and p is diagonally adjacent to q.

Case 3. $\beta = 26$, and p is diametrically adjacent to q.

Note that for Case 1 and Case 2, $K(p,q)$ consists of two unit lattice cubes sharing a unit lattice square; for Case 3, $K(p,q)$ is a single unit lattice cube. For all three cases, since every possible black point in $K(p,q) - \{p,q\}$ is adjacent to both p and q: $\#_B(P \cap K(p,q)) = \#_B(P \cap (K(p,q) - \{q\})) = \#_B(P \cap (K(p,q) - \{p\})) = \#_B(P \cap (K(p,q) - \{p,q\})) = 1$.

Since p is simple with $q = 1$ and p is adjacent to q, all black neighbors of p are connected to q in $N(p) - K(p,q)$. Thus, all black neighbors of p in $N(p) - K(p,q)$ are connected to the only black component in $K(p,q) - \{p,q\}$. Hence, p is adjacent to only one black component in $N(p) - \{p,q\}$ and p satisfies Definition 2.3.2.(1) after q is deleted.

By Lemma 4.2.3, $\#_B(P \cap K(p,q)) = \#_B(P \cap (K(p,q) - \{q\})) = \#_B(P \cap (K(p,q) - \{p\})) = \#_B(P \cap (K(p,q) - \{p,q\}))$ implies $\chi(P \cap K(p,q)) = \chi(P \cap (K(p,q) - \{q\})) = \chi(P \cap (K(p,q) - \{p\}))$ $= \chi(P \cap (K(p,q) - \{p,q\}))$. Then by Corollary 4.2.2.(2–3), we have $\chi(P \cap N(p)) = \chi(P \cap (N(p) - \{q\}))$, and $\chi(P \cap (N(p) - \{p\})) = \chi(P \cap (N(p) - \{p,q\}))$. But since p is simple, $\chi(P \cap N(p)) = \chi(P \cap (N(p) - \{p\}))$. Thus $\chi(P \cap N(p)) = \chi(P \cap (N(p) - \{q\})) = \chi(P \cap (N(p) - \{p,q\}))$ which implies that p satisfies Definition 2.3.2.(3) after q is deleted. □

It was shown in [53, page 381, Section 9] that a black point p is a simple point of an image $(\mathbf{Z}^3, \beta, \omega, B)$ if and only if it is a simple point of $(\mathbf{Z}^3, \omega, \beta, (\mathbf{Z}^3 - B) \cup \{p\})$. The proofs of Corollary 4.4.4 and 4.4.6 apply this fact to images with infinitely many black points without giving new definitions. That is because we are interested in the number of cavities in every associated image (that is the number of black components in its complement image).

Corollary 4.4.4 *Let p and q be two ω- but not 6-adjacent black points of a 3D $(6, \omega)$ image P where p is simple and $\omega = 26$ or 18. Let P_1 be the image obtained from P by changing all points out of $K(p,q)$ to black points. Suppose $\#_C(P_1) = \#_C(P_1 - \{q\}) = \#_C(P_1 - \{p\}) = \#_C(P_1 - \{p,q\})$. Then p is simple in P after q is deleted.*

Proof. Apply Lemma 4.4.3 and follow a similar argument in complement images of P_1, $P_1 - \{q\}$, $P_1 - \{p\}$ and $P_1 - \{p,q\}$. □

Lemma 4.4.5 *Let p and q be two β- but not 6-adjacent black points of an image on a 3D $(\beta, 6)$ images where p is simple and $\beta = 26$ or 18. Suppose $\#_B(P \cap K(p,q)) = \#_B(P \cap (K(p,q) - \{q\}))$, and $\#_B(P \cap (K(p,q) - \{p\})) \neq \#_B(P \cap (K(p,q) - \{p,q\}))$. Then p is non-simple in P after q is deleted.*

Proof. Following a similar argument as the last paragraph of the proof of Lemma 4.4.3, we have $\chi(P \cap N(p)) = \chi(P \cap (N(p) - \{q\})) = \chi(P \cap (N(p) - \{p\})) \neq \chi(P \cap (N(p) - \{p,q\}))$. Thus, p does not satisfy Definition 2.3.2.(3) after q is deleted and hence p is non-simple after q is deleted. □

Corollary 4.4.6 *Let p and q be two ω- but not 6-adjacent black points of a 3D $(6,\omega)$ image P where p is simple and $\omega = 26$ or 18. Let P_1 be the image obtained from P by changing all points out of $K(p,q)$ to black points. Suppose $\#_C(P_1) \neq \#_C(P_1 - \{q\})$, and $\#_C(P_1 - \{p\}) = \#_C(P_1 - \{p,q\})$. Then p is non-simple in P after q is deleted.*

Proof. Apply Lemma 4.4.5 and follow a similar argument on complement images of P_1, $P_1 - \{q\}$, $P_1 - \{p\}$ and $P_1 - \{p,q\}$. □

Since we are dealing with (26,6), (18,6), (6,26) and (6,18) images, we will discuss the configurations of a unit lattice cube for all of the four cases. We call an MNS set for the case of n-dimensional (β,ω) images an *n-dimensional (β,ω) MNS set.* Let D be a nonempty proper subset of a set X of black points of a image. It is proved in Proposition 4.3.2 and Corollary 4.3.3 that if X is an MNS set, then D is also an MNS set after $X - D$ is deleted, and if D is not an MNS set after $X - D$ is deleted, then X itself is not an MNS set. Even though by Proposition 4.3.4, every 3D MNS set is contained in a unit lattice cube, it is still necessary to find all possible configurations of 3D MNS sets. In the following sections, we first investigate all 3D (26,6) and (18,6) MNS sets that must be components, then investigate all configurations in a unit lattice cube that cannot be 3D (18,6), (6,26) and (6,18) MNS sets, and finally give formal proofs to Theorem 4.1.2–Theorem 4.1.5.

Let X be a set of some black corners of a unit lattice cube K. To simplify the investigation, we call X an *x-type set* if and only if X is isometric to the black point set shown in Figure 4.2.(x) where a point in $K - X$ could be either a black point or a white point; we call X an *$\{x_1, \ldots, x_k\}$-type set* if and only if X is an x-type set for some $x \in \{x_1, \ldots, x_k\}$. An *$x$-type (β,ω) MNS set* is an x-type set which is an (β,ω) MNS set.

4.5 The 3D MNS sets which are components

In 3D (26,6) and (18,6) images, MNS sets of some special configurations must be a component. This section explores all these configurations.

Proposition 4.5.1 *If a 3D $(26,6)$ MNS set contains an $\{\mathrm{e},\mathrm{h}\}$-type subset, then it is a component of any $(26,6)$ image.*

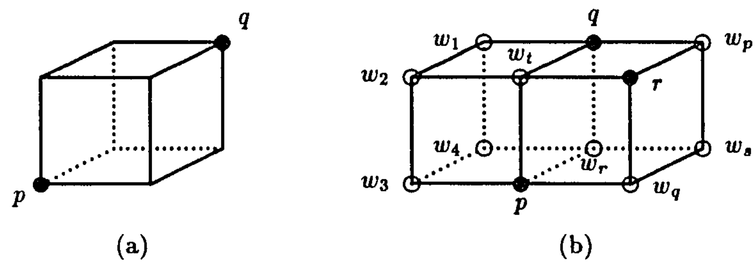


Figure 4.3: Special configurations in (26,6) images — $\{p, q\}$ in (a), and $\{p, q, r\}$ in (b) (up to their rotations and reflections). An unmarked corner point can be either a black point or a white point.

Proof. Let D be an {e, h}-type subset of a (26,6) MNS X. The proof is on the image P where $X - D$ is deleted. By Corollary 4.4.2, to show that X is a component in the original image, it is enough to show that D is a component in P for the following two cases:

Case 1. $D = \{p, q\}$ is an e-type set (see Figure 4.3.(a)).

Case 2. $D = \{p, q, r\}$ is an h-type set (see Figure 4.3.(b)).

By Proposition 4.3.2, D is an MNS set of P. For Case 1, if there exists a black point in $K(p, q) - \{p, q\}$, then $\#_B(P \cap K(p, q)) = \#_B(P \cap (K(p, q) - \{q\})) = \#_B(P \cap (K(p, q) - \{p\})) = \#_B(P \cap (K(p, q) - \{p, q\})) = 1$. But since p is simple when $q = 1$, by Lemma 4.4.3, p is simple after q is deleted. Thus, D is not an MNS set, a contradiction. Hence, all points in $K(p, q) - D$ are white points which by Lemma 4.4.1 implies that D is a component in P.

For Case 2, clearly, while $q = 1$, p is simple with $r = 0$ or 1. If $w_r = 1$ or $w_t = 1$, then $\#_B(P \cap (K(p, q) - \{r\})) = \#_B(P \cap (K(p, q) - \{q, r\})) = \#_B(P \cap (K(p, q) - \{p, r\})) = \#_B(P \cap (K(p, q) - \{p, q, r\})) = 1$. By Lemma 4.4.3, p is simple after $\{q, r\}$ is deleted. Since q is simple after only r is deleted, D can be arranged as a simple sequence $\langle r, q, p\rangle$ and hence is not an MNS set, a contradiction. Hence, $w_r = w_t = 0$.

If any point in $\{w_1, w_2, w_3, w_4\}$ is a black point, then $\#_B(P \cap K(p, q)) = \#_B(P \cap (K(p, q) - \{q\})) = \#_B(P \cap (K(p, q) - \{p\})) = 1 \neq 2 = \#_B(P \cap (K(p, q) - \{p, q\}))$ which by Lemma 4.4.5 implies that p is non-simple after only q is deleted (while $r = 1$). But by Proposition 4.3.2, $\{p, r\}$ is a (26,6) MNS

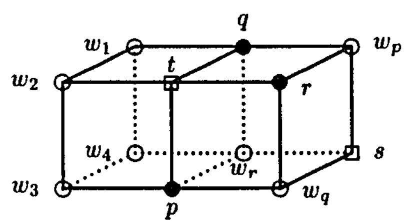


Figure 4.4: Special configurations in (18,6) images — $\{p, q, r, s\}$ for $s = 1$, and $\{p, q, r, t\}$ for $t = 1$ (up to their rotations and reflections).

set after q is deleted. Thus, p is simple after only q is deleted, a contradiction. Hence, $w_1 = w_2 = w_3 = w_4 = 0$.

If any point in $\{w_p, w_q, w_s\}$ is a black point, then $\#_B(P \cap (K(p,q) - \{r\})) = \#_B(P \cap (K(p,q) - \{q,r\})) = \#_B(P \cap (K(p,q) - \{p,r\})) = \#_B(P \cap (K(p,q) - \{p,q,r\})) = 1$. Thus, as above, $\langle r, q, p\rangle$ is a simple sequence and D is not a (26,6) MNS set, a contradiction. Thus, $w_p = w_q = w_s = 0$, and all points in $K(p,q) - D$ are white points which by Lemma 4.4.1 implies that D is a component in P. □

In 3D (18,6) images, some MNS sets of special configurations must be components. All such MNS sets contain three points where every two of them are diagonally adjacent points.

Lemma 4.5.2 *Let $D = \{p, q, r\}$ be a* h*-type MNS set of an* (18,6) *image P (refer to Figure 4.4). Then*

1. $w_p = w_q = w_r = 0$; *and*

2. *if $s = t = 0$, then $w_1 = w_2 = w_3 = w_4 = 0$.*

Proof. Clearly, while $q = 1$, p is simple with $r = 0$ or 1. For (1), suppose $w_r = 1$. Then $\#_B(P \cap (K(p,q) - D)) \geq 1$. Since every point, except w_2, in $K(p,q) - D$ is 18-adjacent to w_r, $\#_B(P \cap (K(p,q) - D)) \leq 2$. Thus, there are only two cases:

Case 1. $\#_B(P \cap (K(p,q) - D)) = 1$.

Case 2. $\#_B(P \cap (K(p,q) - D)) = 2$.

Case 1 implies that $\#_B(P \cap (K(p,q) - \{r\})) = \#_B(P \cap (K(p,q) - \{q,r\})) = \#_B(P \cap (K(p,q) - \{p,r\})) = \#_B(P \cap (K(p,q) - \{p,q,r\})) = 1$. Thus by Lemma 4.4.3, p is simple in P after $\{q,r\}$ is deleted. Since q is simple after only r is deleted, D can be arranged as a simple sequence $\langle r,q,p \rangle$ and hence is not an MNS set, a contradiction.

For Case 2, since $w_r = 1$, the only condition is $w_2 = 1$ and $w_1 = w_3 = w_4 = t = 0$. Then $\#_B(P \cap K(p,q)) = \#_B(P \cap (K(p,q) - \{q\})) = \#_B(P \cap (K(p,q) - \{p\})) = 1 < \#_B(P \cap (K(p,q) - \{p,q\}))$. Hence by Lemma 4.4.5, p is non-simple in P after only q is deleted which implies that D is not an MNS set, a contradiction. Thus, for both Case 1 and Case 2, $w_r = 0$. Now since (r, w_r), (p, w_p) and (q, w_q) are congruent pairs, $w_p = w_q = w_r = 0$ and (1) holds.

Now consider (2), suppose $s = t = 0$. If $w_2 = 1$, or $w_4 = 1$, or $w_1 = w_3 = 1$, then $\#_B(P \cap K(p,q)) = \#_B(P \cap (K(p,q) - \{q\})) = \#_B(P \cap (K(p,q) - \{p\})) = 1 \neq 2 = \#_B(P \cap (K(p,q) - \{p,q\}))$ which, by Lemma 4.4.5, implies that p is non-simple in P after only q is deleted (while $r = 1$), and hence D is not an MNS set, a contradiction. Thus, $w_2 = w_4 = 0$ and either $w_1 = 0$ or $w_3 = 0$.

Suppose $w_3 = 1$. Then $w_1 = 0$. But all points in $K(p,q)) - \{p,q,r,w_3\}$ are white points, p is 18-adjacent to two black components, $\{q\}$ and the one containing w_3, in $N(p) - \{p,r\}$. Thus p is non-simple in P after only r is deleted (while $q = 1$) and hence D is not an MNS set, a contradiction. Thus, $w_3 = 0$. Similarly $w_1 = 0$ for otherwise q is adjacent to two black components in $N(q) - \{q,r\}$.
□

Proposition 4.5.3 *If a 3D* (18,6) *MNS set is a* {k, m}*-type set, then it is a component of any* (18,6) *image.*

Proof. Let $D = \{p,q,r,t\}$ be a k-type set, or $D = \{p,q,r,s\}$ be an m-type set of an (18,6) image P (see Figure 4.4). For both cases, since $\{p,q,r\}$ is a proper subset of D, by Proposition 4.3.2, $\{p,q,r\}$ is still an (18,6) MNS set after the other point in D is deleted. Thus, by Lemma 4.5.2.(1), $w_p = w_q = w_r = 0$.

Suppose $D = \{p,q,r,t\}$ is a k-type set. Clearly, p is simple after only r is deleted. If $s = 1$, then after r is deleted, the deletion of $\{p,q\}$ disconnects s and t, that is, $\#_B(P \cap (K(p,q) - \{r\})) = \#_B(P \cap (K(p,q) - \{q,r\})) = \#_B(P \cap (K(p,q) - \{p,r\})) = 1 < \#_B(P \cap (K(p,q) - \{p,q,r\}))$. Thus, by Lemma 4.4.5, p is non-simple in P after only $\{q,r\}$ is deleted (while $t=1$). This contradicts the fact that D is an MNS set.

Thus, $s=0$. Then $\{p,q,r\}$ is a (18,6) MNS set in P after t is deleted (i.e., $s=t=0$). By Lemma 4.5.2.(2), $w_1=w_2=w_3=w_4=0$. Thus all points in $K(p,q)-D$ are white points which by Lemma 4.4.1 implies that the k-type MNS set D is a component in P.

Now suppose $D=\{p,q,r,s\}$ is an m-type set. By Lemma 4.5.2.(1), since (s,t) is congruent to (r,w_r), $t=0$. Since $\{p,q,r\}$ is an (18,6) MNS set in P after s is deleted (i.e., $s=t=0$), similar as above, $w_1=w_2=w_3=w_4=0$. Hence, all points in $K(p,q)-D$ are white points which then implies that the m-type MNS set D is a component in P. □

4.6 The configurations which are not 3D MNS sets

Lemma 4.6.1 *An* $\{\mathrm{o},\mathrm{q},\mathrm{s},\ldots,\mathrm{v}\}$*-type set is not a 3D* (18,6) *MNS set.*

Proof. Let X be an $\{\mathrm{o},\mathrm{q},\mathrm{s},\ldots,\mathrm{v}\}$-type MNS set of a 3D (18,6) image P. By checking Figure 4.2, X contains a k-type proper subset D. Then by Proposition 4.3.2, D is an (18,6) MNS set after $X-D$ is deleted, and by Proposition 4.5.3, D is a component in P after $X-D$ is deleted. By Corollary 4.4.2, X is a component in P. By checking Figure 4.2, X contains two diametrically opposite points $\{p,q\}$. Since X is a component in P, both $\{p\}$ and $\{q\}$ are one-point components in P after $X-\{p,q\}$ is deleted, that is, p is non-simple in P after $X-\{p,q\}$ is deleted (while $q=1$) which contradicts the fact that X is an (18,6) MNS set in P. □

Now we investigate all possible configurations in 3D (6,26) and (6,18) image that are not MNS sets.

Lemma 4.6.2 *An* $\{\mathrm{f},\mathrm{g}\}$*-type set is not a 3D* (6,26) *MNS set.*

Proof. Suppose $X=\{p,q,r\}$ is an $\{\mathrm{f},\mathrm{g}\}$-type MNS set of a 3D (6,26) image P (see Figure 4.5.(a) and 4.5.(b) respectively). Let P_1 be the image obtained from P by changing all points out of $K(p,q)$ to black points. Then after r is deleted, we have, $\#_C(P_1-\{r\})=\#_C(P_1-\{q,r\})=\#_C(P_1-\{p,r\})=\#_C(P_1-\{p,q,r\})$. Since both p and q are simple in $P-r$, by Corollary 4.4.4, p is simple in $P-\{q,r\}$. Thus, $\langle r,q,p\rangle$ is a simple sequence and X is not an MNS set, a contradiction. □

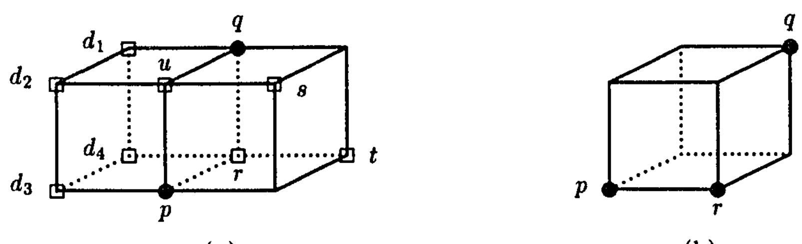


Figure 4.5: Special configurations in (6,26) images — in (a), $\{p,q,r\}$ for $r = 1$ or $\{p,q,s,t\}$ for $s = t = 1$, and in (b), $\{p,q,r\}$ (up to their rotations and reflections). An unmarked corner point can be either a black point or a white point.

Lemma 4.6.3 *An* m*-type set is not a 3D* $(6,26)$ *MNS set.*

Proof. Suppose $X = \{p,q,s,t\}$ is an m-type MNS set of a 3D (6,26) image P (see Figure 4.5.(a)). Let P_1 be the image obtained from P by changing all points out of $K(p,q)$ to black points. Then since $\{s,t\}$ is a white component in $P_1 - \{s,t\}$, $\#_C(P_1 - \{s,t\}) \geq 1$. But since two white points are adjacent if they are 26-adjacent, $\#_C(P_1 - \{s,t\}) \leq 2$. Thus, we consider the following two cases:

Case 1. $\#_C(P_1 - \{s,t\}) = 1$.

