TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

DEPARTMENT OF MATHEMATICS

MASTER THESIS

Topology Preserving Skeletonization of 2D and 3D Binary Images

by

Tesfaye Admassie Aberra

November 2004

Supervised by
Dr. Katja Schladitz
Prof. Dr. Jurgen Franke

Contents

Acknowledgment

First and foremost, I would like to thank my advisors, Dr. Katja Schladitz and Prof. Dr. Jurgen Franke, who made this thesis possible with their persistent guidance, patience and encouragement. The financial support I got from the University of Kaiserslautern and the Institute for Industrial Mathematics (ITWM) for the entire duration of my study is also gratefully acknowledged.

The work of this Thesis was done at the ITWM. I would like to thank the staff of the Institute and students for helping me in times of difficulty.

1 Introduction

Skeletonization has been a part of image processing for a wide variety of applications. The usefulness of reducing images to their skeletons can be attributed to the need to process a reduced amount of data as well as to the fact that shape analysis can be more easily made on skeletons. This has been used for a long time in two-dimensional image processing (Borgefors et al 1998).

Recent advances in three-dimensional imaging technology and availability of sophisticated computers has introduced 3D image processing techniques as a solution to everyday life as well as specialized problems. These applications include luggage-scanning systems, quality control systems for factory products, medical image processing, automated navigation and animation generation. Due to inherent large amount of data associated with 3-D images, it is desirable to deal with certain structural features of interest of imaged objects. Therefore representing objects by their skeleton is a very efficient way to deal with 3D data with respect to memory usage and data compression while highlighting structural features of the objects at hand.

Different skeletonization techniques are used to get a skeleton. These techniques can be classified into thinning, distance transform and voronoi methods. Depending on the type of skeleton that we wanted, we can apply any of the methods. Some of them preserve topology while others preserve geometry of the original object.

We are interested in finding a topology and geometry preserving skeletonization algorithm based on the thinning methods for both 2D and 3D digital images. A number of algorithms have already been discussed in the literature. But most of the algorithms do not preserve topology. Therefore, we are proposing a new sequential thinning algorithm which preserves the topology and geometry of the image for both 2D and 3D images.

Therefore, this thesis is organized as follows. Chapter 3 deals with definitions of basic concepts that are used throughout the thesis. It defines such concepts as topology, adjacency pairs, Euler number and topology preserving.

Chapter 4 deals with defintions of skeletonization, essential characteristics of skeletons and similar works by different authors. Chapter 5 deals with different techniques for skeletonization: thinning, distance transform and voronoi methods with detailed treatment of thinning methods. Also examples of thinning algorithms both in 2D and 3D images are discussed in this chapter.

In chapter 6, we discussed our new sequential thinning algorithms for 2D and 3D skeletonization. Chapter 7 deals with the main results of application of our algorithm for both 2D and 3D images. An attempt is made to present results for test bodies considered as well as sample image data. Finally, chapter 8 focuses on the discussion of main results of the thesis.

2 Preliminaries

The only sets which can be handled on computers are discrete or digital sets. Therefore, to represent continuous objects in computers, they must be discretized, i.e., they must be approximated by discrete objects. Discretization is performed in image processing where the continuous image is represented by a discrete set of pixels. It is represented as arrays of values.

It is clear that, no discrete model can exhibit all relevant features of a continuous original. Therefore, one has to accept compromises. The compromise chosen depends on the specific application. A branch of mathematics called Digital topology is an attempt to evaluate the price one has to pay for discretization. A digital topology can be defined in terms of a graph which is obtained when a neighborhood relation is introduced into the digital sets. Neighbourhood relations allow us to define connectivity of sets. Digital topology is the theoretical basis for understanding properties of sets in images (Eckhardt and Latecki)

n-dimensional digital images can be thought of as n-dimensional arrays of values such that each array element represents exactly one image element. Array elements of 2D images are called pixels (picture elements) while array elements of 3D images are called voxels (volume elements). A digital image is called a binary image if its values are either 1 (foreground) or 0 (background) (Lohman 1998).

In two-dimensional images, each pixel has four immediate neighbors whose address differ in at most one coordinate. In addition there are four diagonal neighbors whose coordinates differ in both places.

In image analysis object and background pixels are usually endowed with different adjacencies (neighborhoods) in order to ensure a digital Jordan surface theorem (Ohser et al (2003)). Jordan theorem states that in continuous images, a closed continuous curve always separates the interior from the exterior part of the object. But the discrete version of the Jordan theorem does not hold if we assume the same connectivity for both foreground and background. Therefore, a simple way to avoid this problem is to adopt to different connectivities for the background and foreground. It has been shown that the discrete version of the Jordan theorem holds in 2D if we use the adjacency pairs (4,8) and (8,4). In addition Ohser, Nagel and Schladitz (2003) have shown that the discrete version of the Jordan theorem holds for the two self-complementary adjacency pairs (6.1,6.1) and (6.2,6.2). In this notation, the first component of a pair indicates the number of neighbors a pixel is connected to if this pixel belongs to the foreground and the second component denotes the number of used connections to neighbors if the pixel belongs to the complement(background). Therefore, in two-dimensions, connectivity is defined as 4-, 8-, 6.1- and 6.2- connectivity. As can be seen in Table 1, the point P_0 is:

- 8-adjacent to the neighbors $P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8$.
- 4-adjacent to the neighbors P_1, P_3, P_5, P_7.
- 6.1-adjacent to the neighbors $P_1, P_2, P_3, P_5, P_6, P_7$.

- 6.2-adjacent to the neighbors $P_1, P_3, P_4, P_5, P_7, P_8$.

Table 1: Adjacency relationship of neighbors of a point

P_2	P_1	P_8
P_3	P_0	P_7
P_4	P_5	P_6

Any of the neighborhood configurations may be adopted for analysis of any particular problem. But only the four adjacency pairs shown in Table 2 are valid since they satisfy the discrete version of the Jordan curve theorem.

Table 2: Adjacency pairs for foreground and background pixels for 2D images

Foreground	Background
4	8
8	4
6.1	6.1
6.2	6.2

A digital 3D image defined on a 3D cubic grid is a straightforward extension of 2D image to the 3D space. A 3D digitized image is represented by a 3D array $F = f_{ijk}$, where i, j and k are integers and f_{ijk} represent voxel value located on the i^{th} row and j^{th} column on the k^{th} plane.

Now, let us define what a neighborhood is in a 3D digitized image.

For each voxel x=i,j,k, three kinds of neighborhoods, the 6-neighborhood $N^{(6)}(x)$, 18-neighborhood $N^{(18)}(x)$ and the 26-neighborhood $N^{(26)}(x)$are defined as follows and are shown in the Figure 1 (Serra 1982).

