Abstract of thesis entitled

Topological Analysis and Visualization of Micro Structure of Trabecular Bone

submitted by

Wang Xiaoting

for the degree of Master of Philosophy
at The University of Hong Kong
in February 2004

Today computer graphics techniques are used in many fields in our daily lives. Medical visualization is one of the most important applications of these techniques. With the development of medical imaging techniques, such as X-ray, CT, and MRI, diagnostic accuracy has been improved greatly and many lives have been saved. Applying visualization techniques to medical data allows a better understanding of the internal structure of human bodies. However, topology study and analysis of the structure are also indispensable parts of this process, since they provide a sound mathematical foundation for image processing.

The study of trabecular bone is important in osteoporosis research. In this thesis, 3D trabecular bone volume data obtained from Micro CT are used for visualization and topological analysis. Algorithms in image processing, such as thinning, classification and segmentation, are used in trabecular bone topological analysis. Computer graphics techniques are used to make the study intuitive and direct. We focus on extracting rod-like structures in trabecular bone, which have been the subject of most studies on its topology and visualization.

Because of the complexity of the trabecular bone topological structure, the use of existing algorithms to extract the rod-like structures of trabecular bone is unable to yield satisfactory results. We present a new method which can distinguish more accurately whether a structure is rod-like or not. Visualization techniques are used to illustrate the results of the study.

Contents

Chapter 1

Introduction

Scientific technology has greatly changed the life of human beings. In recent years, computer technology is among the fastest developing techniques and it is playing a more and more important part in everyday life. Computer Graphics and Vision, as an important field of computer science, enables us to "copy" the real world into computers, so that remote or over-sized objects become accessible. Using computer graphics, we can design and simulate something before really constructing it or create significant effects on screen which cannot be realized in the real world. We also can display tiny things and internal structures that cannot be observed by the naked eyes. For example, with the aid of medical imaging techniques, like X-ray, Computed X-ray Tomography (CT), Magnetic Resonance (MR) and ultrasound (US), doctors improve greatly their surgical performance and more lives are saved.

Many doctors rely on their experience and anatomical knowledge to diagnose disease from patient's symptom or find the cut position on the surface of the body during operation. But this kind of diagnosis often has errors that may cause unnecessary suffering or poor results sometimes. More than one hundred years ago, X-ray was discovered and used to produce 2D medical imaging. With the development of medical imaging technology, doctors are no longer content with 2D images. Since the human body exists in 3D, 2D images can only give partial information and some important spacial information is missed. Now doctors are more and more inclined to get integrated and accurate 3D medical images. CT, MRI and US have appeared in succession and play an important role in modern medicine. With the great help of 3D medical images, doctors can examine the internal body structure carefully to identify the abnormal organ or determine the most appropriate position of operation. And after operation, they can also use medical imaging to keep up with the recovery process of the patient.

Although 3D internal structures can be obtained from CT and MR, the resolution of these images is still too low for some small objects, such as tooth and trabecular bone. High resolution and more precise Micro CTs came into being. Studying trabecular structure is a meaningful aid to osteoporosis. With the process of bone mass loss and bone tissue deterioration, osteoporosis occurs. This process often has no obvious symptoms and may appear at any age. Patients with osteoporosis have higher risk of bone fracture. One fourth of women and one eighth of men over 50 years old suffer from osteoporosis [1]. Osteoporosis affects bone health seriously and has attracted great interests of doctors. Although osteoporosis is mainly determined by bone density, the topological structure of trabecular bone also contributes significantly to bone strength [2]. Study of the bone structure can be meaningful in judging osteoporosis and can greatly help doctors to better understand osteoporosis. One of the useful tools to study trabecular bone is Micro CT. After Micro CT scanning, volume data of trabecular bone is obtained and 3D trabecular structures are reconstructed and displayed for subsequent manipulation and analysis.

In some image processing operations, such as image thinning, component counting, connectivity judging etc., sometimes we only need to care about the relative positions of two points but not their absolute positions. Digital topology provides us with suitable mathematical foundations to solve this kind of problem. Using topological analysis, some problem in visualization can be understood and solved in a mathematical way.