Case 2. $\#_C(P_1 - \{s,t\}) = 2$.

For Case 1, since X is a (6,26) MNS set, p is simple in $P - \{s,t\}$ (while $q = 1$) and p is non-simple in $P - \{q,s,t\}$. But since both p and q are 26-adjacent to $\{s,t\}$, the deletion of p or q cannot increase the number of cavities in $P_1 - \{s,t\}$. Thus, $\#_C(P_1 - \{s,t\}) = \#_C(P_1 - \{q,s,t\}) = \#_C(P_1 - \{p,s,t\}) = \#_C(P_1 - \{p,q,s,t\}) = 1$ which by Corollary 4.4.4 implies that p is simple in P after $\{q,s,t\}$ is deleted, a contradiction.

For Case 2, similar as above, p is simple in $P - \{s\}$ and $P - \{q,s\}$. Since $\#_C(P_1 - \{s,t\}) = 2$, at least one point in $\{d_1,d_2,d_3,d_4\}$ is a white point and $r = u = 1$. Then there are two cavities in $P_1 - \{s\}$. Thus $\#_C(P_1 - \{s\}) = 2 \neq 1 = \#_C(P_1 - \{q,s\}) = \#_C(P_1 - \{p,s\}) = \#_C(P_1 - \{p,q,s\})$ which by Corollary 4.4.6 implies that p is non-simple in P after $\{q,s\}$ is deleted, a contradiction.

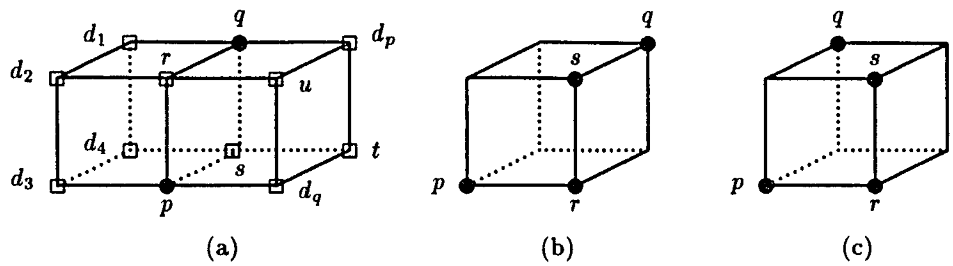


Figure 4.6: Special configurations in (6, 18) images — in (a), $\{p, q, r, s\}$ for $r = s = 1$, or $\{p, q, s, t\}$ for $s = t = 1$, or $\{p, q, t, u\}$ for $t = u = 1$; and in (b–c), $\{p, q, r, s\}$ (up to their rotations and reflections). An unmarked corner point can be either a black point or a white point.

Hence, an m-type set is not a 3D (6,26) MNS set. □

Since an {f, g, m}-type set is not a (6,26) MNS set, by Corollary 4.3.3, a set of corners of a unit lattice cube that contains an {f, g, m}-type subset is not a (6,26) MNS set. So we have the following corollary.

Corollary 4.6.4 *An* {f, g, i, ..., v}*-type set is not a 3D* (6, 26) *MNS set.*

Proof. By Lemma 4.6.2 and 4.6.3, an {f, g, m}-type set is not a (6,26) MNS set. By Corollary 4.3.3, an n-type set is not a (6,26) MNS set since it contains a g-type subset, and an {i, ..., l, o, ..., v}-type set is not a (6,26) MNS set since each of them contains an f-type subset. □

Lemma 4.6.5 *An* {i, j, l}*-type set is not a 3D* (6, 18) *MNS set.*

Proof. Suppose $X = \{p, q, r, s\}$ is an {i, j, l}-type MNS set of a 3D (6,18) image P (see Figure 4.6.(a–c), respectively). Let P_1 be the image obtained from P by changing all points out of $K(p, q)$ to black points. Then since X is a (6,18) MNS set, p is simple in $P - \{r, s\}$ but is non-simple in $P - \{q, r, s\}$. But for all three cases, $\#_C(P_1 - \{r, s\}) = \#_C(P_1 - \{q, r, s\}) = \#_C(P_1 - \{p, r, s\}) = \#_C(P_1 - \{p, q, r, s\}) = 1$. Then by Corollary 4.4.4, p is simple in P after $\{q, r, s\}$ is deleted, a contradiction. Thus, an {i, j, l}-type set is not a 3D (6,18) MNS set. □

Lemma 4.6.6 *A* k*-type set is not a 3D* (6, 18) *MNS set.*

Proof. Suppose $X = \{p,q,s,t\}$ is a k-type MNS set of a 3D (6,18) image P (see Figure 4.6.(a)). Let P_1 be the image obtained from P by changing all points out of $K(p,q)$ to black points. Then since $\{s,t\}$ is a white component in $P_1 - \{s,t\}$, $\#_C(P_1 - \{s,t\}) \geq 1$; since all points, except d_2, in $K(p,q) - X$ are 18-adjacent to $\{s,t\}$ $\#_C(P_1 - \{s,t\}) \leq 2$. Thus, we consider the following two cases:

Case 1. $\#_C(P_1 - \{s,t\}) = 1$.

Case 2. $\#_C(P_1 - \{s,t\}) = 2$.

For Case 1, since X is a (6,18) MNS set, p is simple in $P - \{s,t\}$ (while $q = 1$) and is non-simple in $P - \{q,s,t\}$. But since both p and q are 18-adjacent to $\{s,t\}$, the deletion of p or q cannot increase the number of cavities of $P_1 - \{s,t\}$. Thus, $\#_C(P_1 - \{s,t\}) = \#_C(P_1 - \{q,s,t\}) = \#_C(P_1 - \{p,r,s\}) = \#_C(P_1 - \{p,q,s,t\}) = 1$. Then by Corollary 4.4.4, p is simple in P after $\{q,r,s\}$ is deleted, a contradiction.

For Case 2, since $s = t = 0$ in $P_1 - \{s,t\}$, the only condition is $d_2 = 0$ and $d_1 = d_3 = d_4 = r = 1$. Hence, $\#_C(P_1 - \{s,t\}) = 2 \neq 1 = \#_C(P_1 - \{q,s,t\}) = \#_C(P_1 - \{p,s,t\}) = \#_C(P_1 - \{p,q,s,t\})$. By Corollary 4.4.6, p is non-simple in $P - \{s,t\}$ (while $q = 1$). But since X is a (6,18) MNS set, p is simple in $P - \{s,t\}$, a contradiction. Hence, a k-type set is not a 3D (6,18) MNS set. □

Lemma 4.6.7 *An* m-*type set is not a 3D* (6,18) *MNS set.*

Proof. Suppose $X = \{p,q,t,u\}$ is an m-type MNS set of a 3D (6,18) image P (see Figure 4.6.(a)). Let P_1 be the image obtained from P by changing all points out of $K(p,q)$ to black points. Since $\{t,u\}$ is a white component in $P_1 - \{t,u\}$, $\#_C(P_1 - \{t,u\}) \geq 1$. ; since two white points are adjacent if they are 18-adjacent, $\#_C(P_1 - \{t,u\}) \leq 2$. Thus, we consider the following two cases:

Case 1. $\#_C(P_1 - \{t,u\}) = 1$.

Case 2. $\#_C(P_1 - \{t,u\}) = 2$.

For Case 1, we have $\#_C(P_1 - \{t,u\}) = \#_C(P_1 - \{q,t,u\}) = \#_C(P_1 - \{p,t,u\}) = \#_C(P_1 - \{p,q,t,u\}) = 1$. By a similar argument as in Lemma 4.6.6, X is not an MNS set in P, a contradiction.

For Case 2, if $s = 0$, then to have two white components in $P_1 - \{t,u\}$, it must be $d_2 = 0$ and $d_1 = d_3 = d_4 = r = 1$. But then $\#_C(P_1 - \{t,u\}) = 2 \neq 1 = \#_C(P_1 - \{q,t,u\}) = \#_C(P_1 - \{p,t,u\}) =$

$\#_C(P_1 - \{p,q,t,u\})$ which by a similar argument as in Lemma 4.6.6 implies that X is not an MNS set in P, a contradiction.

Thus, $s = 1$. Since (u,s), (p,d_p), (q,d_q) and (t,r) are congruent pairs, $r = d_p = d_q = 1$. Then since $\#_C(P_1 - \{t,u\}) = 2$, at least one point in $\{d_1,d_2,d_3,d_4\}$ is a white point. If $d_2 = 0$ or $d_4 = 0$ or $d_1 = d_3 = 0$, then $\#_C(P_1 - \{t,u\}) = 2 \neq 1 = \#_C(P_1 - \{q,t,u\}) = \#_C(P_1 - \{p,t,u\}) = \#_C(P_1 - \{p,q,t,u\})$. Again, by an argument similar to the one in Lemma 4.6.6, X is not an MNS set in P.

Thus, $d_2 = d_4 = 1$ and either $d_1 = 1$ or $d_3 = 1$. WLOG let $d_3 = 0$. Then $d_1 = 1$. Since $r = s = d_p = d_q = 1$, p is 18-adjacent to two white components in $N(p)$ after only q is deleted (while $t = u = 1$), i.e., $\{q\}$ and the one containing d_3 in $N(p)$. Thus, p is non-simple in $P - \{q\}$. But since X is a (6,18) MNS set, p is simple in $P - \{q\}$, a contradiction. Similarly, if $d_1 = 0$, then $d_3 = 1$ which implies that q is both simple and non-simple in $P - \{p\}$, a contradiction. Thus, $d_1 = d_3 = 1$ which contradicts the fact that $\#_C(P_1 - \{t,u\}) = 2$. Hence, an m-type set is not a 3D (6,18) MNS set. □

Corollary 4.6.8 *An* {i, ..., m, o, ..., v}*-type set is not a 3D* (6, 18) *MNS set.*

Proof. By Lemma 4.6.5, Lemma 4.6.6 and Lemma 4.6.7, an {i, j, k, l, m}-type set is not an MNS set in P. Then by Corollary 4.3.3, an o-type set is not a (6,18) MNS set, since it contains a k-type subset; a {p, ..., v}-type set is not a (6,18) MNS, since each of them contains a j-type subset. □

4.7 The proofs of main theorems

If a parallel thinning algorithm never deletes any MNS set of points, then the algorithm is connectivity preserving (clearly, "deletes an MNS set" means "completely deletes all points in an MNS set"). We claim that if a parallel thinning algorithm satisfies Theorem 4.1.2–Theorem 4.1.5 according to its adjacency relations, then no MNS set can be deleted completely, and hence the algorithm is connectivity preserving. By Proposition 4.3.4, since a 3D MNS set is contained in a unit lattice cube, we only need to consider the configurations in a unit lattice cube. Since an MNS set cannot be an empty set of black points, we will not discuss the a-type set in the following four proofs.

Proof of Theorem 4.1.2. Let X be an MNS set of a 3D (26,6) image P. Suppose X is a {b,c,d,f,i}-type set (i.e., X is contained in a unit lattice square). By Theorem 4.1.2.(1), such a non-simple set X cannot be deleted completely. Now suppose X is an {e,g,h,j,...,v}-type set (i.e., X contains an e-or h-type subset). By Proposition 4.5.1, X is a component of P. But by Theorem 4.1.2.(2), no black component of P that is contained in a unit lattice cube can be deleted. Hence, X cannot be deleted completely. Since all possible configurations of a unit lattice cube are investigated, Theorem 4.1.2 holds. □

Proof of Theorem 4.1.3. By Corollary 4.6.1, an {o,q,s,...,v}-type set is not a 3D (18,6) MNS set. We consider all other configurations of a unit lattice cube as follows. Let X be an MNS set of a 3D (18,6) image P. Suppose X is an {b,c,d,f,i}-type set (i.e., X is contained in a unit lattice square). By Theorem 4.1.3.(1), X cannot be deleted completely. Suppose X is an {e,g,h,j,l,n,p,r}-type set. Let K be the unit lattice cube containing X. Then there exist two 26- but not 18-adjacent points in $K - X$. By Theorem 4.1.3.(2), such an X cannot be deleted completely. Now suppose X is a {k,m}-type set. By Proposition 4.5.3, X is a component of P. But by Theorem 4.1.3.(3), no such black component of P can be deleted. Hence, X cannot be deleted completely. Since all possible configurations of a unit lattice cube are investigated, Theorem 4.1.3 holds. □

Proof of Theorem 4.1.4. By Corollary 4.6.4, an {f,g,i,...,v}-type set is not a 3D (6,26) MNS set. We consider all other configurations of a unit lattice cube as follows. Let X be an MNS set of a 3D (6,26) image P. Suppose X is a {b,c,d,e}-type set (i.e., X contains at most two black points of P). By Theorem 4.1.4.(1), X cannot be deleted completely. Now suppose X is an h-type set. Then X is the set of corners of a $(\sqrt{2}, \sqrt{2}, \sqrt{2})$ triangle. By Theorem 4.1.4.(2), X cannot be deleted completely. Since all possible configurations of a unit lattice cube are investigated, Theorem 4.1.4 holds. □

Proof of Theorem 4.1.5. By Corollary 4.6.8, an {i,...,m,o,...,v}-type set is not a 3D (6,18) MNS set. We consider all other configurations of a unit lattice cube as follows. Let X be an MNS set of a 3D (6,18) image P. Suppose X is a {b,...,h}-type set (i.e., X contains at most three black points of P). By Theorem 4.1.5.(1), X cannot be deleted completely. Now suppose X is an n-type set. Then X is the set of corners of a $1 \times \sqrt{2}$ parallelogram. By Theorem 4.1.5.(2), X cannot be deleted completely. Since all possible configurations of a unit lattice cube are investigated, Theorem 4.1.5 holds. □

4.8 Discussion

A set of black points that is a (β, ω) MNS set may not be an (ω, β) MNS set. For example, let an f-type set be a component of a 3D (26,6) image. Then such an f-type set is a (26,6) MNS set. But by Lemma 4.6.2, an f-type set is not a (6,26) MNS set. Suppose a set X of black point is a (β, ω) MNS set and must be a component in a (β, ω) image. Then if X is an (ω, β) MNS set, X may not be a cavity in an (ω, β) image. For example, if an e-type set is a (26,6) MNS set, then it must be a component in a (26,6) image. But if an e-type set is a (6,26) MNS set, it may not be a cavity of a (6,26) image. We can simply let $\{p, q\}$ be an e-type set of a (6,26) image P where all points in $K(p, q)$ are black points and all points out of $K(p, q)$ be white points. Then it is not difficult to see that both p and q are simple but $\{p, q\}$ is not simple (since the deletion of $\{p, q\}$ creates a "tunnel"). Thus, $\{p, q\}$ is a (6,26) MNS set. However, for this case, $\{p, q\}$ is not a cavity of P

Theorem 4.1.2–4.1.5 are 3D analogs of Ronse's 2D results. Some results have been proposed on other kinds of 2D and 3D images. For example, Hall in [34] proposed sufficient conditions for verifying a parallel thinning algorithm on 2D (6,6) images; Hafford and Preston in [31] proposed a parallel thinning algorithm on 3D (12,12) images. Refer to [52, 53] for the structures of other images.

A parallel thinning algorithm can be proved to be connectivity preserving if no MNS set can be deleted completely by the algorithm. Since a 3D MNS set is contained in a unit lattice cube, we only need to check 2^8 different configurations of a unit lattice cube. In (18,6), (6,26) and (6,18) images, such a number could be further reduced if all configurations which cannot be MNS sets are ruled out. Thus, to prove a parallel thinning algorithm is connectivity preserving, only a rather small number of configurations of a unit lattice cube need to be checked.

Chapter 5

Using 3D connectivity preservation tests

In this chapter, we are going to introduce some properties based on the results in Chapter 4. Using the results in Chapter 4 and these extended properties, we prove that both of our fully parallel thinning algorithms are connectivity preserving. We also compare our 3D connectivity preservation tests with other similar results. Finally, we verify the connectivity preservation of existing 3D thinning algorithms.

5.1 More about Theorem 4.1.2

The following lemma is useful in establishing proofs of connectivity preservation of thinning algorithms.

Lemma 5.1.1 *Let p and q be two diagonally or diametrically adjacent black points of a 3D* $(26,6)$ *image P where p is simple. Then $\#_B(P \cap (K(p,q) - \{p,q\})) = 1$ if and only if p is simple after q is deleted.*

Proof. Since we are dealing with (26,6) images, by Lemma 4.4.3, if $\#_B(P \cap (K(p,q) - \{p,q\})) = 1$, then p is simple after q is deleted, and, by Lemma 4.4.5, if $\#_B(P \cap (K(p,q) - \{p,q\})) \neq 1$, then p is non-simple after q is deleted. □

Lemma 5.1.2 *Let p and q be two 6-adjacent black points of a 3D $(26,6)$ image P where p is simple. Then $\chi(P \cap (K(p,q) - \{p,q\})) = 1$ if and only if p is simple after q is deleted.*

Sketched Proof. Since p, q are 26-adjacent to every points in $K(p,q) - \{p,q\}$, it can be easily seen that there is no cavity and no "tunnel" in $K(p,q)$. Hence, $\chi(P \cap K(p,q)) = 1$. Similarly, $\chi(P \cap (K(p,q) - \{p\})) = \chi(P \cap (K(p,q) - \{q\})) = 1$.

Since p is simple with $q = 1$ and p is 6-adjacent to q, all black neighbors of p are connected to q in $N(p) - K(p,q)$. Thus, all black neighbors of p in $N(p) - K(p,q)$ are connected to the only black component in $K(p,q) - \{p,q\}$. Hence, p is adjacent to only one black component in $N(p) - \{p,q\}$ and p satisfies Definition 2.3.2.(1) after q is deleted.

By Corollary 4.2.2.(2–3), since $\chi(P \cap K(p,q)) = \chi(P \cap (K(p,q) - \{q\})) = \chi(P \cap (K(p,q) - \{p\}))$ $= \chi(P \cap (K(p,q) - \{p,q\}))$, we have $\chi(P \cap N(p)) = \chi(P \cap (N(p) - \{q\}))$, and $\chi(P \cap (N(p) - \{p\})) =$ $\chi(P \cap (N(p) - \{p,q\}))$. But since p is simple, $\chi(P \cap N(p)) = \chi(P \cap (N(p) - \{p\}))$. Thus $\chi(P \cap N(p))$ $= \chi(P \cap (N(p) - \{q\})) = \chi(P \cap (N(p) - \{p,q\}))$ which implies that p satisfies Definition 2.3.2.(3) after q is deleted. Thus, p is simple after q is deleted. □

For two 6-adjacent points p and q, there are two ways to divide $K(p,q)$ into two two-cubes (a two-cube is a union of two unit lattice cubes sharing a common unit lattice square). When $K(p,q)$ is divided into two two-cubes, these two two-cubes sharing a common face, we call such a face a *two-square*. There are two two-squares in $K(p,q)$ that can divide $K(p,q)$ into two two-cubes.