$N^{(6)}(x)$=$\{(p, q, r) : |p - i| + |q - j| + |r - k| = 1\}$

$N^{(18)}(x)$=$\{(p, q, r) : 0 < |p - i|^2 + |q - j|^2 + |r - k|^2 \leq 2\}$

$N^{(26)}(x)$=$\{(p, q, r) : max(|p - i|, |q - j|, |r - k|) = 1\}$

We do not use the 18-neighborhood for our purpose, it is mainly used in computer graphics. Our main objective is consistency for the Euler number (see Ohser, Nagel, and Schladitz (2003)).

Any voxel p in the 6-neighborhood of a voxel x shares one face, in the 18-neighborhood shares at least one edge and in the 26-neighborhood shares at least one vertex with the voxel x.

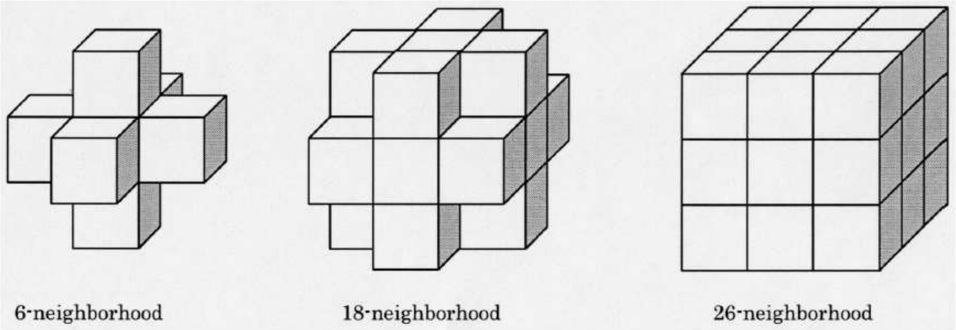

Figure 1: The three kinds of neighbors in 3D pictures: adapted from Toriwaki and Yonekura 2002

As in the 2D case, connectivity is defined in terms of adjacency pairs. The same problem that we discussed in the 2D case arises also for the 3d case regarding the discrete version of the Jordan theorem. But it has been shown that it holds for the adjacency pairs - (6,26),(26,6),(18,6), and (6,18) where the first component represents connectivity of the foreground while the second component represent connectivity for the background. In addition to these four adjacency pairs Ohser, Nagel and Schladitz (2003) have shown that there are two more adjacency pairs based on tessellations of the unit cell (Figure 2). These are (14.1,14.1) and (14.2,14.2) which are self-complementary, i.e., foreground and background components are treated with the same neighborhood. The 18-neighborhood is modified to yield the two 14-neighborhoods which are defined as follows.

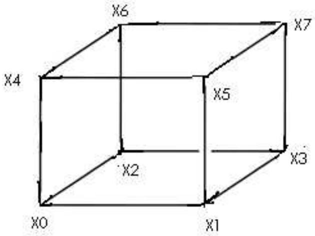

Figure 2: The unit cell and its vertices

- The 14.1 neighborhood (shown in Figure 3)is defined by considering the tessellation of the unit cell into the following 6 tetrahedrons $G_1, ..., G_6$ which are convex hulls of the vertices of the unit cell $(x_0, ..., x_7)$:
  $G_1 = Conv(\{x_0, x_1, x_3, x_7\})$, $G_2 = Conv(\{x_0, x_1, x_5, x_7\})$, $G_3 = Conv(\{x_0, x_2, x_3, x_7\})$,
  $G_4 = Conv(\{x_0, x_4, x_5, x_7\})$, $G_5 = Conv(\{x_0, x_4, x_6, x_7\})$, $G_6 = Conv(\{x_0, x_2, x_6, x_7\})$.

- The 14.2 neighborhood (shown in Figure 3) is defined by considering the tessellation of the unit cell into the following 6 tetrahedrons $G_1, ..., G_6$ which are convex hulls of the vertices of the unit cell $(x_0, ..., x_7)$:

$$G_1 = Conv(\{x_0, x_1, x_3, x_5\}), G_2 = Conv(\{x_0, x_2, x_3, x_7\}), G_3 = Conv(\{x_0, x_2, x_4, x_7\}),$$
$$G_4 = Conv(\{x_0, x_3, x_5, x_7\}), G_5 = Conv(\{x_0, x_4, x_5, x_7\}), G_6 = Conv(\{x_2, x_4, x_6, x_7\}).$$

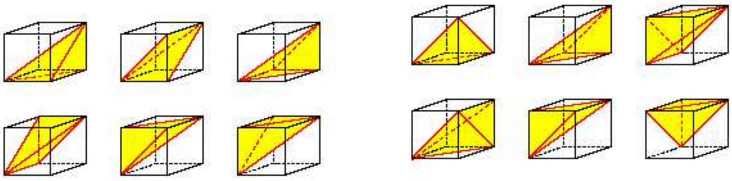

Figure 3: The tessellation of the unit cell defining(from left to right) the 14.1 and 14.2 neighborhood

Any of the neighborhood configurations may be adopted for analysis of any particular problem. But only the six adjacency pairs shown in Table 3 are possible since they satisfy the discrete version of the Jordan theorem in 3D.

Table 3: Adjacency pairs for foreground and background pixels in 3D

Foreground	Background
6	26
26	6
18	6
6	18
14.1	14.1
14.2	14.2

Object points in an image may be grouped as a set of connected components. An object component may contain cavities and tunnels. A cavity is a 3D analog of a hole in 2D signifying the existence of a component of the background points surrounded by an object component. A tunnel, on the other hand, does not signify a new background component. However, an object component contains a tunnel if it contains a solid handle or a hollow torus.

A local operation is said to be topology preserving, if the topology of an input image does not change by its execution. In other words, an operator O for a digital picture is topology preserving if, by its execution, none of the following occurs (Tariwaki et al 2002):

- generation and extinction of connected components,
- generation and extinction of tunnels (handles),
- generation and extinction of cavities.

Topology preserving can also be defined in terms of the Euler number. The Euler number describes the connectivity of components and is a topologically invariant property of an object. It is calculated by using the Euler Poincare formula:

Euler number = number of connected components - number of tunnels + number of cavities in 3D and

Euler number = number of connected components - number of holes in 2D. It can also define in terms of the number of vertices, edges and faces as :

Euler Number = number of vertices - number of edges + number of faces.

Therefore, two objects are topologically equivalent if their Euler number is the same. In the subsequent sections we will use the Euler number to study topology preserving skeletonization.

3 Skeletonization

Skeletonization is the process of reducing foreground regions in a binary image to skeletal remnant that preserves the connectivity of the original image while throwing away most of the original foreground pixels/voxels. In other words, given an input binary image, skeletonization changes non-skeletal object pixels/voxels into background pixels/voxels.