Computer visualization is a technique that can be used in many fields to model complex or tiny objects that cannot be seen by human beings directly or to simulate phenomena and scenes in nature. Through computer visualization, objects are exhibited in screen vividly and then people can observe them from every angle, modify some parts or even to simulate their changes in nature.

A joint project is conducted in cooperation between the Department of Computer Science and Information Systems and the Department of Orthopedic Surgery of the University of Hong Kong. The aim of the project is to use advanced computer visualization techniques to display tiny trabecular bone structure and to apply topological analysis methods to understand the trabecular structure better. Rod-like structures in trabecular bone are the parts on which our interests are focused. Using existing algorithms, rod-like structures cannot be identified satisfactorily. A new method is introduced to improve the accuracy of rod-like structure identification. The results of this project can be used in teaching demonstration and the data obtained from topological analysis can be used to aid the understanding of osteoporosis better.

The remainder of this thesis is organized as follows. In Chapter 2 we shall give a brief overview of the background of this research. In Chapter 3, some related work and algorithms used in this project are introduced in detail. The modules and manipulation methods are presented in Chapter 4. At last, in Chapter 5, unsolved problems and future work are discussed.

Chapter 2

Background

2.1 Surface and Volume

The real world is 3D. Using modern scanning techniques, we can get the volumetric dataset of 3D objects. There are different situations for different applications, so different representation methods are used in computer graphics. For example in a movie or a game, what we need to do is to make a person or object look like "real", so we do not need to draw the internal structures of the virtual person or the virtual object for such information is invisible. If we still use the whole volumetric dataset to represent an object, it will seriously slow down the speed. Imagine that in the film Matrix II, when 100 Agent Smith are fighting with Nero in the same scene, if all 100 Agent Smith are represented with volumetric datasets, the storage of the computer must be very big and the rendering time will be extremely slow when actions are performed. In this situation, we only need to set up the 3D surface models and outline their surfaces. This kind of rendering is called surface-based rendering and the elements of surface-based rendering are points, lines and polygons (Figure 2.1).

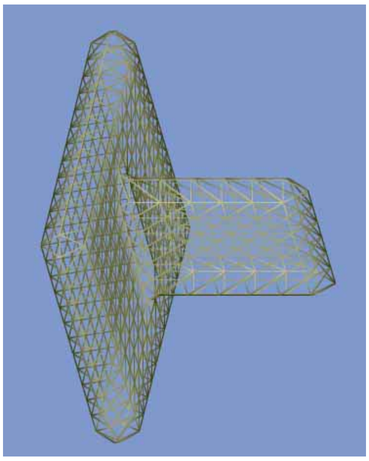


Figure 2.1 Surface Rendering

In contrast to surface-based rendering, there is volume-based rendering whose element is voxel. As Foley mentioned in [3], *volume rendering* is used to show the characteristics of the interior of a solid region. Volume rendering is a technique for directly displaying a sampled 3D field without first fitting geometric primitives to the samples. In some applications, especially in medical applications, having only the surfaces of an organ, bone or whole human body cannot give us enough information. In volumetric dataset collected from CT scan or MR scan, each point of the 3D grid has a value that represents the tissue's density. Using volume-based rendering techniques, we not only can represent the surfaces, but also the interior, for example use a cut plane to examine the interior part or only take out a small part to observe carefully. This is not what surface-based modeling and surface-based rendering can do.

There are many algorithms for speeding up surface-based rendering and volume-based rendering respectively. Scan converting line algorithm [3], hidden surface algorithm [3] are included in surface-based rendering. Marching cube algorithm [4] and ray casting method [5] are included in volume-based rendering. Besides software speed up, hardware speed up for rendering techniques is growing rapidly for the wide market of 3D computer games, virtual reality and 3D art in film. They can handle points, lines and polygons efficiently. As to volume rendering, hardware accelerator is not as available as surface-based rendering. VolumePro [6] is a real-time volume rendering system which includes a PCI card and a software interface. VolumePro is based on the Cube-4 volume rendering architecture and implements Ray Casting algorithm.