Lemma 5.1.3 *Let p and q be two 6-adjacent black points of a 3D $(26,6)$ image. Let $S(p,q)$ be either of the two-squares such that $S(p,q)$ divides $K(p,q)$ into two two-cubes, K_1 and K_2. Then $\chi(P \cap K(p,q)) = \chi(P \cap K_1) + \chi(P \cap K_2) - \chi(P \cap S(p,q))$.*

Proof. Using the continuous analog on $P \cap K(p,q)$ and by the inclusion-exclusion principle. □

Lemma 5.1.4 *Let p and q be two 6-adjacent black points of a 3D $(26,6)$ image. Let $S(p,q)$ be either of the two-squares such that $S(p,q)$ divides $K(p,q)$ into two two-cubes, K_1 and K_2. Then $\chi(P \cap K(p,q)) = \#_B(P \cap K_1) + \#_B(P \cap K_2) - \#_B(P \cap S(p,q))$.*

Proof. By the above lemma and Lemma 4.2.3. □

Lemma 5.1.5 *Let p and q be two 6-adjacent black points of a 3D $(26,6)$ image. Let p_1, p_2, p_3, and p_4 be four distinct 6-neighbors of p in $K(p,q)-\{p,q\}$, and let q_1, q_2, q_3 and q_4 be four distinct 6-neighbors of q in $K(p,q)-\{p,q\}$ where p_i is 6-adjacent to q_i. Then there exists exactly one black component in $K(p,q)-\{p,q\}$ and at least one pair of p_i and q_i are white points if and only if p is simple after q is deleted.*

Sketched Proof. WLOG let $p_1 = q_1 = 0$. Let $S(p,q)$ be the two-square in $K(p,q)$ containing p_1 and q_1. Let K_1 and K_2 be the two two-cubes of $K(p,q)$ divided by $S(p,q)$. Suppose $\#_B(P \cap (S(p,q)-\{p,q\})) = 1$. Then $\#_B(P\cap(K(p,q)-\{p,q\})) = 1$ implies that $\#_B(P\cap(K_1-\{p,q\})) = \#_B(P\cap(K_2-\{p,q\})) = 1$. Thus, $\chi(P\cap(K(p,q)-\{p,q\})) = 1$ which by Lemma 5.1.2 implies that p is simple after q is deleted. Now suppose $\#_B(P\cap(S(p,q)-\{p,q\})) = 0$. WLOG let $\#_B(P\cap(K_1-\{p,q\})) = 0$. Then $\#_B(P\cap(K_2-\{p,q\})) = 1$ which, again by Lemma 5.1.2 implies that p is simple after q is deleted.

Now suppose p is simple after q is deleted. Then by Lemma 5.1.2 $\chi(P\cap(K(p,q)-\{p,q\})) = 1$. If at least one point in each of the following sets, $\{p_1,q_1\}$, $\{p_2,q_2\}$, $\{p_3,q_3\}$ and $\{p_4,q_4\}$, is a black point, then $\#_B(P\cap(K_1-\{p,q\})) = \#_B(P\cap(K_2-\{p,q\})) = 1$ and $\#_B(P\cap(S(p,q)-\{p,q\}) = 2$. Then $\chi(P\cap(K(p,q)-\{p,q\})) = 0$, a contradiction. Hence, at least one $\{p_i,q_i\}$ is a set of two white points. But then if there are two black components in $K(p,q)-\{p,q\}$, then the deletion of $\{p,q\}$ changes the Euler characteristic of $K(p,q)$, a contradiction. □

For easier establishing the connectivity soundness of our thinning algorithms, we restate Theorem 4.1.2 as follows. It is not difficult to see that the following proposition is equivalent to Theorem 4.1.2.

Proposition 5.1.6 *A thinning algorithm for 3D $(26,6)$ images preserves connectivity if all the following conditions hold:*

1. *only simple points can be deleted;*
2. *if two black corners, p and q, of a unit lattice square are deleted, then $\{p,q\}$ is simple;*
3. *if three black corners, p, q, and r, of a unit lattice square are deleted, then $\{p,q,r\}$ is simple;*

4. *if four black corners, p, q, r, and s, of a unit lattice square are deleted, then $\{p,q,r,s\}$ is simple; and*

5. *no black component that is contained in a unit lattice cube can be deleted completely.* □

It was pointed out by Kong that Theorem 4.1.2 is equivalent to the following proposition. We give a proof to show the correctness of the following proposition.

Proposition 5.1.7 *A thinning algorithm $\mathcal{A}$ for 3D $(26,6)$ images preserves connectivity if all the following conditions hold:*

1. *whenever $\mathcal{A}$ deletes any subset X of black corners of a unit lattice square, every point p in X is simple after $X - \{p\}$ is deleted.*

2. *no black component contained in a unit lattice cube can be deleted by $\mathcal{A}$ completely in one iteration.*

Proof. It is enough to show that Theorem 4.1.2.(1) is equivalent to Proposition 5.1.7.(1). Suppose a thinning algorithm $\mathcal{A}$ satisfies Theorem 4.1.2.(1) in 3D (26,6) images. Let X be a set of black points in any 3D (26,6) image such that X is contained in a unit lattice square and is deleted by $\mathcal{A}$. Then X is simple. Let p be a point in X. Then by Theorem 4.1.2.(1), $X - \{p\}$ is also simple. Now by Proposition 4.3.1, since X and $X - \{p\}$ are both simple, p is simple after $X - \{p\}$ is deleted. Since p is arbitrarily chosen, any point in X is simple after all other point in X are deleted. Hence, Proposition 5.1.7.(1) holds.

Now suppose $\mathcal{A}$ satisfies Proposition 5.1.7.(1). Let X be a set of black points of any 3D (26,6) image where X is contained in a unit lattice square and is deleted by $\mathcal{A}$. Suppose X contains only one point p. Then by Proposition 5.1.7.(1), p is simple (since $X - \{p\}$ is an empty set and is simple). Thus, any black point deleted by $\mathcal{A}$ is simple. Now suppose X contains two points $\{p,q\}$. Then both p and q are simple points. Again by Proposition 5.1.7.(1), q is simple after p is deleted. Thus, $\langle p,q\rangle$ is a simple sequence and hence $\{p,q\}$ is a simple set. Suppose X contains three points $\{p,q,r\}$. Then similarly, p, q and r are simple points and $\{p,q\}$, $\{p,r\}$ and $\{q,r\}$ are simple sets. Then by Proposition 5.1.7.(1), r is simple after $\{p,q\}$ is deleted which implies that $\langle p,q,r\rangle$ is a simple sequence and $\{p,q,r\}$ is a simple set. Suppose X contains four points $\{p,q,r,s\}$. By a similar argument, X

is a simple set. Thus, every set of black points contained in a unit lattice square and deleted by $\mathcal{A}$ is a simple set. Hence, Theorem 4.1.2.(1) holds. □

5.2 Verifications of our 3D thinning algorithms

5.2.1 The connectivity preservation of the medial-face algorithm

Let p and q be black points where p is deleted by Algorithm 3.2.3, that is, p satisfies $\mathcal{O}_f$. By "$q \in \mathcal{O}_f(p)$" we mean that q must be a black point for p to satisfy $\mathcal{O}_f$, that is, p does not satisfy $\mathcal{O}_f$ after q is deleted. Then "$q \notin \mathcal{O}_f(p)$" means that p still satisfies $\mathcal{O}_f$ after q is deleted. The following lemmas are useful in proving the connectivity preservation of Algorithm 3.2:3.

Lemma 5.2.1 *Let p, q be two diagonally or diametrically adjacent black points in a $(26,6)$ 3D image where both of p and q satisfy $\mathcal{O}_f$. Then $q \notin \mathcal{O}_f(p)$ and $p \notin \mathcal{O}_f(q)$.*

Proof. By investigating both configurations in Figure 3.2, whether p satisfies $\mathcal{O}_f$ or not does not depend on any diagonal or diametrical black neighbor of p. Thus, $q \notin \mathcal{O}_f(p)$. Similarly, $p \notin \mathcal{O}_f(q)$. □

Lemma 5.2.2 *Let p, q be two 6-adjacent black points in a $(26,6)$ 3D image where both of p and q satisfy $\mathcal{O}_f$. Then either $q \notin \mathcal{O}_f(p)$ or $p \notin \mathcal{O}_f(q)$.*

Proof. Let a–q–p–b be four distinct points on a straight line such that a is 6-adjacent to q and b is 6-adjacent to p. Suppose $p \in \mathcal{O}_f(q)$ and $q \in \mathcal{O}_f(p)$. By checking Figure 3.2, $p \in \mathcal{O}_f(q)$ implies that $a = 0$ and $q \in \mathcal{O}_f(p)$ implies that $b = 0$. Then either p or q must be a north, east or up border point. WLOG let p be an east border point. Since $q = w(p)$ and $a = w(w(p))$, a must be a black point for p to satisfy $\mathcal{O}_f$, a contradiction. □

Now we establish the following result by using the above lemmas.

Proposition 5.2.3 *The set of black points deleted by $\mathcal{O}_f$ satisfies all five conditions of Proposition 5.1.6.*

Proof. In Figure 3.2.(a), p is 6-adjacent to exactly one white component in the 18-neighborhood of p and is 26-adjacent to exactly one black component in $N(p)$. Hence, the configuration in Figure 3.2.(a) can only delete simple points. Similarly, the configurations in Figure 3.2.(b–c) can only delete simple points. Thus, $\mathcal{O}_f$ can only delete simple points and Proposition 5.1.6.(1) holds.

Let p and q be two distinct black corners of a unit lattice square. Suppose both of them satisfy $\mathcal{O}_f$. Since p is 18-adjacent to q, by Lemma 5.2.1 and 5.2.9, WLOG let $q \notin \mathcal{O}_f(p)$. Then p still satisfies $\mathcal{O}_f$ after q is deleted, that is, p is simple after q is deleted. Then, by Definition 2.3.5, $\{p, q\}$ is simple. Thus, Proposition 5.1.6.(2) holds.

Let p, q and r be three distinct black corners of a unit lattice square where q is diagonally adjacent to r. Suppose all three of them satisfy $\mathcal{O}_f$. By Lemma 5.2.1, $q \notin \mathcal{O}_f(r)$ and $r \notin \mathcal{O}_f(q)$. Suppose $p \notin \mathcal{O}_f(r)$. Then r satisfies $\mathcal{O}_f$ after $\{p, q\}$ is deleted which implies that r is simple after $\{p, q\}$ is deleted. Since $\mathcal{O}_f$ satisfies Proposition 5.1.6.(2), $\{p, q\}$ is simple (with $r = 1$) which, by Definition 2.3.5, implies that $\{p, q, r\}$ is simple. Now suppose $p \in \mathcal{O}_f(r)$. Then, by Lemma 5.2.2, $r \notin \mathcal{O}_f(p)$. Hence, both of p and q satisfy $\mathcal{O}_f$ after r is deleted. By Proposition 5.1.6.(2), $\{p, q\}$ is simple after r is deleted. Thus, $\{p, q, r\}$ is simple and Proposition 5.1.6.(3) holds.

Let p, q, r and s be four distinct black corners of a unit lattice square where p is the one with the smallest sum of coordinates. By the above paragraph, $\{q, r, s\}$ is simple (with $p = 1$). By Rule 3.2.2, p is deleted if p is simple after $\{q, r, s\}$ is deleted. Hence, if all four points are deleted, the set of the four points is simple and Proposition 5.1.6.(4) holds.

We prove Proposition 5.1.6.(5) by showing that one cannot construct a black component contained in a unit lattice cube such that every point of this component satisfies $\mathcal{O}_f$. Consider the cube shown in Figure 5.1. Assume all points not in the cube are white points. Suppose $p_7 = 1$. Then p_7 satisfies $\mathcal{O}_f$ (otherwise any set of other black points in the cube is not a component). If p_5 is in $\mathcal{O}_f(p_7)$, then p_7 satisfies $\mathcal{O}_f$ as an east border point and hence $w(p_5) = 1$, a contradiction. Hence, p_5 is not in $\mathcal{O}_f(p_7)$. Similarly, both of p_3 and p_8 are not in $\mathcal{O}_f(p_7)$. However, the fact that p_7 satisfies $\mathcal{O}_f$ does not depend on the rest of the points in the cube. Thus, p_7 does not satisfy $\mathcal{O}_f$ which implies $p_7 = 0$. Similarly, $p_5 = p_8 = 0$ and therefore, $p_6 = 0$. Then the remaining points are contained in a unit square. By the above paragraph , such a component can be deleted completely. Hence, Proposition 5.1.6.(5) holds. □

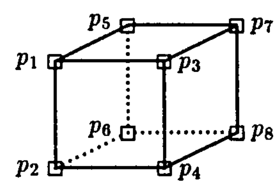


Figure 5.1: A unit lattice cube where p_3, p_5 and p_8 are the south, west and down neighbors of p_7, respectively. Other adjacency relations in this cube can be obtained accordingly.

Theorem 5.2.4 *Algorithm 3.2.3 is a connectivity preserving parallel thinning algorithm.*

Proof. For any 3D image, let X be the set of black points satisfying $\mathcal{O}_f$. Let X' be the set of black points satisfying Algorithm 3.2.3. It is trivial that $X' \subseteq X$. We need to show that X' also satisfies all five conditions in Proposition 5.1.6. But, again, this is trivial since X satisfies all these conditions. □

5.2.2 The connectivity preservation of the medial-line algorithm

To simplify the proof of our medial-line algorithm to preserve connectivity, we use $\mathcal{O}_l$ to denote the set of all deleting templates of Algorithm 3.3.3. A black point in a 3D image is said to *satisfy* $\mathcal{O}_l$ if it satisfies any deleting template in $\mathcal{O}_l$.

Clearly, every point deleted by our algorithm satisfies $\mathcal{O}_l$. It is not difficult to see that it is enough to prove the connectivity preservation of our algorithm by showing that the set of black points deleted by $\mathcal{O}_l$ satisfies all five conditions of Proposition 5.1.6. For verifying Proposition 5.1.6.(1), we must show that $\mathcal{O}_l$ can only delete simple points from any 3D image.

Lemma 5.2.5 *$\mathcal{O}_l$ can only delete simple points of 3D images.*

Proof. WLOG we claim that every configuration in Figure 3.7 deletes only simple points which implies that every deleting template of $\mathcal{O}_l$ can only delete simple points. Let p be a black point. It is trivial to see that if $N(p)$ matches the configuration shown in Figure 3.7(d), then p is a simple point. Suppose $N(p)$ matches any of the other three configurations in (a), (b) or (c). Clearly, for all three cases, p is 26-adjacent to only one black component in $N(p) - \{p\}$, and 6-adjacent to only one 6-connected set W of white points in $N_{18}(p)$ where every unmarked 6-neighbor of p is 6-adjacent to W. Hence, p is adjacent to only one white component in the 18-neighborhood of p. Thus, by Definition 2.3.2, p is a simple point. □

Let p and q be black points where p satisfies $\mathcal{O}_l$. By investigating all configurations in Figure 3.7, it can be seen that if $q \in \mathcal{O}_l(p)$, then q must be 18-adjacent to p. Hence, we have the following lemma.

Lemma 5.2.6 *Let p, q be two diametrically adjacent black points in a 3D $(26,6)$ image where both p and q satisfy $\mathcal{O}_l$. Then $q \notin \mathcal{O}_l(p)$ and $p \notin \mathcal{O}_l(q)$.* □

Lemma 5.2.7 *Let p, q, r, and s be corners of a unit lattice square of a 3D $(26,6)$ image where p is diagonally adjacent to q. Suppose both p and q satisfy $\mathcal{O}_l$. Then either $q \notin \mathcal{O}_l(p)$ or $p \notin \mathcal{O}_l(q)$.*

Proof. Suppose $q \in \mathcal{O}_l(p)$ and $p \in \mathcal{O}_l(q)$ (see the following diagram). Since $q \in \mathcal{O}_l(p)$, p must satisfy a Class D template in $\mathcal{O}_l$ and $p_1 = p_2 = p_3 = 0$. Since $p \in \mathcal{O}_l(q)$, q also must satisfy a Class D template in $\mathcal{O}_l$ and $q_1 = q_2 = q_3 = 0$.

		p_1	p_2
	r	p	p_3
q_1	q	s	
q_2	q_3		

But then either q is one of the following neighbors of p: $sd(p)$, $su(p)$, $se(p)$, $sw(p)$, $wu(p)$ and $wd(p)$, or vice versa. WLOG let $q = sw(p)$. Then by Rule 3.3.1.(3e) at least one point in $\{q_1, q_2, q_3\}$ must be a black point, a contradiction. □

Corollary 5.2.8 *Let p, q, r and s be corners of a unit lattice square of a 3D $(26,6)$ image where p is diagonally adjacent to q. Suppose both p and q satisfy $\mathcal{O}_l$ and either $r = 1$ or $s = 1$. Then $q \notin \mathcal{O}_l(p)$ and $p \notin \mathcal{O}_l(q)$.*

Proof. If $p \in \mathcal{O}_l(q)$, then p must satisfy a Class D template and $r = s = 0$; if $q \in \mathcal{O}_l(p)$, then q must satisfy a Class D template and $r = s = 0$. □

Suppose $q \in \mathcal{O}_l(p)$ and q is 6-adjacent to p. WLOG let $q = d(p)$. By investigating the basic configurations in Figure 3.7, p must be an up border point (since if p is not an up border point, then $q \notin \mathcal{O}_l(p)$). Then we have the following lemma.

Lemma 5.2.9 *Let p, q be two 6-adjacent black points in a 3D $(26,6)$ image where both p and q satisfy $\mathcal{O}_l$. Then either $q \notin \mathcal{O}_l(p)$ or $p \notin \mathcal{O}_l(q)$.*

Proof. Follow an argument similar to the one in Lemma 5.2.2. □

Now we claim the following result by using the above lemmas.

Proposition 5.2.10 *The set of black points deleted by $\mathcal{O}_l$ satisfies all five conditions of Proposition 5.1.6.*

Proof. To prove this proposition, we claim that the parallel deletion by $\mathcal{O}_l$ satisfies Proposition 5.1.6. By Lemma 5.2.5, every point satisfying $\mathcal{O}_l$ is a simple point. Hence Proposition 5.1.6.(1) holds.

Let p and q be two distinct black corner points of a unit lattice square. Suppose both of them satisfy $\mathcal{O}_l$. Since p is 18-adjacent to q, by Lemma 5.2.7 and 5.2.9, WLOG let $q \notin \mathcal{O}_l(p)$. Then p still satisfies $\mathcal{O}_l$ after q is deleted, that is, p is simple after q is deleted. Then, by Definition 2.3.5, $\{p,q\}$ is simple. Thus, Proposition 5.1.6.(2) holds.

Let p, q and r be three distinct black corner points of a unit lattice square where q is diagonally adjacent to r. Suppose all three of them satisfy $\mathcal{O}_l$. Since $p = 1$, by Corollary 5.2.8, $q \notin \mathcal{O}_l(r)$ and $r \notin \mathcal{O}_l(q)$. If $p \notin \mathcal{O}_l(r)$, then r satisfies $\mathcal{O}_l$ after $\{p,q\}$ is deleted which implies that r is simple after $\{p,q\}$ is deleted. But since $\mathcal{O}_l$ satisfies Proposition 5.1.6.(2), $\{p,q\}$ is simple (with $r = 1$) which by Definition 2.3.5 implies that $\{p,q,r\}$ is simple. Suppose $p \in \mathcal{O}_l(r)$. Then, by Lemma 5.2.9, $r \notin \mathcal{O}_l(p)$. Hence, both p and q satisfy $\mathcal{O}_l$ after r is deleted. Again, by Proposition 5.1.6.(2), since $\{p,q\}$ is simple after r is deleted, $\{p,q,r\}$ is simple and Proposition 5.1.6.(3) holds.