The results of skeletonization processes are skeletons. Various definitions of skeletons are provided in the literature by different authors. A very illustrative definition of the skeleton is given by the gross-fire or wavefront propagation analogy by Soille (1999): the body of an object is set on fire and is propagating at uniform speed within the body, the skeleton is the set of points where the fire or wavefronts meet. For example, the skeleton of a disk is its center because all wavefronts meet simultaneously at its center. The skeleton of a square corresponds to its diagonals, etc.

A more formal definition of the skeleton is the locus of the centers of all maximal inscribed hyper-spheres (i.e, discs and balls in 2D and 3D, respectively). An inscribed hyper-sphere is maximal if it is not covered by any other inscribed hyper-sphere.

The extension of the notion of a skeleton to discrete sets is not straightforward. Notions such as wave propagation or disks have no direct and unique discrete equivalent. Disks are always approximations of their continuous counter parts and the concept of skeleton should be redefined. Moreover, on one hand a discrete skeletal line is not infinitely thin since it has a thickness of at least one pixel and on the other hand one pixel thin and centered skeleton are two mutually incompatible properties.

As Borgefors et al (1999) put it, the history of skeletonization of digital objects is almost as old as digital image analysis itself. The purpose of skeletonization for the 2D case is to reduce 2D discrete objects to 1D linear representations preserving topological and geometrical information. And for the 3D case, its purpose is to reduce 3D discrete objects into either 2D surfaces or 1D curves. Some authors call the surfaces and curves as medial surfaces and medial curves/axes respectively (Min Ma et al 2002, Lohman (1998), Tsao and Fu (1981)). A medial surface is a set of object voxels forming a surface of unit thickness, and a medial curve is a set of object voxels forming a curve of unit width.

The reason for obtaining the skeleton of an object is that the skeleton may by easier to analyze than the original image while maintaining the essential properties of the original object. In other words, we use skeletonization when we want to reduce the image to its essentials. The reduction to lower dimensions becomes even more desirable when dealing with 3D volume images.

The compression to 1D curves is possible only for solid objects having no cavities. A hollow torus, for example, could never be reasonably represented by a curve skeleton. The skeleton is a promising tool for an increasing number of applications such as medical image processing, biomedical imaging, and tracking moving objects(Ma, C.M and Wan, S.Y. 2001, Borgefors et al 1999).

The literature on 2D skeletonization is quite rich. But compared to the literature on 2D skeletonization, papers published in 3D skeletonization are still not very numerous. The reasons for this as Borgefors et al (1999) suggests, seems to be the difficulty to address and efficiently solve essential problems, such as topology preservation, in more than two dimensions.

The skeleton generated by a skeletonization process is expected to have the following desirable properties (as given by Bernard and Manzanera (1999)):

- Homotopy: the skeleton must preserve the topology of the original image. In other words, the Euler number should be the same before and after skeletonization.
- One-pixel-thickness: the skeleton should be made of curves or surfaces, i.e. one-pixel-thick objects. In other words, the skeleton should be as thin as possible.
- Mediality: the skeleton should lie in the middle of the shape, with every point of the skeleton having the same distance from the two closest borders of the object.
- Rotation invariance: skeletonization and object rotation should commute.
- Noise immunity: the skeleton should be fairly insensitive to noise on the image (boundary pixels added or removed).

- Reconstructibility: it should be possible to reconstruct the original image from the skeleton.

Generally, these desirable properties can be classified into two categories:

- *topological properties* - to retain the topology of the original object.
- *geometric properties* - to force the skeleton to be in the middle of the object, invariance under translation, rotation, and scaling, and preservation the end points of the original image.

However, in discrete space, most of these desirable characteristics are mutually exclusive, and, hence practical skeletonization methods are a compromise between them. For instance, thinness and reconstructibility. Forcing connectivity of the skeleton may introduce unnecessary points which are not essential for reconstruction, yielding a thick skeleton and hence conflicting with the thinness requirement.

The various applications of skeletons have several conflicting requirements which makes it difficult for a single method to address all of them. While some applications require the skeleton to be as thin as possible, others impose the condition that the object has to be reconstructible from the skeleton, for instance compression. If every minor feature in the original object needs to be captured for reconstruction, the resulting skeleton can be very dense.

4 Skeletonization Techniques

To get a skeleton, different techniques can be used. These techniques are classified into three major categories each of which is discussed below.

- distance transformation,
- Voronoi methods, and
- thinning techniques.

4.1 Distance Transformation

The distance transform at a point within an object is defined as the minimum distance to a boundary point. Since the skeleton is required to be centered with respect to the object boundary, the distance transform gives useful clues for point removal. Points closest to the center of the object would have the maximum distance transform value. Several distance metrics can be used to compute the distance transform, for instance, the Euclidean distance, manhattan and chessboard distance metrics (Lohman 1998).

Skeletonization based on distance transformation is performed using the following three steps:

- The original(binary)image is converted into feature and non-feature elements. The feature elements belong to the boundary of the object. The distance map is generated where each element gives the distance to the nearest feature element.
- The ridges (local extremes) are detected as skeletal points.

The distance transformation can be executed in linear (O(n)) time in arbitrary dimensions (where n is the number of the image elements, e.g. pixels or voxels)(Palagyi).

This method fulfills the geometric requirement but the topological correctness is not guaranteed (Palagyi).

Borgefors et al (1999) described a method for skeletonization of volume objects based on the distance transform which is performed in two steps - first performing reduction of the object to surface skeleton and then reducing the surface skeleton to a curve skeleton.

4.2 Voronoi Methods

The Voronoi diagram of a discrete set of points (called generating points) is the partition of the given space into cells so that each cell contains exactly one generating point and the locus of all points which are nearer to this generating point than to other generating points.

If the density of the boundary points (as generating points) goes to infinity then the corresponding Voronoi diagram converges to the skeleton.

Voronoi methods have the advantage of being very well defined and theoretically sound since they operate on a well known and powerful concept - the Voronoi diagram. Both the topological and geometrical requirements of a skeleton can be fulfilled by the skeletonization based on Voronoi diagrams, but it is an expensive process, especially for large and complex objects.

4.3 Thinning

A thinning algorithm is used to reduce unnecessary information by peeling objects layer by layer so that the result is sufficient to allow topological analysis. A thinning algorithm preserves the topology which implies, for example that an object can not vanish completely or be split into two or more objects, and no background components can be created or merged. It also should preserve the geometry of the object.

Thinning can be done in two ways: by using *morphological operators* and *deleting simple points.*

A thinning operator brings the boundary of the set inward. The skeleton is the result of reducing the boundary of the set to a thin line in the interior of the set.