Let p, q, r and s be four distinct black corner points of a unit lattice square shown below such that every point in $\{p,q,r,s\}$ satisfies $\mathcal{O}_l$.

$$\begin{array}{cccc} - & x_2 & w_4 & - \\ w_1 & p & q & x_1 \\ b_1 & r & s & w_3 \\ - & w_2 & b_2 & - \end{array}$$

Suppose there exists a point in $\{p, q, r, s\}$, say r, such that $p \in \mathcal{O}_l(r)$ and $s \in \mathcal{O}_l(r)$. Then by Lemma 5.2.9, $r \notin \mathcal{O}_l(p)$ and $r \notin \mathcal{O}_l(s)$. Since, by Corollary 5.2.8, $r \notin \mathcal{O}_l(q)$, after r is deleted each point in $\{p, q, s\}$ still satisfies $\mathcal{O}_l$. But since $\mathcal{O}_l$ satisfies Proposition 5.1.6.(3), $\{p, q, s\}$ is simple after r is deleted. Thus, by Definition 2.3.5, $\{p, q, r, s\}$ is simple.

Now suppose there exists a point in $\{p, q, r, s\}$, say r, such that $p \notin \mathcal{O}_l(r)$ and $s \notin \mathcal{O}_l(r)$. Since, by Corollary 5.2.8, $q \notin \mathcal{O}_l(s)$, after $\{p, q, s\}$ is deleted, r still satisfies $\mathcal{O}_l$ and hence is simple. But $\mathcal{O}_l$ satisfies Proposition 5.1.6.(3) which implies that $\{p, q, s\}$ is simple. Thus, by Definition 2.3.5, $\{p, q, r, s\}$ is simple.

Now suppose for every point x in $\{p, q, r, s\}$ exactly one of its 6-neighbors in $\{p, q, r, s\}$ is in $\mathcal{O}_l(x)$. WLOG we consider the following case: $q \in \mathcal{O}_l(p)$, $p \in \mathcal{O}_l(r)$, $r \in \mathcal{O}_l(s)$, $s \in \mathcal{O}_l(q)$. It is easily seen that $q \in \mathcal{O}_l(p)$ implies $w_1 = 0$, $p \in \mathcal{O}_l(r)$ implies $w_2 = 0$, $r \in \mathcal{O}_l(s)$ implies $w_3 = 0$, and $s \in \mathcal{O}_l(q)$ implies $w_4 = 0$. Since $w_1 = w_3 = 0$, either p or s is a north, east or up border point. WLOG let s be an east border point. Then $b_1 = 1$. Since $w_2 = w_4 = 0$, WLOG let q be a north border point. Then $b_2 = 1$ which implies that r cannot satisfy any template in Class A as a south border point. Since $b_1 = b_2 = 1$, r cannot satisfy any template in Class B or Class C. Since $p = s = b_1 = b_2 = 1$, r cannot satisfy any template in Class D. Thus, r must satisfy some template in Class A as an up border point or a down border point. Then $p \notin \mathcal{O}_l(r)$, a contradiction. Hence, Proposition 5.1.6.(4) holds.

We prove Proposition 5.1.6.(5) by showing that one cannot construct an object contained in a unit cube (see Figure 5.1) such that every point of this component satisfies $\mathcal{O}_l$. Assume all points not in the cube are 0's. Suppose $p_7 = 1$. Then p satisfies $\mathcal{O}_l$. By Rule 3.3.1.(2c) and (3e), if either p_1 or p_5 is in $\mathcal{O}_l(p_7)$, then there exists a black point connected to p_7 but not in the cube, a contradiction. Hence, both p_1 and p_5 are not in $\mathcal{O}_l(p_7)$. Similarly, all of p_3, p_4, p_6, p_8 are not in $\mathcal{O}_l(p_7)$. But the fact that p_7 satisfies $\mathcal{O}_l$ does not depend on p_2. Thus, p_7 does not satisfy $\mathcal{O}_l$ which implies that p_7

= 0.

Suppose $p_8 = 1$. Since all points out of the cube are 0's, similar as above, the only way for p_8 to satisfy $\mathcal{O}_l$ is $p_7 = 1$. But since $p_7 = 0$, $p_8 = 0$. Now suppose $p_5 = 1$. Since all points out of the cube are 0's, similar as above, the only way for p_5 to satisfy $\mathcal{O}_l$ is either $p_7 = 1$ or $p_8 = 1$, a contradiction. Hence, $p_5 = 0$. Finally, suppose $p_6 = 1$. Since all points out of the cube are 0's, similarly the only way for p_6 to satisfy $\mathcal{O}_l$ is either $p_5 = 1$ or $p_7 = 1$ or $p_8 = 1$. This leads a contradiction. Hence, $p_6 = 0$.

Since p_5, p_6, p_7 and p_8 are all 0's, the remaining four points are contained in a unit square. Now since Proposition 5.1.6.(4) holds, if every black point in the square with corners $\{p_1, p_2, p_3, p_4\}$ satisfies $\mathcal{O}_l$, then the set is simple. Hence, Proposition 5.1.6.(5) holds. □

Now, we can introduce the final theorem – our parallel thinning algorithm, Algorithm 3.3.3, for generating skeletons like medial lines is connectivity preserving.

Theorem 5.2.11 *Algorithm 3.3.3 is a connectivity preserving parallel thinning algorithm.*

Proof. Similar to Theorem 5.2.4. □

5.3 Other 3D results vs. our 3D results

Hall in [34] proposed sufficient and necessary conditions for 2D (8,4) thinning algorithms to preserve connectivity. In [32], he proposed some results in 3D (26,6) images based on his 2D (8,4) results. His 3D results contain the following conditions.

Rule 5.3.1

Class B *A thinning algorithm $\mathcal{A}$ belongs to Class B if it only deletes simple points.*

RC1-26 *Whenever a black point p is deletable by a thinning algorithm $\mathcal{A}$ and x, y are two black points in $N(p) - \{p\}$, then there is a black 26-path from x to y in $N(p) - \{p\}$ containing no black points deletable by $\mathcal{A}$ other than (perhaps) x or y.*

RC2-26 *No pairwise 26-adjacent black component is completely deleted by $\mathcal{A}$.*

RC3-26 *Whenever a black point p is deleted by a thinning algorithm $\mathcal{A}$ and p has two 6-neighbors x and y where x is a white point and y is either a white point or is a deletable black point, then there is a 6-path from x to y in $N(p)$ containing no black points other than (perhaps) y.*

We claim that if a (26,6) thinning algorithm satisfies our 3D connectivity preservation tests then it satisfies Hall's 3D results as well.

Proposition 5.3.2 *Let $\mathcal{A}$ be a 3D (26,6) thinning algorithm satisfying Theorem 4.1.2. Then all the following conditions hold:*

1. *$\mathcal{A}$ is a Class-B algorithm;*
2. *$\mathcal{A}$ satisfies RC2-26; and*
3. *$\mathcal{A}$ satisfies RC3-26.*

Proof. By Theorem 4.1.2.(1), $\mathcal{A}$ deletes only simple points. Thus, (1) holds. It can be shown that Theorem 4.1.2.(2) implies RC2-26. Thus (2) holds.

Let x be a white 6-neighbor of p where p is deleted by $\mathcal{A}$. If y is a white 6-neighbor of p, then since p is simple, x is 6-connected to y by a white 6-path in $N(p)$. If y is a black 6-neighbor of p where y is deleted by $\mathcal{A}$, then p is simple after y is deleted (since, by Theorem 4.1.2.(1), $\{p, y\}$ is a simple set). Thus, x is connected to y by a white path in $N(p)$ after y is deleted which shows that (3) holds. □

Before introducing the next proposition, we define a ***minimal k-path*** between two points, p and q, to be a k-path between them containing the minimal number of points. A ***black minimal k-path*** from a black point p to a black point q is a black k-path from p to q containing the minimal number of black points. Note that there could be more than one black minimal k-path between p and q. In 3D (26,6) images, a ***black minimal path*** is a black minimal 26-path.

Proposition 5.3.3 *Let $\mathcal{A}$ be a 3D (26,6) thinning algorithm satisfying Theorem 4.1.2. Then $\mathcal{A}$ satisfies RC1-26.*

Proof. Suppose x, y are two black points in $N(p) - \{p\}$ where p is deleted by $\mathcal{A}$. We claim that $\mathcal{A}$ satisfies RC1-26 by showing that after the parallel deletion by $\mathcal{A}$, there is still a black 26-path joining x and y in $N(p)$.

It is trivial if x and y are 26-adjacent. Suppose they are not 26-adjacent. Since p is simple, x is connected to y by a black minimal path π_1 in $N(p) - \{p\}$. Let q be a point in π_1 such that $q \neq x$ and $q \neq y$. Then q is 18-adjacent to p. Thus, $\{p, q\}$ is contained in a unit lattice square. By Proposition 5.1.7.(1), p is simple after q is deleted. Hence, x is still connected to y by a minimal path π_2 in $N(p) - \{p, q\}$.

Suppose there exists a black point r in π_2 such that r is also deleted by $\mathcal{A}$ at the same time. Then, similarly, $\{p, r\}$ is also contained in a unit lattice square. Since r is deleted by $\mathcal{A}$, r is simple. If r is not 26-adjacent to q, then since q is not in $N(r)$, the deletion of q does not change the simplicity of r. Thus, r is still simple after q is deleted. Now suppose r is 26-adjacent to q. Since $p = 1$ is in $K(q, r)$, $\chi(K(q, r) - \{q, r\}) = 1$. This, by Lemma 5.1.1, implies that r is simple after only q is deleted.

Thus, after q is deleted, every point in $N(p) - \{p\}$ that is deleted by $\mathcal{A}$ is still simple. By induction, p is simple after every point in $N(p) - \{p\}$ that satisfies $\mathcal{A}$ is deleted. Hence, there exists a black path joining x and y in $N(p)$ after the application of $\mathcal{A}$. □

By the above proposition, our Theorem 4.1.2 is a "subset" of Hall's 3D results. The reason for this situation is that some 3D thinning algorithms may satisfy Hall's conditions in some 3D images, but the output image does not preserve connectivity.

Proposition 5.3.4 *A 3D thinning algorithm which satisfies Hall's 3D conditions may not preserve connectivity.*

Proof. It is enough to give a special thinning algorithm in some special 3D image and show that the application of the algorithm satisfies Hall's conditions in that image but does not preserve connectivity. Suppose a thinning algorithm contains only two $5 \times 5 \times 5$ deleting templates which delete points in $5 \times 5 \times 5$ neighborhoods like p and q does (see Figure 5.2). Then both p and q are simple and the algorithm is in Class B. Since every pair of black points in $N(p) - \{p\}$ and in $N(q) - \{q\}$ are 26-connected respectively, RC1-26 is satisfied. Clearly, no black component contained

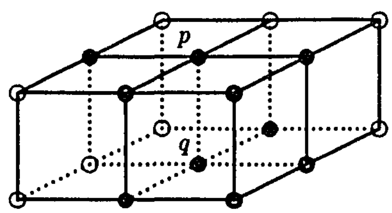


Figure 5.2: All unmarked points and all double-circled points are white points. The deletion of p or q alone satisfies Hall's 3D results. But the deletion of $\{p,q\}$ does not preserve connectivity.

in a unit lattice cube can be deleted completely. Hence, RC2-26 is satisfied. It is easily observed that every pair of white points in $N(p)$ are joined by a white 6-path in $N(p)$, and every white point in $N(p)$ and q are also joined by a white 6-path in $N(p)$. This is also true for q. Hence, RC3-26 is satisfied.

Since, in the image shown in Figure 5.2, there is only one black component but no cavities and no tunnels, $\chi(P \cap K(p,q)) = 1$. However, we know $\chi(P \cap (K(p,q) - \{p,q\})) = 4$ (points) $-$ 4 (edges) $= 0$ which by Lemma 5.1.2 implies that p is non-simple after q is deleted and q is non-simple after p is deleted. Hence, $\{p,q\}$ is not a simple set. Thus, the thinning algorithm passes Hall's results but does not preserve connectivity. □

Hall in [34] argued that if $\{p,q\}$ are both deleted, then there is a white 6-path in the union of the 18-neighborhoods of p and q such that the path is *transformable* to the 6-path formed by p and q. WLOG consider the 6-path from p, passing through all double-circled points, then to q. This 6-path is not *transformable* to the 6-path formed by p, q since there are two diametrically adjacent black points penetrating the circle formed by the two 6-paths. All other possible white 6-paths may have the same problem.

In [32], Hall proposed the following sufficient conditions for 2-subfield and 4-subfield thinning algorithms to preserve connectivity in 3D (26,6) images. We introduce these two conditions and their proofs as follows.

Proposition 5.3.5 *A 4-subfield parallel thinning algorithm $\mathcal{A}$ in a 3D $(26, 6)$ image P preserves connectivity if all the following conditions hold.*

1. *Only simple points can be deleted.*
2. *Let p and q be two diagonally adjacent black points. Then p, q cannot be deleted at the same time if $\#_B(K(p,q) - \{p,q\}) \neq 1$;*
3. *No black components of two, three or four pairwise diagonally adjacent points can be deleted completely.*

Proof. We show that if $\mathcal{A}$ satisfies the above proposition, then $\mathcal{A}$ also satisfies Proposition 5.1.6. It is trivial that (1) above is equivalent to Proposition 5.1.6.(1), and (3) above is equivalent to Proposition 5.1.6.(5).

Since the four corners of a unit lattice square belong to two different subfields, any three, or four corners of such a square cannot be deleted at the same time. Hence, Proposition 5.1.6.(3) and (4) are satisfied.

We only need to consider the case when two black points p and q are diagonally adjacent and are deleted by $\mathcal{A}$. It is clear that $\#_B(P \cap K(p,q)) = 1$ (since every possible black point in $K(p,q) - \{p,q\}$ is 26-adjacent to $\{p,q\}$). By Lemma 4.4.5, if $\#_B(P \cap (K(p,q) - \{p,q\})) \neq 1$, then p is non-simple after q is deleted. Similarly, q is non-simple after p is deleted. By (2) above, since at least one of $\{p,q\}$ cannot be deleted, Proposition 5.1.6.(2) is satisfied. □

Proposition 5.3.6 *A 2-subfield parallel thinning algorithm $\mathcal{A}$ in a 3D $(26, 6)$ image P preserves connectivity if*

1. *Only simple points can be deleted.*
2. *No black components of two diametrically adjacent points can be deleted completely.*

Proof. We show that if $\mathcal{A}$ satisfies the above proposition, then $\mathcal{A}$ also satisfies Proposition 5.1.6.

It is trivial that (1) above is equivalent to Proposition 5.1.6.(1), and (2) above is equivalent to Proposition 5.1.6.(5). Since the four corners of a unit lattice square belong to four different subfields, any two, three, or four corners of such a square cannot be deleted at the same time. Hence, Proposition 5.1.6.(2), (3) and (4) are satisfied. □

5.4 Verifications of existing 3D thinning algorithms

5.4.1 Proofs of existing 3D thinning algorithms

On the basis of Theorem 4.1.2, we propose the following sufficient conditions for 6-subiteration thinning algorithms to preserve connectivity. The idea was motivated by three 6-subiteration thinning algorithms by Gong and Bertrand in [27], Mukherjee, Das, and Chatterji in [77], and Tsao and Fu in [124].

Proposition 5.4.1 *Any 6-subiteration parallel thinning algorithm $\mathcal{A}$ in a 3D* $(26,6)$ *image P is connectivity preserving if all of the following conditions hold:*

1. *any black point p deleted by $\mathcal{A}$ is simple;*
2. *p is simple (in the 2D sense) in $N_x(p)$ and $N_z(p)$; and*
3. *(a) $s(p) = 1$; or*

   (b) p has two or more black neighbors in $N_x(p)$, $N_z(p)$, and $N(p)$.

Proof. We show that $\mathcal{A}$ satisfies all five conditions of Proposition 5.1.6. By (1) above, it is trivial that Proposition 5.1.6.(1) is satisfied. For Proposition 5.1.6.(2), we consider the following cases for two black points p and q that are deleted by $\mathcal{A}$ at the same time.

Case 1. p is 6-adjacent to q.

Case 2. p is diagonally adjacent to q and they have the same y-coordinate.

Case 3. p is diagonally adjacent to q and they have different y-coordinates.

For Case 1, suppose (3.a) holds, i.e., both $s(p)$ and $s(q)$ are black points and by (2), both of them are simple in $N_x(p)$, $N_z(p)$, $N_x(q)$ and $N_z(q)$ respectively. WLOG let $q = e(p)$. We know that $s(p) = s(q) = 1$ and $n(p) = n(q) = 0$. By (2) above, every black point in $K(p,q) - \{p,q,s(p),s(q)\}$ is 26-connected to $\{s(p),s(q)\}$ by a black path in $K(p,q) - \{p,q,s(p),s(q)\}$. Consider the two-square $S(p,q)$ containing $\{n(p),n(q),p,q,s(p),s(q)\}$. It is not difficult to see that $\#_B(P \cap (S(p,q) - \{p,q\}))$ $= 1$. Let K_u be the two-cube in $K(p,q)$ containing $S(p,q)$ and $u(p)$ and $u(q)$. Let K_d be the other two-cube in $K(p,q)$. Then $\#_B(P \cap (K_u - \{p,q\})) = \#_B(P \cap (K_d - \{p,q\})) = 1$. Hence by

Lemma 5.1.4, $\chi(P \cap (K(p,q) - \{p,q\})) = 1$ which by Lemma 5.1.2 implies that p is simple after q is deleted. Thus, $\{p,q\}$ is simple.

Now suppose (3.b) holds where p has two or more black neighbors in $N_x(p)$ and $N_z(p)$ and q also has two or more black neighbors in $N_x(q)$ and $N_z(q)$. WLOG let $q = e(p)$. Then either $s(p) = 1$ or $s(q) = 1$. By a similar argument as in the previous paragraph, we know that $\{p,q\}$ is simple.

For Case 2, suppose (3.a) holds. We know that $s(p) = 1$. WLOG let $q = ed(p)$. Since p is simple in $N_x(p)$ and $N_z(p)$, every possible black point in $K(p,q) - \{p,q\}$ is connected to $s(p)$ by a black path in $K(p,q) - \{p,q\}$. Thus, there is only one black component in $K(p,q) - \{p,q\}$ which by Lemma 4.4.3 implies that p is simple after q is deleted and $\{p,q\}$ is a simple set.

Now suppose (3.b) holds. Suppose both $s(p)$ and $s(q)$ are white points (since if either one of them is a black point, then by the above argument, it is done). Since p is simple and has two or more black neighbors in $N(p)$, there exists a black point in $K(p,q) - \{p,q\}$. If there is only one black component in $K(p,q) - \{p,q\}$, then by Lemma 4.4.3, p is simple after q is deleted and $\{p,q\}$ is a simple set. Suppose there are two black components in $K(p,q) - \{p,q\}$. Then there must be two black points, x and y, in $K(p,q) - \{p,q\}$ – one in the unit lattice square containing $n(p)$ and $n(q)$ and the other one in the unit lattice square containing $s(p)$ and $s(q)$. If x and y are both in $N_x(p)$ or $N_z(p)$, then by (2) above, they cannot be in two distinct black components in $K(p,q) - \{p,q\}$. Suppose x is in $N_x(p)$ but is not in $N_z(p)$. WLOG let $x = sd(p)$. Again by (2), since p has two or more black neighbors in $N_x(p)$, $d(p) = 1$. Then x and y again cannot be in two distinct black components in $K(p,q) - \{p,q\}$. Thus, $\{p,q\}$ is simple.