Thinning can result in the skeleton, the boundary of an object, or the centroid of an object. It can also remove small branches of a skeleton. The results depend upon the shape of the structuring element, the number of times the structuring element is applied, and the different orientations of the structuring element.

The general strategy for 3D and 2D skeletonization does not differ significantly. Object voxels are changed to background voxels under the constraint that topology and geometry of the object are preserved. However, a number of new problems have to be solved in the 3D case. For example, when designing topology preserving removal operations, besides preventing disconnections and creation of cavities, which must also be done in 2D, one must also avoid the creation of tunnels and the excavation of unwanted deep cavities in complex surfaces (Borgefors et al 1999).

Morphological operators employ a method that deletes simple points. A pixel or voxel p is considered to be simple if it is a boundary point and its neighbor set with p and with p removed has exactly the same number of connected components. Deleting a simple point does not change the local topology or connectedness properties of the set. This method needs additional criteria for deleting simple points which prevent excessive shrinking. This involves identifying end-points of branches and disallow their deletion even if these points happen to be simple points (Bertrand and Maladain (1994)).

Thinning can be performed in two ways (Lohman 1998): either in parallel or sequential. In sequential thinning, only a single point may be deleted at a time and it always guarantees the preservation of the topology of the original image. But there is a likelihood that a sequential algorithm may eliminate all voxels/pixels except one.

The second class of thinning algorithms allow the simultaneous deletion of many points at a time. These algorithms are called parallel thinning algorithms as voxels/pixels may be deleted in parallel. Simultaneous deletion of points may destroy the topology of the image.

Parallel thinning algorithms are quite popular, as they often give better results provided some extra precaution is taken. One such precaution may be to allow only certain types of deletions to occur like deleting voxels/pixels that belong to a certain direction in parallel.

4.4 Example Algorithms

2D example

Two dimensional skeletonization can be done in several ways. Here a method that uses morphological operators as described by Soille (1998) is presented. It is based on the Hit-and-Miss Transform. A short overview is given below.

The skeleton is calculated by translating the origin of the structuring element to each possible pixel position in the image and at each such pixel position, comparing it with the underlying image pixels. If the foreground and background pixels in the structuring element exactly match foreground and background pixels in the image, then the image pixel underneath the origin of the structuring element is set to the background. Otherwise, it is left unchanged.

The thinning takes the form of a Hit-and-Miss transform and operates as follows:

1. Sweep the image with one of the given structuring elements.
2. If the 3x3 image pattern matches the structuring element then put a one on the corresponding location of the resulting image, otherwise put a zero.
3. Invert the resulting image and perform a binary AND of it with the initial image. This removes the points produced by the first structuring element.
4. Repeat process 1-3 until all structuring elements have passed over the image (each element takes as input the output of the previous one).
5. Repeat processes 1-4 until the image does not change any more.

The algorithm guarantees that connectivity will be preserved so the over all geometric structure of the object in the image is preserved.

3D Example

Various 3D skeletonization algorithms have been discussed in the literature. Most of them are based on distance transformation and thinning methods. Among the notable ones are the one proposed by Tsao and Fu (1981). But most of the algorithms used in 3D skeletonization suffer from one major shortcoming in that they do not preserve the topological properties of the original image.

Here we present the algorithm based on Tsao and Fu as described by Lohman (1998).

In Tsao and Fu's algorithm, we first compute the medial surface, and then obtain the medial axis skeleton in a second pass.

The algorithm works by parallel deletion of directed border points that are simple and not final. A simple point is called a final point if and only if the number of of black points (foreground) in its 6-(26-)neighborhood is less than 2. In addition to the simple point condition and the final point condition, this algorithm also uses the "checking plane" condition which ensures that the resulting skeleton is smooth.

For each subcycle the two checking planes are the 3x3 planes orthogonal to the current subcycle direction with the current voxel in the center of the plane. The checking plane condition holds if the current voxel is "simple" in a 2D case, i.e. if its deletion does not disconnect the remaining black voxels in the plane and the number of remaining black voxels is not less than 2.

Once the medial surface skeleton is extracted, we can proceed to obtain the medial axis in a similar fashion. The only difference is that the checking plane condition is relaxed in this case. While in the medial surface case number of remaining black voxels in the checking plane was required to be greater than 2, it now needs to be just greater than 1. Thus to obtain the medial axis, we simply use the medial surface as input to the second stage of the skeletonization with the checking plane condition relaxed.

But in the Tsao and Fu's algorithm, the Euler number is not preserved. Therefore, in this paper an attempt is made to design a 3D skeletonization algorithm which can preserve the topological properties of an image.

5 A New Sequential Thinning Algorithm

5.1 2D Thinning

In this paper, with the aim of finding a topology preserving 3D thinning algorithm, we first employ an algorithm based on removing the central pixel in the 3x3 neighborhood of the candidate pixel which preserves the topology and geometry and, then try to adapt the algorithm to the 3D case. The 3x3 neighborhood of the point to be removed is coded as in Table 4 with the candidate pixel to be removed at the center.

Table 4: Encoding of 3x3 neighborhood configuration of the central pixel

1	2	4
8	16	32
64	128	256

The algorithm is based on computing the local Euler number before and after removing the candidate pixel and then checking whether there is a difference in the computed values.

Furthermore, in order to preserve the geometric criteria (preserve end points of the object), we consider change of the boundary length when removing the central pixel. This involves finding the optimal values for the difference of the boundary length so that the resulting skeleton maintains its end points in addition to having the same Euler number as the original image. This depends on the type of neighborhood configuration chosen.

For all the possible neighborhood configurations ((4,8), (6.1,6.1),(6.2,6.2),(8,4)), the Euler number before and after skeletonization is calculated and a check is made whether there is a difference in the Euler number after and before skeletonization and whether the central pixel is a foreground pixel. For the geometric criteria, we consider the difference in the boundary length before and after removing the candidate pixel. Optimal acceptable values for the boundary length difference as well as the type of skeleton generated based on these values have been found (See chapter 6 for the details).

An outline of the algorithm that we used for 2D skeletonization is given below.

1. A look up table is generated which corresponds to the 8 neighbor pixels with different cases for each of the 4 neighborhood configurations. This lookup table stores a boolean value depending on the Euler number difference before and after removing the candidate pixel and the extent of the boundary length difference.

2. We scan the image sequentially and for each pixel we consider its 3x3 neighborhood. We calculate the values for the current configuration of the candidate pixel and compare these values with the values stored in the lookup table. If the two values match, then the candidate pixel is turned into background as it is not part of the skeleton. Otherwise, it is retained and will be part of the skeleton. The fact that the current pixel is removed will be taken into account when processing subsequent candidate pixels.

In the algorithm, consideration is made to treat boundary points.