For Case 3, suppose (3.a) holds. We know $s(p) = 1$. Then by Lemma 4.4.3, p is simple after q is deleted and $\{p,q\}$ is simple. Now suppose (3.b) holds. WLOG let $q = ed(p)$. Since p has two or more black neighbors in $N_z(p)$, $s(p) = 1$. Thus, $\{p,q\}$ is simple. Thus, Proposition 5.1.6.(2) is satisfied for all three cases.

For Proposition 5.1.6.(3), we consider the case where all three black corners p, q and r in a unit lattice square have the same y-coordinates (if not, then at least one is not a north border point and hence cannot be deleted at the same time). Suppose p and q are both 6-adjacent to r. Let s be the other corner of the square. Then s is 6-adjacent to p. Suppose $n(s) = 1$. For (3.a), since $s(p) = 1$ and p is simple in $N_x(p)$ and $N_z(p)$, $s = 1$. For (3.b), since p has two or more black neighbors in

$N_x(p)$ and $N_z(p)$, $s = 1$. Thus, $K(p,q) - \{p,q,r\} = 1$ which by Lemma 4.4.3 implies that $\{p,q,r\}$ is simple. Now suppose $n(s) = 0$. Then for (3.a), $s(p) = 1$ and for (3.b), p has two or more black neighbors in $N_x(p)$ and $N_z(p)$, $s = 1$. Thus, $K(p,q) - \{p,q,r\} = 1$ which again by Lemma 4.4.3 implies that $\{p,q,r\}$ is simple. Thus, Proposition 5.1.6.(3) is satisfied.

For Proposition 5.1.6.(4), we consider the case where all four black corners p, q, r and s in a unit lattice square have the same y-coordinate (if not, then two of them are not north border points and hence p, q, r and s cannot be deleted at the same time). Then for (3.a), $s(p) = 1$ and for (3.b), p has two or more black neighbors in $N_x(p)$ and $N_z(p)$, there is a black point in the south layer of $\{p,q,r,s\}$. Thus, $K(p,q) - \{p,q,r,s\} = 1$ which again by Lemma 4.4.3 implies that $\{p,q,r,s\}$ is simple. Thus, Proposition 5.1.6.(4) is satisfied.

For Proposition 5.1.6.(5), consider (3.a), since $s(p) = 1$ and is not a north border point, $s(p)$ cannot be deleted. For (3.b), consider a set X of black points contained in a unit lattice cube where all these points are deleted at the same time. If there is a black neighbor of the set where that point is not a north border point, then it is not deleted. Suppose all black points in X are north border points. Then the following cases must occur if X is not contained in a unit lattice square (since if X is contained in a unit lattice square, then it is proved in the above four paragraphs).

Case 4. X contains three pairwise diagonally adjacent point.

Case 5. X contains two diametrically adjacent points.

For Case 4, it is not difficult to show that since there are three pairwise diagonally adjacent points in X, at least one point p of X has a black south neighbor $s(p)$. Since $p = 1$, $s(p)$ is not a north border point and cannot be deleted at the same time. For Case 5, we know that X contains two diametrically adjacent simple points, p and q, where both p and q have two or more black neighbors in $N(p)$ and $N(q)$ respectively. Thus, there is a black point r in $K(p,q) - \{p,q\}$. Again it is not difficult to show that there is at least one point in $\{p,q,r\}$ has a black south neighbor which cannot be deleted by $\mathcal{A}$. Hence, no black component contained in a unit lattice cube can be deleted completely by $\mathcal{A}$ at the same time and Proposition 5.1.6.(5) is satisfied. □

Now we introduce some existing 3D thinning algorithms and establish proofs of them.

Tsao and Fu in [124] also proposed a two-step connectivity preserving 6-subiteration thinning algorithm. In Proposition 3.1.1, we present the first step of their algorithm. We now establish the proof.

Proof of Proposition 3.1.1. Since p is a north border point, Proposition 3.1.1.(2) implies that p is simple (in the 2D sense) in $N_x(p)$ and $N_z(p)$. Since p is simple, by Proposition 5.4.1, this proposition is proved. □

Gong and Bertrand in [27] proposed another connectivity preserving 6-subiteration thinning algorithm shown in Proposition 3.1.2. Its proof is as follows.

Proof of Proposition 3.1.2. Since p is a north border point, Proposition 3.1.2.(2) implies that p in simple (in the 2D sense) in $N_x(p)$ and $N_z(p)$. By Proposition 5.4.1, we only need to show that p is a simple point. By Proposition 3.1.2.(2–3), every possible black 26-neighbor of p is 26-connected to $s(p)$ by a black 26-path in $N(p) - \{p\}$, and every possible white 6-neighbor of p is 6-connected to $n(p)$ by a white 6-path in the 18-neighborhood of p. Thus, by Proposition 2.3.3, p is a simple point. □

Mukherjee, Das, and Chatterji in [77] proposed a 6-subiteration thinning algorithm called *MESPTA*. Their algorithm can be proved to preserve connectivity by showing the correctness of Proposition 3.1.3 as follows.

Proof of Proposition 3.1.3. By Proposition 5.4.1, we only need to show that p is a simple point. By Proposition 3.1.3.(2–3) above, every possible black 26-neighbor of p is 26-connected to $s(p)$ by a black 26-path in $N(p) - \{p\}$, and every possible white 6-neighbor of p is 6-connected to $n(p)$ by a white 6-path in the 18-neighborhood of p. Thus, by Proposition 2.3.3, p is a simple point. □

5.4.2 Counterexamples to existing 3D thinning algorithms

Now we introduce some existing algorithms and give counterexamples to show that these algorithms are not connectivity preserving. Mukherjee, Das, and Chatterji in [76] proposed a 6-subiteration thinning algorithm called *ESPTA*. Their algorithm can be stated as follows.

A thinning algorithm in 3D (18, 6) images is connectivity preserving if both of the following conditions are satisfied:

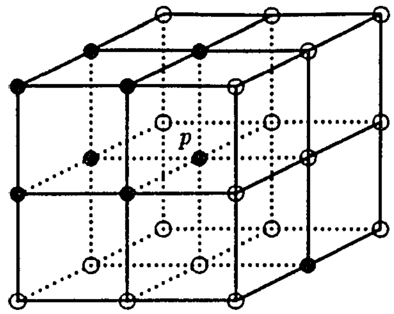


Figure 5.3: In a 3D (18,6) image, p in this $N(p)$ is non-simple. However, p can be deleted by the algorithm ESPTA.

1. $n(p) = 0$ and $s(p) = 1$; and
2. p is a simple non-end point (in the 2D sense) in $N_x(p)$ and in $N_z(p)$.

It was pointed out by Gong and Bertrand in [26] that this algorithm does not preserve connectivity (see the counterexample given by Gong and Bertrand in Figure 5.3). In Figure 5.3, p is a north border point, and is a simple non-end point (in the 2D sense) in $N_x(p)$ and in $N_z(p)$. But p is not a simple point in such a neighborhood configuration. Thus, The ESPTA algorithm does not preserve connectivity.

Morgenthaler in [74] proposed a definition of *end points* in 3D (26,6) images as follows. Let p be a black point in a 3D (26,6) image. Then a unit lattice cube K of $N(p)$ is called *thin* if p and every black 6-neighbor of p in K is adjacent to a white point in K. K is called *thick* if it is not thin. If p or one of its black 6-neighbors is not in a thick cube, then p is called an *end point*. A black point which is not an end point is called a *non-end point*.

According to this definition, a black point which has only one black neighbor is an end point. In that paper, Morgenthaler proposed the following 6-subiteration thinning algorithm.

> A thinning algorithm in 3D $(26,6)$ images is connectivity preserving if all north border simple non-end points are deleted.

One counterexample of this algorithm is shown in Figure 5.4. It is not difficult to generate a

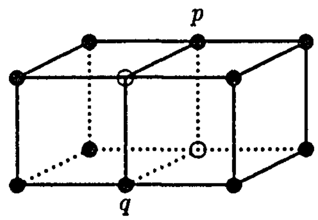


Figure 5.4: Each of p and q can be deleted by Morgenthaler's 3D thinning algorithm. But the deletion of $\{p, q\}$ does not preserve connectivity.

configuration such that each black point in Figure 5.4 (except p and q) is non-simple and the eight points belong to two distinct black components in the image after $\{p, q\}$ is deleted. Then in such a configuration, since p and q are both north border simple non-end points, $\{p, q\}$ is deleted. But then the deletion of $\{p, q\}$ splits the original black component into two disconnected black components. Thus, this algorithm does not preserve connectivity.

Tsao and Fu in [125] took a different approach where they proposed another definition of *non-end points* as follows. Let $C \subseteq B$ be the set of all simple north border points in a 3D (26,6) image P where B is the set of all black points of P. Then $p \in C$ is called a *non-end point* if all the following hold:

1. p is adjacent to a point in $B - C$;
2. every black 6-neighbor of p is adjacent to a point in $(B - C) \cap N(p)$;
3. there is only one black component in $(B - C) \cap (N(p) - \{p\})$; and
4. p has at least two black neighbors.

Note that by Tsao and Fu's definition, if p has only one black neighbor, then p is not a non-end point. Based on this definition, Tsao and Fu proposed a 6-subiteration parallel thinning algorithm as follows.

> A thinning algorithm in 3D $(26, 6)$ images is connectivity preserving if all north border simple non-end points are deleted.

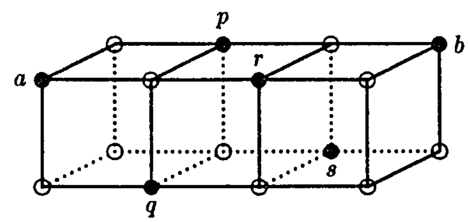


Figure 5.5: Each of p, q, r and s can be deleted by Tsao and Fu's thinning algorithm. However, the deletion of $\{p, q, r, s\}$ does not preserve connectivity.

One counterexample to their algorithm is shown in Figure 5.5. It is not difficult to generate a configuration such that a and b are non-simple points and belong to two distinct black components after $\{p, q, r, s\}$ is deleted. Since, in this example, p, q, r and s are both north border simple non-end points, $\{p, q, r, s\}$ is deleted. But then a and b are disconnected. Thus, this algorithm does not preserve connectivity.

5.5 Discussion

In Section 5.1, we established more properties in 3D (26,6) images. These properties are useful in the remaining sections of this chapter and in the later chapters. We also proved that a thinning algorithm satisfies Hall's 3D (26,6) results in [32] does not guarantee to preserve connectivity. However, Hall's 2- and 4-subfield (26,6) connectivity preservation tests in [32] can be proved to be special cases of our 3D general connectivity preservation tests.

Existing 3D (26,6) thinning algorithms in [27, 77, 124] were proved to preserve connectivity by using our 3D results. The proofs were quite short. For the 3D (18,6) thinning algorithm in [76], we introduce the counterexample given by Gong and Bertrand in [26]. For other existing 3D (26,6) thinning algorithms in [74, 125], we provided counterexamples to show that these algorithms does not guarantee to preserve connectivity.

Chapter 6

3D computerized tests

In this chapter, we are going to introduce the concept of the implementation of our 3D connectivity preservation tests for 3D (26,6) images. We first introduce the implementation of Ronse's 2D connectivity preservation tests. A thinning algorithm can be represented by a set of deleting templates. For a parallel thinning algorithm, its deleting templates can delete adjacent black points at the same time. In the implementation of connectivity preservation tests, the concept of "template merging" is very important. For example, two deleting templates T_1 and T_2 may delete two 6-adjacent black points x_1 and x_2 respectively at the same time. We say that T_1 and T_2 are *mergable.*

6.1 The concept of templates merging

This concept was introduced by Kong (see [46]). We consider the 2D case first. For a 2D parallel thinning algorithm, these templates may delete more than one black point and may delete adjacent black points at the same time. The *core* of a template is the part of the template consisting of the 3×3 points at the center. (Note that for the 3D case, the *core* of a template is the part of the template consisting of the $3 \times 3 \times 3$ points at the center.) A template is called *simple* if all points satisfying the template are simple points. Since the simplicity of a point p is determined by $N(p)$, the simplicity of a template is determined by its core. For example, in the following diagram, the template with the central point p is simple and the template with the central point q is non-simple (where a "2" matches either a black point or a white point).

$$\begin{array}{ccccc} 2 & 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 \\ 2 & 1 & p & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 \end{array} \qquad \begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & q & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{array}$$

Let p contain a value in $\{0,1,2\}$ where 0 stands for a white point, 1 for a black point and 2 for a don't-care point. Define an operation $\otimes$ as follows:

1. $p \otimes p = p$
2. $2 \otimes p = p \otimes 2 = p$
3. $0 \otimes 1 = 1 \otimes 0 =$ undefined

Let T be a template of a 2D thinning algorithm. Then we write $T[i,j]$ for the value of the point, with coordinate (i,j), in T where the central point of the core of T has the coordinate $(0,0)$.

Let T_1 and T_2 be templates of a 2D thinning algorithm. We define $T_1 +_H T_2$ and $T_1 +_V T_2$ to be the templates such that

$$(T_1 +_H T_2)[i,j] = \begin{cases} 0 & \text{if } i = 1, j = 0 \\ T_1[i,j] \otimes T_2[i-1,j] & \text{otherwise} \end{cases}$$

$$(T_1 +_V T_2)[i,j] = \begin{cases} 0 & \text{if } i = 0, j = -1 \\ T_1[i,j] \otimes T_2[i,j+1] & \text{otherwise} \end{cases}$$

We define the following relations between T_1 and T_2:

1. T_1 and T_2 is *WE-mergable* if, for every i, j, $(T_1 +_H T_2)[i,j]$ is defined where the central point of T_1 is the west neighbor of the central point of T_2.
2. T_1 and T_2 is *NS-mergable* if, for every i, j, $(T_1 +_V T_2)[i,j]$ is defined where the central point of T_1 is the north neighbor of the central point of T_2.

Note that the central point of T_2 is turned to a white point in the merged template for both cases. For example, in the first case shown below, p and q are both black points in the corresponding templates and the point related to q in the merged template (the east neighbor of p in $T_1 +_H T_2$) is a white point.

$$\begin{matrix} 0&0&1&1&0 \\ 0&0&1&2&0 \\ 0&0&p&2&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ &&T_1&& \end{matrix} \qquad \begin{matrix} 0&1&1&0&0 \\ 0&1&1&0&0 \\ 0&1&q&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ &&T_2&& \end{matrix} \qquad \begin{matrix} 0&0&1&1&0 \\ 0&0&1&1&0 \\ 0&0&p&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ &&T_1 +_H T_2&& \end{matrix}$$

$$\begin{matrix} 0&0&1&1&0 \\ 0&0&1&2&0 \\ 0&0&p&1&0 \\ 0&0&1&2&0 \\ 0&0&0&0&0 \\ &&T_1&& \end{matrix} \qquad \begin{matrix} 0&1&1&0&0 \\ 0&1&1&0&0 \\ 0&1&q&0&0 \\ 0&0&1&0&0 \\ 0&0&0&0&0 \\ &&T_2&& \end{matrix} \qquad \begin{matrix} 0&0&1&1&0 \\ 0&0&1&1&0 \\ 0&0&p&0&0 \\ 0&0&1&1&0 \\ 0&0&0&0&0 \\ &&T_1 +_H T_2&& \end{matrix}$$

$$\begin{matrix} 0&0&1&1&0 \\ 0&0&1&2&0 \\ 0&0&p&1&0 \\ 0&0&1&1&0 \\ 0&0&0&0&0 \\ &&T_1&& \end{matrix} \qquad \begin{matrix} 0&1&1&0&0 \\ 0&1&1&0&0 \\ 0&1&q&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ &&T_2&& \end{matrix}$$

For the first case, T_1 and T_2 are WE-mergable and the merged template is simple; for the second case, T_1 and T_2 are still WE-mergable, but the merged template is non-simple; for the third case, T_1 and T_2 are not WE-mergable.

For the 3D case, let T_1 and T_2 be templates of a 3D thinning algorithm. Similar to the analogous relations in the 2D case, we define the following relations between T_1 and T_2:

1. T_1 and T_2 is *WE-mergable* if the central point of T_1 is the west neighbor of the central point of T_2 and every point in the merged template is defined.

2. T_1 and T_2 is *NS-mergable* if the central point of T_1 is the north neighbor of the central point of T_2 and every point in the merged template is defined.

3. T_1 and T_2 is *UD-mergable* if the central point of T_1 is the up neighbor of the central point of T_2 and every point in the merged template is defined.

4. T_1 and T_2 is *NU-mergable* if the central point of T_1 is the up-north neighbor of the central point of T_2 and every point in the merged template is defined.

5. T_1 and T_2 is *WU-mergable* if the central point of T_1 is the up-west neighbor of the central point of T_2 and every point in the merged template is defined.

6. T_1 and T_2 is *NW-mergable* if the central point of T_1 is the north-west neighbor of the central point of T_2 and every point in the merged template is defined.

6.2 2D computerized tests

The method used in this section was introduced by Kong (see [46]). When we say "the 2D computerized (Ronse) tests" we mean the implementation of Ronse's 2D (8,4) results. Since the Ronse tests can determine whether a thinning algorithm preserves connectivity by verifying a finite number of configurations, it can be computerized. Hall in [34] proposed the first such 2D computerized tests. Kong in [46] gave another implementation of the 2D Ronse tests. The difference between the above two implementations will be introduced later in this section.

By Ronse's results in Theorem 4.1.1, a thinning algorithm $\mathcal{A}$ in a 2D (8,4) image P preserves connectivity if all the following three conditions are satisfied: (1) Every deleting template of $\mathcal{A}$ is simple; (2) Let x_1 and x_2 be two black points of P and let T_1 and T_2 be two templates of $\mathcal{A}$ where x_1 is deleted by T_1 and x_2 is deleted by T_2. If x_1 is the north or the west neighbor of x_2, then $\{x_1, x_2\}$ is a simple set. (That is, T_1 and T_2 together can only delete simple sets of two points); and (3) Let X be any black component contained in a unit lattice square. Then at least one point in X does not satisfy any deleting template in $\mathcal{A}$.

Since whether a black point p of a 2D (8,4) image P is simple or not is determined by $N(p)$, we only need to check the template core of each deleting template for assuring only simple points can be deleted. The eight points in a template core (other than the central point p) can be ordered in some arbitrary order, p_1, p_2, ..., p_8. Then the simplicity of the template core can be determined by a function $f(p_1, p_2, \ldots, p_8)$. Since each point in a template core may have three different values, 0, 1 or 2 (don't-care). There are a total of 3^8 different configurations of a template core. We construct a table with the following structure

Table $[p_1][p_2][p_3][p_4][p_5][p_6][p_7][p_8]$

We use this table to verify the simplicity of every template of a thinning algorithm. Such a table is called a *2D simple table.*

Let p and q be two black points of a 2D (8,4) image that are deleted by a thinning algorithm $\mathcal{A}$. Ronse in [95] showed that if $\mathcal{A}$ preserves connectivity, then $\{p, q\}$ is a simple set which implies that p is simple after q is deleted, and q is simple after p is deleted. Suppose p is the west neighbor of q. Then we call (p, q) a *west-east pair.* If $\{p, q\}$ is simple, then such a pair is called a *simple* west-east pair. A *north-south pair* and a *simple north-south pair* are defined similarly.

When we try to verify whether two WE-mergable templates, T_1 and T_2, together can only delete simple west-east pairs (see condition 2 above), we only need to check whether $T_1 +_H T_2$ is simple. Similarly, when we try to verify whether two NS-mergable templates, T_1 and T_2, together can only delete simple north-south pairs, we only need to check whether $T_1 +_V T_2$ is simple. Note that the black central point of T_2 is turned to white in both $T_1 +_H T_2$ and $T_1 +_V T_2$. Hence, we only need to check the following conditions for verifying the connectivity preservation of a 2D (8,4) thinning algorithm.

1. every template in $\mathcal{A}$ is a simple template;
2. for any two WE-mergable templates T_1 and T_2 in $\mathcal{A}$, the central point of T_1 is simple in $T_1 +_H T_2$;
3. for any two NS-mergable templates T_1 and T_2 in $\mathcal{A}$, the central point of T_1 is simple in $T_1 +_V T_2$;
4. no black component contained in a unit lattice square can be deleted completely by $\mathcal{A}$.

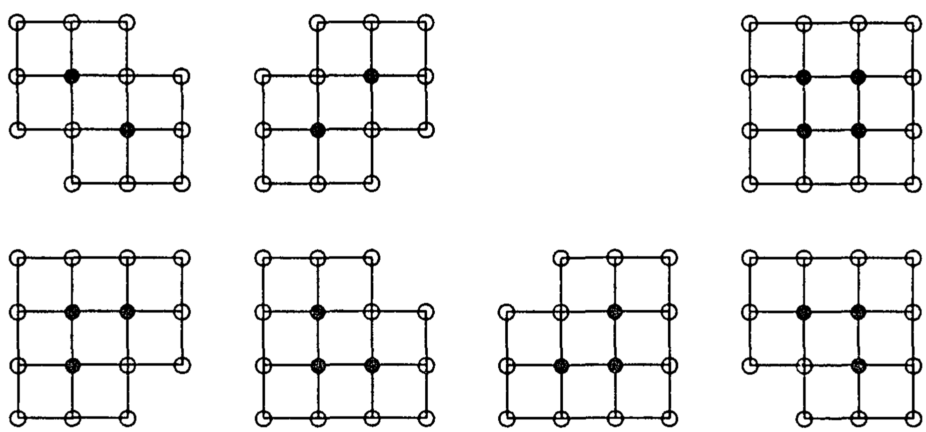


Figure 6.1: To preserve topology, no components can be deleted completely.

The 2D Ronse tests must also guarantee that no component containing two diagonally opposite corners in a unit lattice square can be deleted completely. Thus, any configuration in Figure 6.1 cannot be deleted completely in one iteration.

For all configurations in Figure 6.1, we create seven arrays each of which contains four Boolean variables where each Boolean variable corresponds to one corner of the unit lattice square in the middle of each configuration. If the corresponding corner is a black point, then the Boolean variable is assigned a Boolean value *black*, otherwise assigned a Boolean value *white*. We then match the neighborhood of each black point in every configuration with every template of the thinning algorithm $\mathcal{A}$. If there is a match, then change the value of the corresponding Boolean variable of the corresponding array to *white*. After the entire procedure is finished, if there is an array in which all four variables are *white*, then we know the thinning algorithm $\mathcal{A}$ can delete such a corresponding black component and thus fails to preserve connectivity.

Hall's computerized tests takes about one hour for verifying whether a thinning algorithm preserves connectivity. Kong's computerized tests takes about one minute for the same job. The main difference is that for verifying the simplicity of $T_1 +_H T_2$ (and similarly of $T_1 +_V T_2$), Hall's program trace the black and white configuration of every merged template to see whether it is simple. Our program uses the simple table and hence we only need the time for retrieving the value in the table.

The trade-off is that our approach needs $3^8 = 6,561$ bytes memory space to store the table. Such a memory space can be found in the RAM of almost every computer system.

6.3 3D computerized tests

When we say "the 3D computerized tests" we mean the implementation of Theorem 4.1.2. Since, by Theorem 4.1.2, we can determine whether a 3D (26,6) thinning algorithm preserves connectivity by verifying a finite number of configurations, we can implement that theorem (although it is not very easy to do so). One possible input of the 3D computerized tests is as follows. All n points in a template (other than the central point p) can be arranged in some arbitrary order, p_1, p_2, ..., p_n. Since each point in a template may have three different values, 0, 1 or 2 (don't-care), each template can be expressed as a product of a Boolean function by the following rules:

1. if p_i is a 0, then put p'_i in the product;
2. if p_i is a 1, then put p_i in the product; and
3. if p_i is a 2, then do not put p_i or p'_i in the product.

Then the set of deleting templates of a thinning algorithm is converted to a Boolean function $f(p_1, p_2, \ldots, p_{26})$ of the sum-of-products form where each product represents a deleting template. Let T be a deleting template. Then $T(p)$ means that p is a black point in an image such that the neighborhood of p matches T, i.e., p is deleted by T. Following a similar argument as in the 2D case and by the result in Proposition 5.1.7, we need to check the following conditions for verifying the connectivity preservation of a 3D (26,6) thinning algorithm $\mathcal{A}$.

1. every template of $\mathcal{A}$ is a simple template;
2. for any two templates $T_1(p)$ and $T_2(q)$ of $\mathcal{A}$ where p and q are corners of a unit lattice square, p is still simple after q is deleted;
3. for any three templates $T_1(p)$, $T_2(q)$ and $T_3(r)$ of $\mathcal{A}$ where p, q and r are corners of a unit lattice square, p is still simple after q and r are deleted;

4. for any four templates $T_1(p)$, $T_2(q)$, $T_3(r)$, and $T_4(r)$ of $\mathcal{A}$ where p, q, r and s are corners of a unit lattice square, p is still simple after q, r and s are deleted;

5. no black component contained in a unit lattice cube can be deleted completely by $\mathcal{A}$.

We use a 3D simple table to speed up the verification of the simplicity of every template of a thinning algorithm where the construction of such a 3D table is introduced in Chapter 7. Let $\mathcal{O}$ be the set of all deleting templates of a testing thinning algorithm. Suppose T_1, T_2, ..., T_n are templates in $\mathcal{O}$ such that $T_1, T_2, \ldots, T_n$ can be merged into a merged template X. Then X is called *simple* if the central point of any T_i where $1 \leq i \leq n$ is simple in X after the central points of all other templates are deleted in X. We use the following subroutines to test the above five conditions:

SimpleTemplate For testing condition (1) above. This is a function for testing whether every deleting template of the algorithm is simple. It returns *TRUE* if all templates are simple, and returns *FALSE* if any template is non-simple.

SimpleSquare For testing conditions (2–4) above. This is a function for testing whether a merged template of any two, three or four mergable deleting templates for deleting two, three or four corners of a unit lattice square respectively can only delete simple sets (of black points). It returns *TRUE* if all possible merged templates are simple, and returns *FALSE* if any merged template is non-simple.

SimpleCube For testing condition (5) above. This is a function for testing whether any black component contained in a unit lattice cube can be deleted completely. It returns *TRUE* if no such black components can be deleted completely, and returns *FALSE* if there is such a black component that can be deleted completely.

Let *ConnectivityPreservation* be a Boolean variable indicating whether the thinning algorithm preserves connectivity. The structure of our computerized tests is as follows:

ConnectivityPreservation :=
SimpleTemplate($\mathcal{O}$) and *SimpleSquare*($\mathcal{O}$) and *SimpleCube*($\mathcal{O}$);

The following is the code of the function *SimpleTemplate*; the time complexity of this function is $O(n)$ where n is the number of deleting templates of $\mathcal{O}$.

```
function SimpleTemplate(O: all templates) return(Boolean);
begin
    for each template T_i in O do
    if T_i is non-simple, then return(FALSE);

    return(TRUE);
end;
```

Consider the function *SimpleSquare*, we first try to merge two templates. If they are mergable and the merged template is simple, then we try to merge the resulting template with the third template. If this can be done and the new merged template is still simple, then we try to add the fourth template into the merged template. If two templates are not mergable or are mergable but the merged template is non-simple, then we don't have to worry about merging the third and the fourth templates. There are two cases for merging two templates T_1 and T_2:

Case 1. T_1 and T_2 are merged in such a way that their central points are 6-adjacent in the merged template, i.e., T_1 and T_2 are WE-, NS-, or UD-merged.

Case 2. T_1 and T_2 are merged in such a way that their central points are diagonally adjacent in the merged template, i.e., T_1 and T_2 are NU-, WU- or NW-merged.

Let p_i be the central point of a template T_i. For Case 2, let $T_1 +_{NU} T_2$ be the template resulting from NU-merge of T_1 and T_2. Then p_1 and p_2 are two diagonally opposite corners of a unit lattice square. There are two ways to merge T_3 with $T_1 +_{NU} T_2$:

1. Add T_3 into $T_1 +_{NU} T_2$ in such a way that $p_3 = s(p_1) = u(p_2)$.

2. Add T_3 into $T_1 +_{NU} T_2$ in such a way that $p_3 = d(p_1) = n(p_2)$.

Then T_3 is said to *SU-mergable* with $T_1 +_{NU} T_2$ if condition (1) applied, and said to *DN-mergable* with T_{NU} if condition (2) applied.

Let $T_1 +_{WU} T_2$ and $T_1 +_{NW} T_2$ be the templates resulting from WU- and NW-merge of T_1 and T_2, respectively. Following a similar argument, we define the *EU-merging* and *DW-merging* of T_3 with $T_1 +_{WU} T_2$, and the *SW-merging* and the *EN-merging* of T_3 with $T_1 +_{NW} T_2$. The following is the code of the function *SimpleSquare*; the time complexity of this test is $O(n^4)$.

```
function SimpleSquare (O: all templates) return(Boolean);
var S1, S2, T_NU, T_WU, T_NW: templates;
begin
  for each template Ti in O do
  for each template Tj in O do
  if Ti and Tj are WE-, NS- or UD-mergable to a non-simple template, then return(FALSE);

  for each template Ti in O do
  for each template Tj in O do begin
    if Ti and Tj are NU-mergable to a merged template T_NU then begin
      if T_NU is non-simple, then return(FALSE);
      for each template Tk in O do begin
        if Tk and T_NU are SU-mergable to a merged template S1 then begin
          if S1 is non-simple, then return(FALSE);
          for each template Tl in O do
          if Tl and S1 are DN-mergable to a non-simple template, then return(FALSE);
        end;
        if Tk and T_NU are DN-mergable to a non-simple template S2, then return(FALSE);
      end;
    end;
      ⋮
    (* We omit the cases for Ti and Tj being WU- and NW-mergable *)
      ⋮
  end;
```

return(*TRUE*);

end;

The following variables are employed to introduce the concept of the function *SimpleCube*. We first define the following user-defined types and variables.

type

CubeType : record

corner : array[8] of $\{0, 1\}$;

color : array[8] of $\{0, 1\}$;

end;

var *Cube* : array[255] of *CubeType*;

Assign an integral index to each corner of a unit lattice cube in an arbitrary order. Since there are eight corners of a unit lattice cube, there are totally 256 black and white configurations of a unit lattice cube. Clearly, we don't need to consider the configuration where all eight corners of the unit lattice cube are white points. Thus there are 255 different configurations need to be verified *Cube* is a table containing all 255 different unit lattice cubes where each unit lattice cube *Cube*[*i*] contains eight numbered corners. *Cube*[*i*].*corner*[*j*] is the j^{th} corner point of the unit lattice cube *Cube*[*i*]. Each such corner is associated with a Boolean value *Cube*[*i*].*color*[*j*]; if *Cube*[*i*].*corner*[*j*] is a black point, then *Cube*[*i*].*color*[*j*] contains 1, otherwise, *Cube*[*i*].*color*[*j*] contains 0. Now we can introduce the function *SimpleCube*; its time complexity is $O(n)$.

function *SimpleCube* ($\mathcal{O}$: all templates) return(boolean);

begin

for each template T_i in $\mathcal{O}$ do

for each *Cube*[*j*] do

for each *corner*[*k*] of *Cube*[*j*] do

if *Cube*[*j*].*color*[*k*] = 1 and *Cube*[*j*].*corner*[*k*] can be deleted by T_i, then *Cube*[*j*].*color*[*k*] = 0;

for each *Cube*[*j*] do

```
    if all color[k] of Cubes[j] are 0's, then return(FALSE);

    return(TRUE);
end;
```

6.4 Techniques to speed-up 3D computerized tests

For the function *SimpleTemplate*, we can establish a simple table to check the simplicity of each template. Refer to Chapter 7 for the details of such a simple table. Since the function *SimpleSquare* has time complexity $O(n^4)$, we emphasize the development of techniques to speed-up this particular test.

Based on 6-subiteration thinning algorithms by Gong and Bertrand in [27], Mukherjee, Das, and Chatterji in [77], and Tsao and Fu in [124], we introduced Proposition 5.4.1 in Chapter 5. To implement Proposition 5.4.1, we need the function *SimpleTemplate* to guarantee that every template of a thinning algorithm is simple.

Before introducing the implementation of other conditions in Proposition 5.4.1, recall that for 6-subiteration thinning algorithms, we only consider the case when only north border points can be deleted. Let $T(p)$ be a deleting template of a thinning algorithm where p is the central point. Consider the $3 \times 3 \times 3$ template core of $T(p)$. Let $T_x(p)$ and $T_z(p)$ be the 2D 3×3 neighborhoods of p in $T(p)$ where $T_x(p)$ and $T_z(p)$ are defined similar to $N_x(p)$ and $N_z(p)$ in the $3 \times 3 \times 3$ neighborhood $N(p)$. $T_x(p)$ and $T_z(p)$ are called ***simple*** if they can only delete simple points in the 2D (8,4) sense; they are ***non-simple*** if they are not simple.

We introduce a revised version of the function *SimpleSquare* and call it *SimpleSquare2*. The new function is much faster then the old one since the time complexity of the function *SimpleSquare2* is $O(n)$ (rather than $O(n^4)$) where n is the number of templates in the testing thinning algorithm.

```
function SimpleSquare2 (O: all templates) return(Boolean);
var NorthBorder, Simple2D, SouthBlack, EndPoint: Boolean;
begin
    NorthBorder = Simple2D = SouthBlack = EndPoint = TRUE;
```

for each template T in $\mathcal{O}$ with a central point p do

begin

if $n(p) \neq 0$, then *NorthBorder* = *FALSE*;

if $T_x(p)$ or $T_z(p)$ is non-simple, then *Simple2D* = *FALSE*;

if $s(p) \neq 1$, then *SouthBlack* = *FALSE*;

if p has only one black neighbor in $T_x(p)$, $T_z(p)$ or $T(p)$, then *EndPoint* = *FALSE*;

end;

if *NorthBorder* and *Simple2D* and (*SouthBlack* or *EndPoint*), then return(*TRUE*)

else return(*FALSE*);

end;

With this revised function, the structure of our main program is turned to the following one:

ConnectivityPreservation :=

SimpleTemplate($\mathcal{O}$) and (*SimpleSquare2*($\mathcal{O}$) or *SimpleSquare*($\mathcal{O}$)) and *SimpleCube*($\mathcal{O}$);

The 6-subiteration thinning algorithms by Gong and Bertrand, by Mukherjee, Das, and Chatterji, and by Tsao and Fu were taken as testing cases for a program of this new structure. The program took about one minute to verify the connectivity soundness for each of these algorithms on a 33 MHz IBM-PC 386 AT.

Compared with the 2D case, few fully parallel thinning algorithms for the 3D case have been proposed. Hence, we cannot get a good intuition about the fundamental restrictions of such thinning algorithms. We believe that with more 3D fully parallel thinning algorithms, we can have a better understanding of 3D fully parallel thinning algorithms and can establish more efficient computerized tests.

6.5 Discussion

The 3D connectivity preservation tests are efficient for checking whether a given 3D parallel thinning algorithm preserves connectivity. They are also very useful in designing new 3D thinning algorithms.

It is not very difficult to implement the 2D Ronse tests. But, this is not true for the 3D case. Two major difficulties are as follows.

1. The size of the simple point lookup table is very big. Since for each point in $N(p)$, there are three different possible colors – black, white or don't care, totally there are 3^{26} combinations. If we use one byte for one entry, then the size of the table is $2,540$ MBytes which is not feasible for current computer systems.

2. Function *SimpleSquare* of the 3D computerized tests needs to merge up to four templates. For an algorithm contains n deleting templates, the time complexity is $O(n^4)$. If n is a large number, the merging process will take a long time.

In the next chapter, we will introduce a very efficient method for establishing a much smaller simple table. The second difficulty can be partially reduced by using heuristic techniques to skip the difficulties related to the merging of four templates. Further improvements are dependent on the speed and the size of the main memory of computer systems.

Chapter 7

3D simplicity Tests

The material in this chapter is from a joint paper with Dr. Longin Latecki (see [59]). In this chapter, we consider only the case of 3D (26,6) images. Every thinning algorithm can be stated as a list of deleting templates. If a neighborhood of a given black point matches one of these templates, the point will be deleted. In order to make the design of a thinning algorithm easy to handle and therefore the list of templates as short as possible, every template is described in three colors: 0, 1 and 2 (don't-care).

As shown in Chapter 6, we need a function *SimpleTemplate* to verify the simplicity of every template of a thinning algorithm. In this chapter, we present a 3D algorithm for determining the simplicity of any three-color deleting template. This algorithm is memory efficient and fast. Hence it can be a very useful tool for 3D computerized tests.

7.1 3D lookup tables

The task is not trivial due to the number of such configurations in 3D case. There are 26 neighbors which have to be considered and each of them can have one of the three colors, therefore, we obtain 3^{26} different 3D configurations. Since each configuration is either simple or non-simple, we can construct a lookup table to store all this information. Note that this lookup table is an array in which each entry is a Boolean value (simple or non-simple). Each three-color configuration is arbitrarily assigned a unique number identifying the corresponding entry in the lookup table. Given a neighborhood configuration of some point p, the number of the entry can be computed and the Boolean value in that entry can be obtained. Such a Boolean value signifies the simplicity of p in this neighborhood configuration.

This procedure is very fast, since it computes the number of the entry in constant time with respect to the size of the input configuration, and then retrieves the Boolean value of the entry in one unit time. However, the lookup table has to contain $3^{26} \approx 2.54 \times 10^{12}$ entries. Since each entry contains a Boolean value, we can use one bit for each entry. Thus the table uses about $2,540/8 \approx 318$ GBytes, which certainly is not feasible.

7.2 A space-saving algorithm for the simplicity test

Different approaches for finding efficient ways to determine the simplicity of a black point p in a 3D image have been proposed. For the 2D case, two major approaches are as follows.

- Computation of the number of black components in $N(p) - \{p\}$, which can be accomplished using Rutovits or Hildtich crossing numbers (see [53]).
- Retrieval from a lookup table. Since in the 2D case the simplicity of a point depends only on its eight neighbors, each having one of the three possible colors, a lookup table with $3^8 = 6561$ entries is sufficient to hold all possible configurations. Each entry of the lookup table contains a Boolean value indicating the simplicity of the testing configuration.

Hall's 2D computerized tests follows the first approach. Another 2D computerized tests by Kong follows the second approach. The second approach is faster than the first one, since it only requires calculation of the number of the entry in the lookup table assigned to a given configuration, and then retrieving the Boolean value of the entry with this number. The lookup table occupies only $6561/8 \approx 820$ Bytes (if each entry occupies one bit).

Our 3D tests (shown in Chapter 6) follow the second approach. However, since the memory space is a serious difficulty in extending this approach to the 3D case, here we present an algorithm for the simplicity test which uses a lookup table of size 1.4 MBytes. Let p be a black point of a 3D

$(26, 6)$ image. We first restate the two conditions of Proposition 2.3.3 where p is simple if both C1 and C2 are true.