5.2 3D Thinning

Here, we try to employ a similar thinning algorithm as in the 2D case. We remove the central voxel in the 3x3x3 neighborhood which preserves the Euler number and the geometry of the original object.

In this case, to make sure that the skeleton preserves the geometry of the original object,we make use of the change in surface area and the integral of mean curvature before and after removing the candidate voxel. Based on the values of the change in the surface area and integral of mean curvature and the type of neighborhood configuration chosen, different kinds of skeletons are obtained. The 3x3x3 neighborhood of the point to be removed is coded as in Figure 4
To calculate the surface area and integral of mean curvature, we use the method by Ohser and Mcklich (2000)which makes use of the Crofton Formula from Integral Geoemetry where the 2x2x2 neighborhood configuration is used.

Another method is also given by Blasquez (2003) to calculate the surface area and the curvature based on the number of open cubes, open edges, open vertices and open faces.

The algorithm we used in this 3D thinning is as follows:

1. A look up table is generated which corresponds to the 26 neighbor

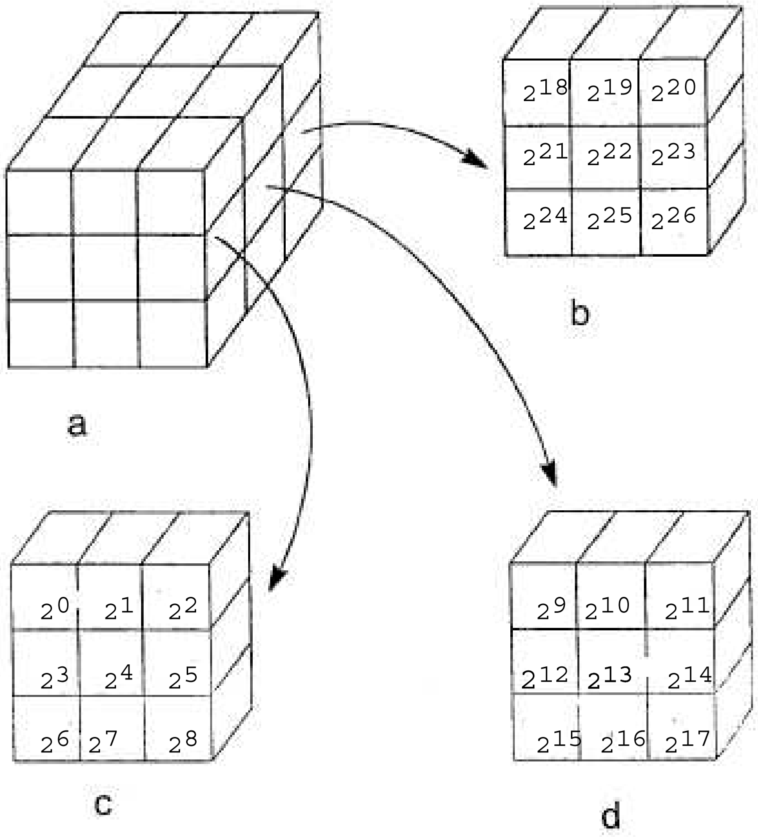

Figure 4: The coding of the 3x3x3 neighborhood of voxel. (Clockwise from upper left: 3x3x3 neighborhood(a); back plane(b); middle plane(d); front plane(c).)

pixels. All of the 26 negighbours are examined in order to compute the address of the entry in the lookup table. This lookup table stores a boolean value depending on the Euler number difference before and after removing the candidate pixel and the extent of the difference in the surface area and integral of mean curvature.

2. We scan the image sequentially and for each pixel we consider its 3x3x3 neighborhood. We calculate the values for the current configuration of the candidate pixel and compare these values with the values stored in the lookup table. If the two values match, then the candidate pixel is turned into background as it is not part of the skeleton. Otherwise, it is retained and it will be part of the skeleton. The fact that the current pixel is removed will be taken into account when processing subsequent candidate pixels.

6 Experimental Results

6.1 2D Thinning/Skeletonization Results for test bodies

The test bodies considered are rectangles. The optimal values for the difference in the boundary length are found to be different under different neighborhood configurations. The values for a pair of neighborhood configurations coincide. In other words, the values for (4,8) and (6.1,6.1) neighborhood configurations are identical and the values for (8,4) and (6.2,6.2) are also the same. These values along with the type of skeleton they can produce are given in Table 5 and Table 6.

Table 5: Optimal values for boundary length difference for (4,8) and (6.1,6.1) adjacency pairs

Boundary length difference	Resulting skeleton
$<= -0.95$	a point
between -0.68 and -0.94	a vertical or horizontal line
between -0.28 and -0.67	in between horizontal/vertical line and a skeleton with end points preserved
between -0.27 and 0.39	a skeleton with its end points preserved
$>= 0.40$	no change in the original image

Table 6: Optimal values for boundary length difference for (8,4) and (6.2,6.2) adjacency pairs

Boundary length difference	Resulting skeleton
$<= -0.95$	a point
between -0.28 and -0.94	a vertical or horizontal line
between -0.27 and 0.39	a skeleton with its end points preserved
$>= 0.40$	no change in the original image

The following results were obtained during the experiment:

- Euler number is preserved for the (4,8) neighborhood configuration for all types of skeletons for the test bodies used.

- Euler number for (6.2,6.2) neighborhood configuration was preserved for all types of skeleton while (6.1,6.1) neighborhood configuration was not tested for preservation of Euler number.

- Euler number is preserved for the (8,4) neighborhood for all types of skeletons for the test bodies used.

For complex images the algorithm also showed a slight difference in the Euler number of the original image and its skeleton which is due to the boundary effects. But when the image is padded with zeros along its boundaries, the Euler number for the (4,8) and (6.2,6.2) is preserved. But Euler number is not preserved for the (8,4) neighborhood.

(4,8) Adjacency Skeletonization Result

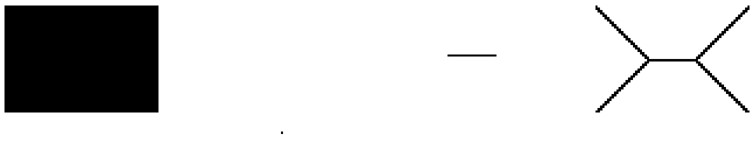

Figure 5: Skeleton of a rectangle for (4,8) adjacency. From left to right: Original, Skeleton for boundary length difference >=-0.95, between -0.68 and -0.94, and between -0.27 and 0.39 respectively.

(6.1,6.1) and (6.2,6.2) Adjacency Skeletonization Result

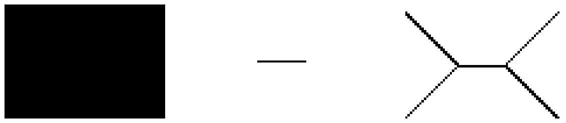

Figure 6: Skeleton of a rectangle for (6.1,6.1)adjacency. From left to right: Original, Skeleton for boundary length difference between -0.68 and -0.94, and between -0.27 and 0.39.

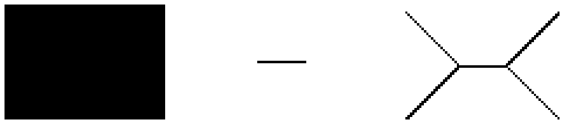

Figure 7: Skeleton of a rectangle for (6.2,6.2) adjacency. From left to right: Original, Skeleton for boundary length difference between -0.28 and -0.94, and between -0.27 and 0.39.

(8,4) Adjacency Skeletonization Result

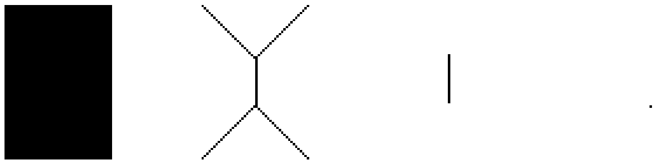

Figure 8: Skeleton of a rectangle for (8,4) adjacency. From left to right: Original, Skeleton for boundary length difference between -0.27 and 0.39, between -0.28 and -0.94, and <=-0.95.

6.2 Application to sample 2D image data

Since the original image shown in Figure 9 is not suitable for skeletonization, we performed a preprocessing step. That is, we first applied closure by using the structuring element {15,15,15,15,15,15,15,15,15} and then opening by the structuring element {3,3,3,3,3,3,3,3,3} to remove the noise. Finally it is padded with zeros along its boundaries to take account of boundary effects. Therefore, after these preprocessing steps,the filtered image shown in Figure 10 is obtained. By applying our skeletonization algorithm to this filtered image, we obtained a skeleton which preserves both topology and geometry for (4,8) and (6.2,6.2) adjacency pairs while topology is not preserved for the (8,4) adjacency pair. But our opinion is that there should be an error in the algorithm for the Euler number calculation for the (8,4) adjacency as the number of connected components for the resulting skeleton aree easily counted to be nine which is equal to the number of connected components of the original image. However, the number of connected components for the (8,4) adjacency pair using the program gives an Euler number of 2836 for the skeleton which is not the correct result.

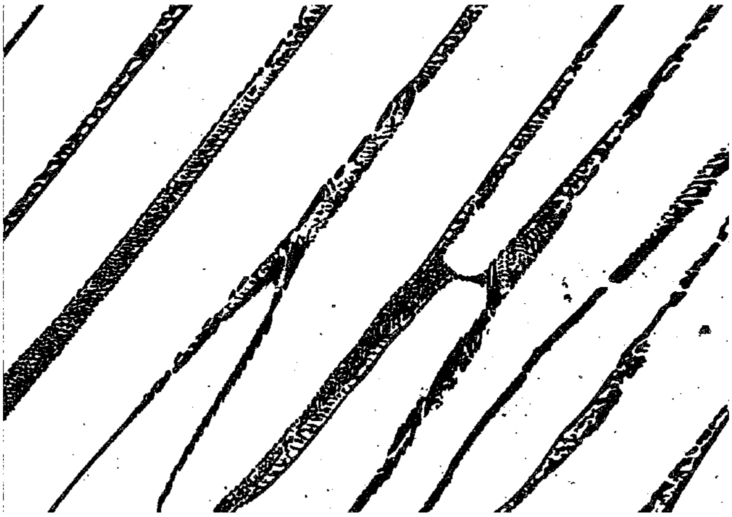

Figure 9: original image

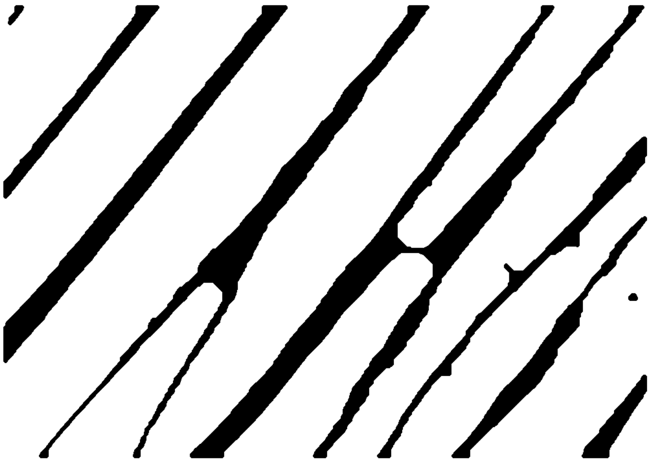

Figure 10: The original image closed by the structuring element:{15,15,15,15,15,15,15,15,15} and then opened with the structuring element {3,3,3,3,3,3,3,3,3} to remove the noise and then padded with zeros along its borders

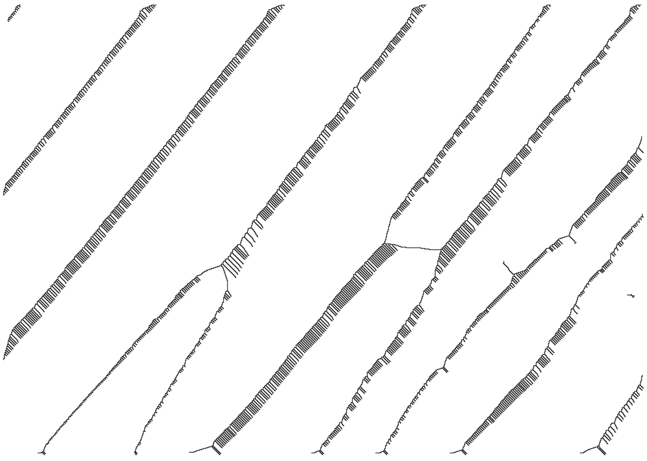

Figure 11: Skeleton for (4,8) adjacency and boundary length difference >=-0.20. Euler number: Original:9,Skeleton:9

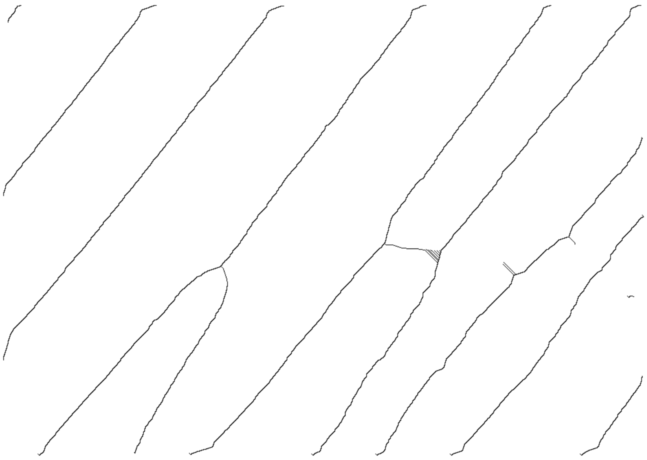

Figure 12: Skeleton for (6.2,6.2) adjacency and boundary length difference >=-0.20.Euler number: Original:9,Skeleton:9