C1 p is adjacent to only one black component in $N(p) - \{p\}$; and

C2 p is adjacent to only one white component in $N_{18}(p)$.

Let X be a set of three-color points. An *assignment* for X is a function which assigns white (value 0) or black (value 1) to each don't-care point in X. The strategy of the test is the following:

> Given a neighborhood $N(p)$ of a black point p, we try to find an assignment for $N(p)$ which makes C1 or C2 fail. If such an assignment exists, p is non-simple. If this is not possible, then C1 and C2 are satisfied, and therefore p is simple.

A space-saving algorithm for the simplicity test

function simple($N(p)$): Boolean;

begin

  if not C1($N(p)$) or not C2($N_{18}(p)$), then return(*FALSE*) else return(*TRUE*);

end;

(* C1($N(p)$) is true if and only if p is adjacent to only one black component in $N(p) - \{p\}$ *)

function C1($N(p)$): Boolean;

begin

  (* Test 1.1. p has a black 6-neighbor *)

  if p has a black 6-neighbor q then

  begin

    Let $N_q(p)$ be $N(p) - (\{p, q\} \cup \{x : x$ is contained in the layer containing q which spans a plane perpendicular to line segment pq (the 8 unmarked points in Figure 7.1.(a))$\})$;

    Let $N_q'(p)$ be $N_q(p)$ with all don't-care corner points in $N(p)$ that are not 26-adjacent to q (marked ▣ in Figure 7.1.(a)) assigned black and all don't-care points in $N_q(p)$ that are 26-adjacent to q (marked ◻) assigned white;

return(Check-Table 1.1($N'_q(p)$)); (* Function Check-Table 1.1 is defined below. *)

end;

(* Test 1.2. All 6-neighbors of p are not black, but p has a black diagonal neighbor *)

if p has a black diagonal neighbor q then

begin

Let $N_q(p)$ be $N(p)$ as illustrated in Figure 7.1.(b) without p, q and without 4 points marked ○ and two corner (unmarked) points of $N(p)$ which are both 6-adjacent to q;

Let $N'_q(p)$ be $N_q(p)$ with all don't-care corner points in $N(p)$ that are not 26-adjacent to q (marked ▣ in Figure 7.1.(b)) assigned black and all don't-care points in $N_q(p)$ that are 26-adjacent to q (marked ⊠) assigned white;

return(Check-Table 1.2($N'_q(p)$)); (* Function Check-Table 1.2 is defined below. *)

end;

(* Test 1.3. All 18-neighbors of p are not black, but p has a black diametrical neighbor *)

if p has a black diametrical neighbor q then

begin

(* If any point marked ▣ in Figure 7.1.(c) is black or don't-care, p is non-simple *)

if ($\sum$ values of all points in $N(p)$ that are not 26-adjacent to q) > 0

then return (*FALSE*) else return (*TRUE*);

end;

(* Test 1.4. All 26-neighbors of p are not black *)

return (*FALSE*);

end;

(* C2($N_{18}(p)$) is true if and only if p is adjacent to only one white component in $N_{18}(p) - \{p\}$ *)

function C2($N_{18}(p)$): Boolean;

begin

(* Test 2.1. p has a white 6-neighbor *)

if p has a white 6-neighbor q then

begin

Let $N'_{18}(p)$ be $N_{18}(p)$ with all diagonal don't-care neighbors of p (marked ■ in Figure 7.1.(d)) assigned black;

return(Check-Table 2($N'_{18}(p)$)); (* Function Check-Table 2 is defined below. *)

end;

(* Test 2.2. All 6-neighbors of p are not white *)

return(*FALSE*);

end;

Declarations of Check-Table functions and lookup table descriptions

Check-Table 1.1 is a Boolean function. The input $N'_q(p)$ illustrated in Figure 7.1.(a) is as described in the algorithm. Check-Table 1.1 ($N'_q(p)$) returns *TRUE* if there is only one black component in $N'_q(p)$ adjacent to q for every assignment and *FALSE* otherwise. The function gives the output based on a lookup table *Table 1.1*. This table contains only $2^4 \times 2^8 \times 3^5 = 995,328$ entries, since 4 points marked ■ as well as 8 points marked ⊡ are either black or white, and only the remaining 5 points marked □ can have one of the three different colors (see Figure 7.1.(a)).

Check-Table 1.2 is a Boolean function similar to Check-Table 1.1. Note that Check-Table 1.2 will only be applied if none of 6-neighbors of p is black. The input $N'_q(p)$ illustrated in Figure 7.1.(b) is as described in the algorithm. Check-Table 1.2 ($N'_q(p)$) returns *TRUE* if there is only one black component in $N'_q(p)$ adjacent to q for every assignment and *FALSE* otherwise. The function gives the output based on a lookup table *Table 1.2*. This table contains only $2^6 \times 2^4 \times 2^2 \times 3^7 = 8,957,952$ entries, since (1) 6 corner points marked ■ are black or white, (2) 4 points marked ⊡ are black or white, (3) 2 points marked Δ are either white or don't-care, and (4) only the remaining 7 points marked □ can have one of the three different colors (see Figure 7.1.(b)).

Check-Table 2 is a Boolean function. The input $N'_{18}(p)$ illustrated in Figure 7.1.(d) is as described in the algorithm. Check-Table 2 ($N'_{18}(p)$) returns *TRUE* if there is only one white 6-component in $N'_{18}(p)$ 6-adjacent to q for every assignment and *FALSE* otherwise. The function gives the output based on a lookup table *Table 2*. This table contains only $2^{12} \times 3^5 = 995,328$ entries, since 12 points marked ▣ are black or white, and only the remaining 5 points marked □ can have one of the three different colors (see Figure 7.1.(d)).

Memory requirements

Note that by this approach the size of the lookup table which is necessary for the simplicity test has been greatly reduced. We need only three sub-tables which together have 10,948,608 entries. Thus the size of the three sub-tables together is about $10,948,608/8 \approx 1.4$ MBytes if one bit for each entry is used.

7.3 Verification

For C1, we try to find out whether there can be two black components in $N(p) - \{p\}$. There are two black components in $N(p) - \{p\}$ if and only if there are two black points in $N(p) - \{p\}$ that cannot be joined by a black 26-path in $N(p) - \{p\}$. We illustrate the idea of the proof for Test 1.1. In Test 1.1, p has a black 6-neighbor q. p fails Test 1.1 if and only if there is a black point in $N(p) - \{p\}$ which cannot be joined to q by a black 26-path in $N(p) - \{p\}$. If p fails Test 1.1 in $N'_q(p)$, then p fails Test 1.1 in $N(p)$, because an assignment for $N(p)$ is found which makes $C1(N(p))$ false. If p passes Test 1.1 in $N'_q(p)$, then by Lemmas 7.3.3, 7.3.4, and 7.3.5 p passes Test 1.1 for every possible assignment of $N(p)$.

Theorem 7.3.1 *The algorithm for the simplicity test is correct, i.e. the function simple($N(p)$) returns* TRUE *if and only if p is simple in $N(p)$.*

Proof. It follows from Propositions 7.3.2 and 7.3.6. □

Proposition 7.3.2 *The test C1 is correct, i.e. the function C1($N(p)$) returns* TRUE *if and only if p satisfies condition C1 in $N(p)$.*

Proof. First we show that Test 1.1 is correct. By Lemma 7.3.3, the color of 8 unmarked points in Figure 7.1.(a) does not influence the existence of only one black component in $N(p) - \{p\}$. Therefore, we do not need to consider these points in the input to function Check-Table 1.1.

By Lemma 7.3.4, making all corner points which are not 26-adjacent to q (marked ▣ in Figure 7.1.(a)) and are don't-care to be black does not influence the fact whether there is only one black component in $N(p) - \{p\}$ or not. Therefore, we do not need to consider these points to be don't-care in the input to function Check-Table 1.1.

By Lemma 7.3.5, making all points marked ▣ in Figure 7.1.(a) which are don't-care to be white does not influence the existence of only one black component in $N(p) - \{p\}$. Therefore, we do not need to consider these points to be don't-care in the input to function Check-Table 1.1.

Hence, Check-Table 1.1($N'_q(p)$) returns *TRUE* if and only if there is only one black component in $N(p) - \{p\}$ for every assignment to don't-care points.

Note that Test 1.2 applies only if all 6-neighbors of p are either white or don't-care. Therefore, by Lemma 7.3.5, if a point marked ○ in Figure 7.1.(b) is don't-care, it can be colored white. Hence we do not need to consider the points marked ○ in the input to function Check-Table 1.2. Taking this into consideration, the correctness of Test 1.2 can be obtained in the same way as for Test 1.1.

Note that Test 1.3 applies only if all 18-neighbors of p are either white or don't-care. Since q is a black corner of $N(p)$, if there exists a black point or a don't-care point x in $N(p) - N(q)$, we can assign x to be black and all don't-care points in $N(p) \cap N(q)$ to be white. With this assignment, q and x are two black points which cannot be joined by a 26-black path in $N(p) - \{p\}$, and therefore $N(p)$ fails C1. If all points in $N(p) - N(q)$ are white, then p is adjacent to only one black component for any possible assignment of $N(p)$, and therefore $N(p)$ passes C1.

Test 1.4 is trivially correct, since we can color all 26-neighbors of p white. □

The following fact was observed in [115].

Lemma 7.3.3 *Suppose p is a black point in $N(p)$. Then both of the following hold:*

1. *Let q be a black 6-neighbor of p. Then, refer to Figure 7.1.(a), the colors of the eight unmarked points do not influence the fact that $N(p)$ satisfies C1 or not.*

2. *Let r be a black diagonal neighbor of p. Then, refer to Figure 7.1.(b), the colors of the two unmarked points do not influence the fact that p satisfies C1 or not.*

Proof. (1) holds since the black point q is adjacent to p and every unmarked point is adjacent to q. Therefore, none of the unmarked points, if it were black, can belong to a different component than the component containing q. Similarly, (2) holds. □

Lemma 7.3.4 *Suppose p is a black point having a black neighbor q. Then changing the color of all don't-care corner points of $N(p)$ to black does not influence the fact that p satisfies C1 or not.*

Proof. If any corner of $N(p)$ is adjacent to some black points in $N(p) - \{p\}$, then by above Lemma 7.3.3 its color does not influence the simplicity of p. Suppose these is a don't-care corner x such that every point in $K(p,x) - \{p,x\}$ is either white or don't-care. We can let all points in $K(p,x) - \{p,x\}$ be white. If we also let x be black, then we obtain an assignment making p adjacent to two black components ($\{x\}$ and the one containing q). Thus p is non-simple. □

Lemma 7.3.5 *Suppose p is a black point having a black neighbor q. Then changing the color of all don't-care points in $N(p) \cap N(q)$ to white does not influence the fact that p satisfies C1 or not.*

Proof. If $N(p)$ with all don't-care points in $N(p) \cap N(q)$ assigned white fails C1, then there exists such an assignment that $N(p)$ fails C1. If $N(p)$ with all don't-care points in $K(p,q)$ assigned white passes C1, then every black or don't-care point which is not 26-adjacent to q is connected to q by a black 26-path in $N(p) - \{p\}$. Since every point in $K(p,q)$ is 26-adjacent to q, there can be only one 26-black component in $N(p) - \{p\}$ for every assignment to don't-care points. □

Proposition 7.3.6 *The test C2 is correct, i.e., function C2($N_{18}(p)$) returns* TRUE *if and only if p satisfies condition C2 in $N_{18}(p)$.*

Proof. First we show that Test 2.1 is correct. By Lemma 7.3.7, coloring all don't-care diagonal neighbors of p black does not influence the existence of only one white 6-component in $N_{18}(p)$ which is 6-adjacent to p. Therefore, we do not need to consider these points to be don't-care in the input to function Check-Table 2. Hence, Check-Table 2($N_{18}(p)$) returns *TRUE* if and only if there is only one white 6-component 6-adjacent to p in $N_{18}(p)$ for every assignment to don't-care points.

Test 2.2 is trivially true, since we can color all 6-neighbors of p black. □

Lemma 7.3.7 *Suppose p is a black point having a white 6-neighbor q. Then changing the color of all don't-care diagonal neighbors of p to black does not influence the fact that p satisfies C2 or not.*

Proof. If $N_{18}(p)$ with all don't-care diagonal neighbors of p assigned black fails C2, then there exists such an assignment that $N(p)$ fails C2. If $N_{18}(p)$ with all don't-care diagonal neighbors of p assigned black passes C2, then every white or don't-care 6-neighbor of p is connected to q by a white 6-path in $N_{18}(p)$. Since any assignment to the don't-care diagonal neighbors of p cannot influence the existence of only one white 6-component 6-adjacent to p, there is only one such component in $N_{18}(p)$. □

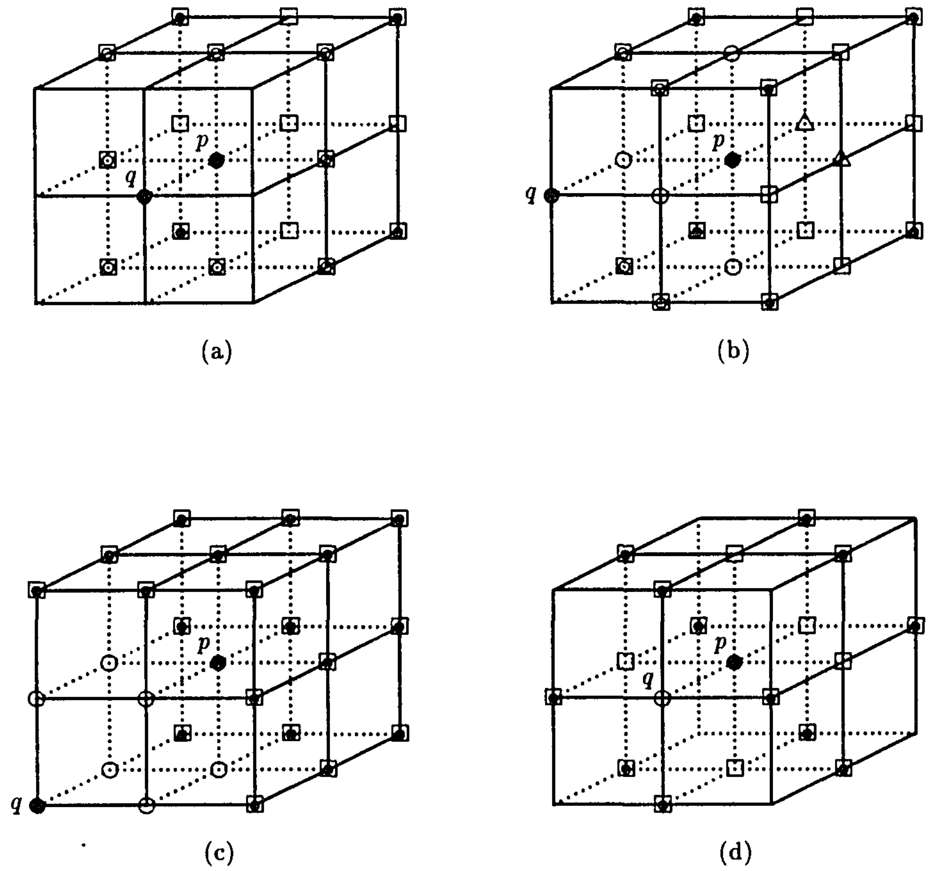


Figure 7.1: If any point marked ▣ is a don't-care point, it will be converted into a black point. If any point marked ◙ is a don't-care point, it will be converted into a white point. A point marked □ is either black, white or don't-care. Test 1.1 uses the figure in (a) where the eight unmarked points are ignored. Test 1.2 uses the figure in (b) where the four points marked ○ are white points, two points marked △ are either white or don't-care, and the two unmarked points are ignored. Test 1.3 uses the figure in (c). Test 2.1 uses the figure in (d) where the six points marked ○ are either white or can be assigned white.

Chapter 8

Conclusion

8.1 The four objectives in this thesis

In this thesis, we have introduced two fully parallel thinning algorithms for generating skeletons like medial-faces and medial-lines respectively. These algorithms can be applied to different applications. If the original image contains only bars, trees, lines, etc., then the medial-line algorithm is appropriate. If the original image contains planes, walls, etc., then the medial-face algorithm is appropriate. For preserving the geometry of a thin line, we only need to preserve any black point with only one black neighbor. However, it is not clear how to preserve the geometry of a thin face. In this thesis, we proposed a possible definition for this problem. Further study is necessary.

The main results of this thesis are the 3D connectivity preservation tests in Chapter 4. Using such 3D tests, we only need to check the configurations in a unit lattice cube for determining the connectivity preservation of a 3D thinning algorithm. Thus, when we try to design advanced 3D thinning algorithms, we only need to worry about the situations which occur in such a small area. The results in this chapter cover all 3D (26,6), (18,6), (6,26), and (6,18) images. All of them are *conventional* images, i.e., images embedded in regular grids.

On the basis of 3D connectivity preservation tests, we are able to prove that both of our 3D thinning algorithms in Chapter 3 preserve connectivity in Chapter 5. In fact, the designs of our thinning algorithms were motivated by the results in Chapter 4. In Chapter 5, other existing thinning algorithms were verified by using the properties of the 3D connectivity preservation tests – some of them preserve connectivity and some do not. Other possible 3D results for connectivity preservation were also discussed in Chapter 5.

For verifying the connectivity preservation of a 3D thinning algorithm, we only need to consider a finite number of configurations. Thus, we are able to use computers to generate the proofs. The straightforward design of such a program is: (1) list all possible such configurations, and (2) let computers check every configuration in the list. This approach has been adopted for establishing 2D computerized tests in Chapter 6. In the same chapter, 3D computerized tests based on this approach are also established. For our fully parallel thinning algorithms (both of them contain less than 50 deleting templates), the 3D computerized tests take about one hour on a 33 MHz 386 PC/AT to show that both of them preserve connectivity. Such a program needs to merge up to four deleting templates of a testing thinning algorithm, i.e., this program has the time complexity $O(n^4)$. We established a fast test to overcome the problem of merging four templates for 6-subiteration thinning algorithms. The time complexity of the fast test is $O(n)$.

The simplicity of a deleting template of a 3D thinning algorithm can be determined based on its core. Since each point in a template core can be either a black point, a white point or a don't-care point, there are totally 3^{26} different cores. A complete simple table needs 318 GBytes to maintain every core (one bit for each entry). In Chapter 7, we used several smaller tables for checking the simplicity of any template core in 3D (26,6) images. The total size of all these smaller tables is 1.4 MBytes (one bit for each entry).

8.2 Future research

For the 2D case, it is known that thinning algorithms on images using the (4,8) adjacency pair may generate skeletons look "closer" to original objects than on images using the (8,4) adjacency pair. There are several thinning algorithms for 3D (26,6) images. It may be interesting to design thinning algorithms for 3D (6,26) images and see whether the skeletons look closer to the original objects. Thinning algorithms for 3D (18,6) and (6,18) may also be topics for future research.

Our 3D connectivity preservation tests are for images embedded in regular grids. There are some images that are embedded in irregular grids, for example, 3D (14,14) images and 3D (12,12) images. The advantage of these two 3D adjacency pairs is that the black adjacency and the white adjacency are the same. Thus, we only need to consider one adjacency relation. Since the connectivity preservation tests for 2D (6,6) images (which is embedded in irregular grids) have been proposed by Hall, it might be interesting to establish 3D connectivity preservation tests for these 3D images.

For an object of an image, its skeleton after the application of thinning is contained in the original object. We only need to consider the deletion operation in thinning, i.e., only black points can be changed to white points. However, the result after the application of *shrinking* is *not* necessarily contained in the original object. For example, consider a 2D image containing a big thick black circle. Intuitively, the result of thinning is a 2D image containing a big thin black circle, but the result of shrinking is a 2D image containing a *small* thin black circle. Clearly, it should be easier to recognize the result by shrinking. However, we need consider not only the deletion but also the *augmentation* (i.e., change white points to black points) in shrinking. One possible topic for future research is to find 2D and 3D connectivity preservation tests for shrinking algorithms.

Bibliography

[1] T. Agui and K. Iwata. Topological structure analysis of pictures by digital computer. *Systems, Computers and Control*, **10**:333–340, 1979.

[2] J. C. Alexander and A. I. Thaler. The boundary count of digital pictures. *Journal of the ACM*, **18**(1):105–112, 1971.

[3] H. M. Alnuweiri and V. K. Prasanna. Parallel architectures and algorithms for images component labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**:1014–1034, 1992.

[4] C. Arcelli. Pattern thinning by contour tracing. *Computer Graphics and Image Processing*, **17**:130–144, 1981.

[5] C. Arcelli, L. P. Cordella, and S. Levialdi. Parallel thinning of binary pictures. *Electronic Letters*, **11**:148–149, 1975.

[6] C. Arcelli, L. P. Cordella, and S. Levialdi. From local maxima to connected skeletons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-3**:134–143, 1981.

[7] C. Arcelli and G. S. di Baja. On the sequential approach to medial line transformation. *IEEE Transactions on Systems, Man and Cybernet*, **SMC-8**:139–144, 1978.

[8] C. Arcelli and G. S. di Baja. A thinning algorithm based on prominence detection. *Pattern Recognition*, **13**:225–235, 1981.

[9] C. Arcelli and G. S. Di Baja. A width-independent fast thinning algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-7**:463–474, 1985.

[10] C. Arcelli and S. Levialdi. Parallel shrinking in three dimensions. *Computer Graphics and Image Processing*, **1**:21–30, 1972.

[11] E. Artzy, G. Frieder, and G. T. Herman. The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm. *Computer Graphics and Image Processing*, **15**:1–24, 1981.

[12] H. H. Atkinson, I. Gargantini, and M. V. S. Ramanath. Improvements to a recent 3D-border algorithm. *Pattern Recognition*, **18**:215–226, 1985.

[13] H. H. Atkinson, I. Gargantini, and W. T. R. S. Counting regions, holes, and their nesting level in time proposional to the border. *CVGIP*, **29**:196–215, 1985.

[14] M. J. Biggar and A. G. Constantinides. Thin line coding techniques. In *Digital Signal Processing*, pages 471–475. Elservier Science Publisher B. V. (North-Holland), 1987.

[15] A. D. Brink. Gray-level thresholding of images using a correlation criterion. *Pattern Recognition Letters*, **9**:335–341, 1989.

[16] D. W. Capson. An improved algorithm for the sequential extraction of boundaries from a raster scan. *CVGIP*, **28**:109–125, 1984.

[17] K. R. Castleman. *Digital Image Processing.* Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1979.

[18] C. S. Chen and W. H. Tsai. A new fast one-pass thinning algorithm and its parallel hardware implementation. *Pattern Recognition Letters*, **11**:471–477, 1990.

[19] Y. S. Chen and W. H. Hsu. A comparison of some one-pass parallel thinnings. *Pattern Recognition Letters*, **11**:35–41, 1990.

[20] R. T. Chin, H. K. Wan, D. L. Stover, and R. D. Iverson. A one-pass thinning algorithm and its parallel implementation. *CVGIP*, **40**:30–40, 1987.

[21] I. Cseke and Z. Fazekas. Comments on gray-level thresholding of images using a correlation criterion. *Pattern Recognition Letters*, **11**:709–710, 1990.

[22] E. R. Davies and A. P. N. Plummer. Thinning algorithms: A critique and a new methodology. *Pattern Recognition*, **14**:53–63, 1981.

[23] E. S. Deutsch. Comments on a line thinning scheme. *Computing Journal*, **12**:412, 1969.

[24] E. R. Dougherty. *An introduction to morphological image processing*. SPIE optical engineering press, 1992.

[25] M. J. E. Golay. Hexagonal parallel pattern transformations. *IEEE Transactions on Computers*, **C-18**:733–740, 1969.

[26] W. X. Gong and G. Bertrand. A note on "Thinning of 3-D images using the safe point thinning algorithm (SPTA)". *Pattern Recognition Letters*, **11**:499–500, 1990.

[27] W. X. Gong and G. Bertrand. A simple parallel 3D thinning algorithm. In *IEEE International Conference on Pattern Recognition*, pages 188–190, 1990.

[28] S. B. Gray. Local properties of binary images in two dimensions and three dimensions. In *Information International*, 1970.

[29] S. B. Gray. Local properties of binary images in two dimensions. *IEEE Transactions on Computers*, **C-20**:551–561, 1971.

[30] Z. Guo and R. W. Hall. Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, **32**:359–373, 1989.

[31] K. J. Hafford and K. Preston, Jr. Three-dimensional skeletonization of elongated solids. *CVGIP*, **27**:78–91, 1984.

[32] R. W. Hall. Connectivity preserving parallel operators in 2D and 3D images. In *the Proceedings of 1992 SPIE Conference on Vision Geometry*, pages 172–183, Boston, MA, 1992.

[33] R. W. Hall. Fast parallel thinning algorithms: Parallel speed and connectivity preservation. *Communications of the ACM*, **32**:124–131, 1992.

[34] R. W. Hall. Tests for connectivity preservation for parallel reduction operators. *Topology and Its Applications*, **46**:199–217, 1992.

[35] G. T. Herman. On topology as applied to image analysis. *CVGIP*, **52**:409–415, 1990.

[36] G. T. Herman and D. Webster. A topological proof of a surface tracking algorithm. *CVGIP*, **23**:162–177, 1983.

[37] C. M. Holt, A. Stewart, M. Clint, and R. H. Perrott. An improved parallel thinning algorithm. *Communications of the ACM*, **30**:156–160, 1987.

[38] B. K. Jang and R. T. Chin. Analysis of thinning algorithms using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-12**:541–551, 1990.

[39] B. K. Jang and R. T. Chin. One-pass parallel thinning: Analysis, Properties, and Quantitative Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**:1129–1140, 1992.

[40] J. F. Jenq and S. Sahni. Serial and parallel algorithms for the medial axia transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**:1218–1224, 1992.

[41] L. Ji and J. Piper. Fast homotopy – Preserving skeletons using methematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-14**:653–664, 1992.

[42] S. Kawai. On the topology preservation property by local parallel operations. *Computer Graphics and Image Processing*, **19**:265–270, 1982.

[43] S. Kawai. Topology quasi-preservation at local parallel operations. *CVGIP*, **23**:353–365, 1983.

[44] E. Khalimsky, R. Kopperman, and P. R. Meyer. Computer graphics and connected topologies on finite ordered sets. *Topology and Its Applications*, **36**:1–17, 1990.

[45] R. Klette. The m-dimensional grid point space. *CVGIP*, **30**:1–12, 1985.

[46] T. Y. Kong. 2D computerized tests. an unpublished program, 1991.

[47] T. Y. Kong. A digital fundamental group. *Computers and Graphics*, **13**(2):159–166, 1989.

[48] T. Y. Kong. On the problem of determining whether a parallel reduction operator for n-dimensional binary images always preserves topology. In *the Proceedings of 1993 SPIE Conference on Vision Geometry*, pages 69–77, Boston, MA, 1993.

[49] T. Y. Kong and A. W. Roscoe. Characterizations of simply-connected finite polyhedra in 3-space. *Bulletin of London Mathematics Society*, **17**:575–578, 1985.

[50] T. Y. Kong and A. W. Roscoe. Continuous analogous of axiomatized digital pictures. *CVGIP*, **29**:60–86, 1985.

[51] T. Y. Kong and A. W. Roscoe. A theory of binary digital pictures. *CVGIP*, **32**:221–243, 1985.

[52] T. Y. Kong, A. W. Roscoe, and A. Rosenfeld. Concepts of digital topology. *Topology and Its Applications*, **46**:219–262, 1992.

[53] T. Y. Kong and A. Rosenfeld. Digital topology: introduction and survey. *CVGIP*, **48**:357–393, 1989.

[54] R. Kopperman, P. R. Meyer, and R. G. Wilson. A Jordan surface theorem for three-dimensional digital spaces. *Discrete and Computational Geometry*, **6**:155–161, 1991.

[55] E. H. Kronheimer. A note on alternative digital topologies. *Topology and Its Applications*, **46**:269–277, 1992.

[56] E. H. Kronheimer. The topology of digital images. *Topology and Its Applications*, **46**:279–303, 1992.

[57] P. C. K. Kwok. A thinning algorithm by contour tracing. *Communications of the ACM*, **31**:1314–1324, 1988.

[58] L. Latecki. Well-composed sets. To appear in CVGIP: Image Understanding.

[59] L. Latecki and C. M. Ma. An algorithm for 3D simplicity test. Submitted to CVGIP: Image Understanding.

[60] C. N. Lee, T. Poston, and A. Rosenfeld. Winding and Euler numbers for 2D and 3D digital images. *CVGIP: Graphical Models and Image Processing*, **53**:522–537, 1991.

[61] C. N. Lee and A. Rosenfeld. Note: Simple connectivity is not locally computable for connected 3D images. *CVGIP*, **51**:87–95, 1990.

[62] S. Levialdi. On shrinking binary picture patterns. *Communications of the ACM*, **15**:7–10, 1972.

[63] B. Li and C. Y. Suen. A knowledge-based thinning algorithm. *Pattern Recognition*, **24**:1211–1221, 1991.

[64] C. M. Ma. A fast 3D parallel thinning algorithm for generating medial faces. To appear in Pattern Recognition Letters.

[65] C. M. Ma. Topology preservation on 3D images. In *the Proceedings of 1993 SPIE Conference on Vision Geometry*, pages 201–207, Boston, MA, 1993.

[66] C. M. Ma. On topology preservation in 3D thinning. *CVGIP: Image Understanding*, **59-3**:328–339, 1994.

[67] C. M. Ma and M. Sonka. A fast 3D parallel thinning algorithm and its applications. In preparing.

[68] S. A. Mahmoud, I. AbuHaiba, and R. J. Green. Skeletonization of Arabic characters using clustering based skeletonization algorithm (CBSA). *Pattern Recognition*, **24**:453–464, 1991.

[69] G. Malandain and G. Bertrand. Fast characterization of 3D simple points. In *IEEE International Conference on Pattern Recognition*, pages 232–235, 1992.

[70] M. P. Martínes-Pérez, J. Jiménez, and J. L. Navalón. A thinning algorithm based on contours. *CVGIP*, **39**:186–201, 1987.

[71] A. McAndrew and C. F. Osborne. Some new results in digital topology for hexagonal grids. In *Digital Image Computing; Techniques and Applications, (DICTA '91), Melbourne*, pages 546–553, 1991.

[72] M. Milgram and T. S. Pierre. Boundary detection and skeletonization with a massively parallel architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**:74–77, 1990.

[73] A. Montanvert, P. Meer, and A. Rosenfeld. Hierarchical image analysis using irregular tessellations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-13**:307–316, 1991.

[74] 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, College Park, MD, 1981.

[75] D. G. Morgenthaler and A. Rosenfeld. Surfaces in three-dimensional digital images. *Information and Control*, **51**:227–247, 1981.

[76] J. Mukherjee, B. N. Chatterji, and P. P. Das. Thinning of 3D images using the safe point thinning algorithm (SPTA). *Pattern Recognition Letters*, **10**:167–173, 1989.

[77] J. Mukherjee, P. P. Das, and B. N. Chatterji. On connectivity issues of ESPTA. *Pattern Recognition Letters*, **11**:643–648, 1990.

[78] J. P. Mylopoulos and T. Pavilidis. On the topological properties of quantized spaces. I. The notion of dimension. *Journal of the ACM*, **18**:239–246, 1971.

[79] J. P. Mylopoulos and T. Pavilidis. On the topological properties of quantized spaces. II. Connectivity and order of connectivity. *Journal of the ACM*, **18**:247–254, 1971.

[80] N. J. Naccache and R. Shinghal. SPTA: A proposed algorithm for thinning binary patterns. *IEEE Transactions on Systems, Man and Cybernet*, **SMC-14**:409–418, 1984.

[81] A. Nakamura and K. Aizawa. On the recognition of properties of three-dimensional pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-7**:708–713, 1985.

[82] V. Neumann-Lara and R. G. Wilson. Digital Jordan curves – a graph-theoretical approach to a topological theorem. *Topology and Its Applications*, **46**:263–268, 1992.

[83] C. W. Niblack, P. B. Gibbons, and D. W. Gapson. Generating skeletons and centerlines from the distance transform. *CVGIP: Graphical Models and Image Processing*, **54**:420–437, 1992.

[84] H. Ogawa and K. Taniguchi. Thinning and stroke segmentation for handwritten Chinese character recognition. *Pattern Recognition*, **15**:299–308, 1982.

[85] L. O'Gorman. $k \times k$ thinning. *CVGIP*, **51**:195–215, 1990.

[86] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernet*, **SMC-9**:62–66, 1979.

[87] S. Pal. Fuzzy skeletonization of an image. *Pattern Recognition Letters*, **10**:17–23, 1989.

[88] T. Pavlidis. An asynchronous thinning algorithms. *CVGIP*, **20**:133–157, 1982.

[89] K. Preston, Jr. Multidimensional logical transformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-5**:539–554, 1983.

[90] K. Qian and P. Bhattacharya. Determining holes and connectivity in binary images. *Computers and Graphics*, **16**:283–288, 1992.

[91] C. V. K. Rao, B. Prasada, and K. R. Sarma. A parallel shrinking algorithm for binary patterns. *Computer Graphics and Image Processing*, **5**:265–270, 1976.

[92] G. M. Reed. On the characterization of simple closed surfaces in three-dimensional digital images. *CVGIP*, **25**:226–235, 1984.

[93] S. Riazanoff, B. Cervelle, and J. Chorowics. Parameterisable skeletonization of binary and multilevel images. *Pattern Recognition Letters*, **11**:25–33, 1990.

[94] C. Ronse. A topological characterization of thinning. *Theoretical Computer Science*, **43**:31–41, 1986.

[95] C. Ronse. Minimal test patterns for connectivity preservation in parallel thinning algorithm for binary images. *Discrete Applied Mathematics*, **21**:67–79, 1988.

[96] A. Rosenfeld. Connectivity in digital pictures. *Journal of the ACM*, **17**(1):146–160, 1970.

[97] A. Rosenfeld. Arcs and curves in digital pictures. *Journal of the ACM*, **20**(1):81–87, 1973.

[98] A. Rosenfeld. Adjacency in digital pictures. *Information and Control*, **26**:24–33, 1974.

[99] A. Rosenfeld. A characterization of parallel thinning algorithms. *Information and Control*, **29**:286–291, 1975.

[100] A. Rosenfeld. A converse of the digital Jordan curve theorem for digital curves. *Information and Control*, **29**:292–293, 1975.

[101] A. Rosenfeld. Digital topology. *The American Mathematics Monthly*, **86**:621–630, 1979.

[102] A. Rosenfeld. Fuzzy digital topology. *Information and Control*, **40**:76–87, 1979.

[103] A. Rosenfeld. Three-dimensional digital topology. *Information and Control*, **50**:119–127, 1981.

[104] A. Rosenfeld. On connectivity properties of grayscale pictures. *Pattern Recognition*, **16**:47–50, 1983.

[105] A. Rosenfeld. The fuzzy geometry of image subsets. *Pattern Recognition Letters*, **2**:311–317, 1984.

[106] A. Rosenfeld. Distances between fuzzy sets. *Pattern Recognition Letters*, **3**:229–233, 1985.

[107] A. Rosenfeld. Continuous functions on digital pictures. *Pattern Recognition Letters*, **4**:177–184, 1990.

[108] A. Rosenfeld. Fuzzy rectangles. *Pattern Recognition Letters*, **11**:677–679, 1990.

[109] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*, volume 1. Academic Press, New York, 2nd edition edition, 1982.

[110] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*, volume 2. Academic Press, New York, 2nd edition edition, 1982.

[111] A. Rosenfeld, T. Y. Kong, and A. Y. Wu. Digital surfaces. *CVGIP: Graphical Models and Image Processing*, **53**:305–312, 1991.

[112] A. Rosenfeld and R. A. Melter. Digital geometry. *Mathematical Intelligence*, **11**:69–72, 1989.

[113] A. Rosenfeld and J. L. Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM*, **13**(1):471–494, 1966.

[114] D. Rutovitz. Pattern recognition. *Journal of the Royal Statistical Society, Series A*, **129**:504–530, 1966.

[115] P. K. Saha, B. B. Chaudhuri, and D. Dutta Majumder. Principles and algorithms for 2D and 3D shrinking. Technical Report TR/KBCS/2/91, N.C.K.B.C.S. Library, Indian Statistical Institute, Calcutta, India, 1991.

[116] J. Sklansky and D. F. Kibler. A theory of nonuniformly digitized binary pictures. *IEEE Transactions on Systems, Man and Cybernet*, **SMC-6**:637–647, 1976.

[117] J. H. Sossa. An improved parallel algorithm for thinning digital patterns. *Pattern Recognition Letters*, **10**:77–80, 1989.

[118] R. Stefanelli and A. Rosenfeld. Some parallel thinning algorithms for digital pictures. *Journal of the ACM*, **18**:255–264, 1971.

[119] L. N. Stout. Two discrete forms of the Jordan curve theorem. *The American Mathematics Monthly*, **95**:332–336, 1988.

[120] S. Suzuki and K. Abe. Topological structural analysis of digitized binary images by contour following. *CVGIP*, **30**:32–46, 1985.

[121] L. Thurfjell, E. Bengtsson, and B. Nordin. A new three-dimensional connected components labeling algorithm with simultaneous object feature extraction capability. *CVGIP: Graphical Models and Image Processing*, **54**:357–364, 1992.

[122] J. I. Toriwaki, S. Yokoi, T. Yonekura, and T. Fukumura. Topological properties and topology-preserving transformation of a three-dimensional binary picture. In *IEEE International Conference on Pattern Recognition*, pages 414–419, 1982.

[123] G. Tourlakis and J. Mylopoulos. Some results in computation topology. *Journal of the ACM*, pages 439–455, 1732.

[124] Y. F. Tsao and K. S. Fu. A parallel thinning algorithm for 3D pictures. *Computer Graphics and Image Processing*, **17**:315–331, 1981.

[125] Y. F. Tsao and K. S. Fu. A 3D parallel skeletonwise thinning algorithm. In *IEEE Pattern Recognition and Image Processing Conference*, pages 678–683, 1982.

[126] J. Udupa, S. N. Srihari, and G. T. Herman. Boundary detection in multidimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-4**:41–50, 1982.

[127] J. K. Udupa and V. G. Ajjanagadde. Boundary and object labelling in three-dimensional images. *CVGIP*, **51**:355–369, 1990.

[128] J. S. Weszka. A survey of threshold selection techniques. *Computer Graphics and Image Processing*, **7**:259–265, 1978.

[129] W. Xu and C. X. Wang. CGA: A fast thinning algorithm implemented on a sequential computer. *IEEE Transactions on Systems, Man and Cybernet*, **SMC-17**:847–851, 1987.

[130] S. S. Yu and W. H. Tsai. A new thinning algorithm for gray-scale images by the relaxation technique. *Pattern Recognition*, **10**:1067–1076, 1990.

[131] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, **27**:236–239, 1984.