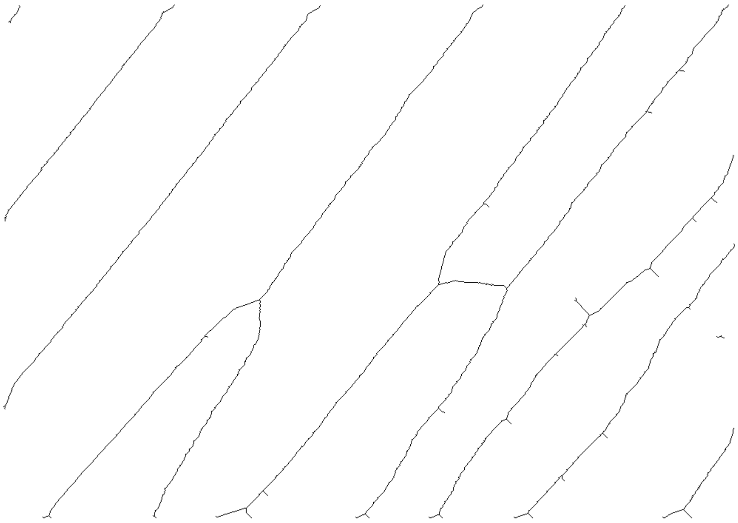

Figure 13: Skeleton for (8,4) adjacency and boundary length difference >=-0.20.Euler number: Original:9,Skeleton:2836

6.3 3D Thinning/Skeletonization Results for test bodies

Here we considered a cube, a pyramid, a ball and a cylinder as our test bodies.

Due to lack of time, tests are made only for the (6,26) adjacency systems. Under different values for the difference in the surface area and integral of mean curvature, different types of skeletons are obtained.

The different values for the difference in the surface area and the difference in the integral of mean curvature before and after removal of the candidate pixel which gives different skeletons for the (6,26) adjacency pairs are given in Table 7.

Table 7: Values for surface area difference and inegral of mean curvature for (6,26) adjacency pairs for different test bodies

Surface Area	mean curvature	test body	Resulting skeleton
$>= 4.0$	any value	cube	surface skeleton
$>= -1.5$	$>= -4.0$	ball	surface skeleton
$>= 2.5$	$>= -4.0$	pyramid	surface skeleton
$>= 3.5$	$>= 0.0$	cylinder	surface skeleton

The values given in the table for the difference in the surface area and integral of mean curvature are those which preserves the topology as well as the geometry of the original object.

Skeleton of a cube

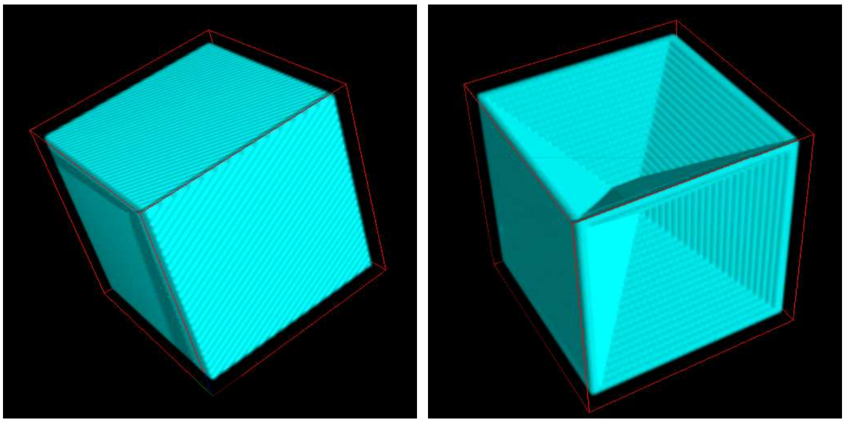

Figure 14: Skeleton of a cube for (6,26) adjacency. From left to rigt: Original, Skeleton

Skeleton of a pyramid

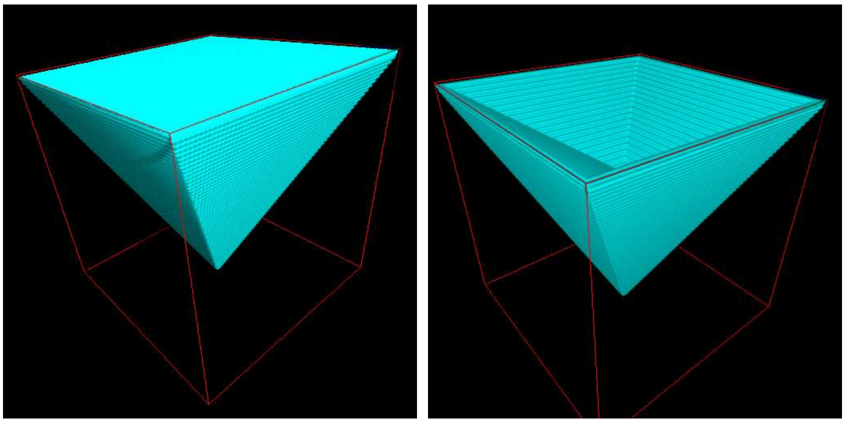

Figure 15: Skeleton of a pyramid for (6,26) adjacency. From left to right: Original, Skeleton

Skeleton of a ball

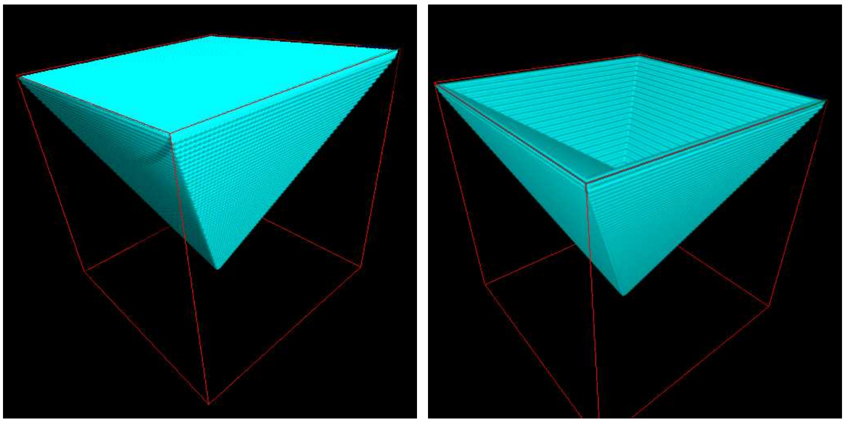

Figure 16: Skeleton of a ball for (6,26) adjacency. From left to right: Original, Skeleton

Skeleton of a cylinder

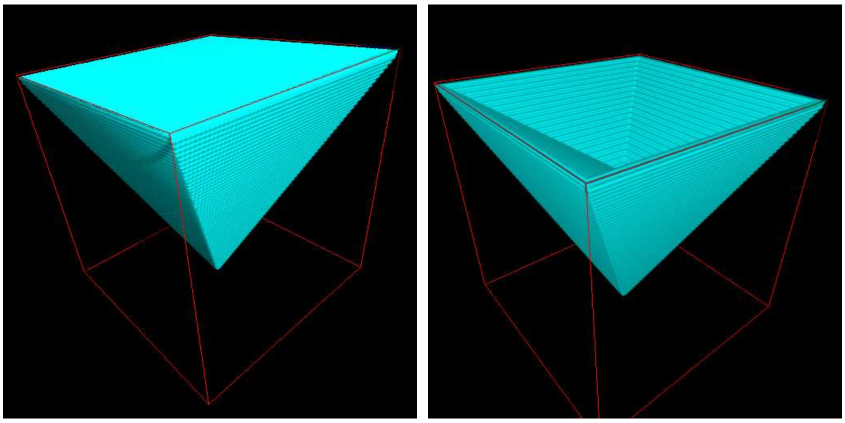

Figure 17: Skeleton of a cylinder for (6,26) adjacency. From left to right: Original, Skeleton

6.4 Application to sample 3D image data

Here we present application of our algorithm to a 3D image of an open foam. When we apply our skeltonization algorithm to the open foam, there is a difference in the euler number of the original and its skeleton but the geometry is preserved. Our opinion is that the difference in the euler number can be attributed to the boundary effects. The orginal foam as well as its skelton are given in Figure —.

7 Discussion

We presented a new sequential thinning algorithm for both 2D and 3D binary images which preserves the topology and geometry of the image.

In the case of 2D binary images, we considered the 3x3 neighborhood of the candidate pixel. We used the difference in the local Euler number before and after removing the candidate pixel and as well as the difference in the boundary length as a criteria for determining whether a given pixel belongs to the skeleton or not. While the Euler number is used for determining the topological criteria, the difference in the boundary length is used to determine the geometric criteria. Using these criteria, we got the expected results for our test bodies (rectangles and squares) and also for a sample 2D image. In other words, we arrived at a topology preserving sequential thinning algorithm which preserves the topology and geometry of the original image and gives different skeletons depending on the values in the boundary length difference. All the possible adjacency pairs (4,8), (8,4), (6.1,6.1), (6.2,6.2) were considered and expected results obtained.

In 3D binary images, we considered the 3x3x3 neighborhood of the candidate pixel. We made use of the difference in the surface area and the difference in the integral of mean curvature for the geometric criteria in addition to the difference in the local Euler number for the topological criteria for determining the deletability of a candidate pixel. Just like in the 2D case, depending on the extent of the surface area difference, the integral of mean curvature difference and the difference in the local Euler number, different skeletons are obtained for our test bodies (cubes and pyramids). But due to lack of time, only the (6,26) adjacency is considered.

We know of no previous work (to the best of our knowledge) which used these criteria for determining the deletability of a given pixel.

Further studies can be done in testing the rest of the adjacency systems for the three dimensional case.

8 References

1. Ma, C.M., Wan, S.Y. and Chang, H.K. 2002. Extracting Medial curves on 3D images. *Pattern Recognition Letters* 23, 895-904.

2. Ma, C.M. and Wan, S.Y. 2001. A Medial-Surface oriented 3-D two subfield Thinning Algorithm. *Pattern Recognition Letters* 22, 1493-1446.

3. Michielsen, K. and De Raedt, H. 2000. Morphological Image Analysis. *Computer Physics Communication* 132, 94-103.

4. Borgefors, G. , Nystrm, I., and Di Bja, G.S. 1999. Computing Skeletons in three dimensions. *Pattern Recognition* 32, 1225-1236.

5. Blasquez, I. and Poiraudean, J.F. 2003. Efficient Processing of Minkowski Functionals on a 3D Binary Image Using Binary Decision Diagrams. *Journal of WSCG*, Vol. 11, No.1.

6. Ma, C.M. and Wan, S.Y. 2000. Parallel Thinning Algorithms on 3D (18,6) Binary Images. *Computer Vision and Image Understanding* 80, 364-378.

7. Saha, P.K. and Chaudhuri, B.B. 1996. 3D Digital Topology under Binary Transformation with Applications. *Computer Vision and Image Processing* Vol.63, No.3, 418-429.

8. Toriwaki, J. and Yonekura, T. 2002. Euler Number and Connectivity Indexes of a 3D digital Picture. *Review Forma*, 17, 183-209.

9. Toriwaki, J. and Yonekura, T. 2002. Local Patterns and Connectivity Indexes of a 3D digital Picture. *Review Forma*, 17, 275-291.

10. Bertrand, G. and Maladain, G. 1994. A new characterization of three-dimensional simple points. *Pattern Recognition Letters* 15, 169-175.

11. Tsao, Y.F. and Fu, K.S. 1981. A Parallel Thinning Algorithm for 3D Pictures. CGIP. 17, 315-331.

12. Lohman, G. 1998. Volumetric Image Analysis. Chichester, New York: Wiley Teubner.

13. Bernard, T.M. and Manzanera, A. 1999. Improved Low Complexity Fully Parallel Thinning Algorithm.

14. Gagvani, N. 2001. Parameter Controlled Skeletonization - A Framework for Volume Graphics, Ph.D. Thesis, the State University of New Jersey.

15. Ohser, J. and Mcklich, F. 2000. Statistical Analysis of Microstructures in Materials Science. Chichester, New York: John Wiley and Sons.

16. Ohser, J., Nagel, W. and Schladitz, K. 2002b. The Euler number of discretized sets - on the choice of adjacency in homogenous lattices. In: Mecke KR, Stoyan D, eds. *Morphology of Condensed Matter*. Heidelberg: Springer.

17. Ohser, J., Nagel, W. and Schladitz, K. 2003. The Euler number of discretized sets - surprising results in three dimensions. *Image Analysis and Stereology* 22, 11-19.

18. Serra, J. 1982. Image Analysis and Mathematical Morphology, Vol.1. London: Academic press.

19. Soille, P. 1999. Morphological Image Analysis : Principles and Application.

20. Eckhardt, U. and Latecki, L. Digital Topology.

21. http://www.inf.u-szeged.hu/ palagyi/skel/skel.html

9 Appendices