2.2 Volume Visualization

Volume visualization, which includes the representation, manipulation, and rendering of volumetric datasets, is a method of extracting meaningful information from volumetric datasets by using interactive graphics and imaging [7]. Kaufman mentioned the following advantages of volume visualization [8]:

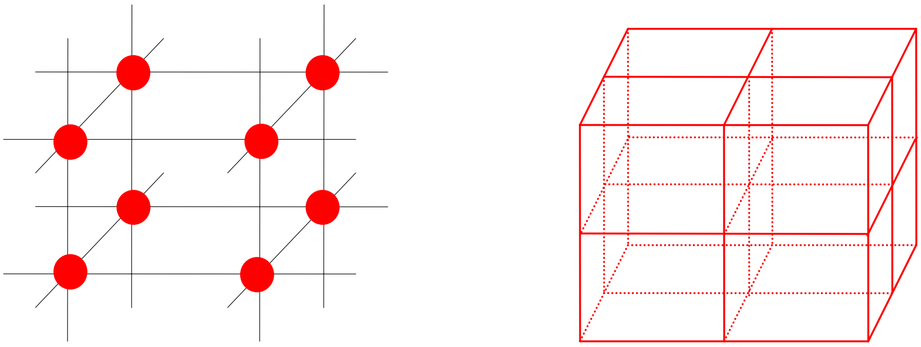
- Insensitivity to scene complexity.
- Insensitivity to object complexity.

- Viewpoint independence.
- Support sampled and simulated datasets.
- Can display inner information of an object.
- Boolean or block operations can be applied.

Volumetric dataset is represented as a 3D grid of voxel which is the element of volume.

2.2.1 Voxel

A voxel is the 3D counterpart of a pixel in 2D space. A voxel can be represented by a quadruple (*x,y,z,v*). Cartesian coordinates (*x,y,z*) are its position in 3D space and "*v*" is some kind of characteristic associated to this voxel. The value "*v*" can be either a scalar or a vector. To represent density, temperature and penetrability etc., a scalar value is enough, while representing direction and strength of fluid or wind, a vector is needed. In this thesis, the value used in volumetric dataset is a scalar, e.g. bone density, only. There are two different representations of a voxel (Figure 2.2).



(a) (b)

Figure 2.2 Voxel Representation. (a) As grid point; (b) As solid cube.

The first is that a voxel is a grid point in a 3D lattice without occupying any space. The other is that a voxel is a solid cube and all the voxels are packed to form the whole volumetric dataset. Those two representations are in fact the same. If we use the center of each cube as a voxel, the second representation can be converted to the first one.

If the scalar value is only either 0 or 1, the volume is called a binary volume. And if the scalar value can be any number in a range, this volume is called a gray-scaled volume. In this thesis, some manipulations treat the volumetric dataset as a gray-scaled volume such as in visualization part and some may take it as a binary volume such as in thinning and classification parts. If a voxel's value is 1 (for binary volume) or higher than a user-defined threshold (for gray-scaled volume), this voxel is also called a black, object or bone voxel in this thesis. A voxel with value 0 (for binary volume) or value smaller than a threshold (for gray-scaled volume) is also called a white, background or marrow voxel. In this thesis, the terms *volume dataset* and *3D digital image* are also used interchangeably.

2.2.2 Volume Rendering

Volume rendering can be classified into indirect volume rendering (IVR) and direct volume rendering (DVR) [9]. IVR first constructs an iso-surface from the volumetric dataset, and then uses traditional surface-based rendering techniques to display the profile of objects. In IVR, the most famous method is the Marching Cube (MC) algorithm [4]. Contrary to IVR, DVR does not need to convert volume element to surface element, but uses voxels directly. In DVR, the well known algorithms are Ray Casting (RC) approach [5], the forward projection approach [10], the splatting algorithm [11] and the 3D texture mapping based projection technique [12]. Now we will focus on MC and RC which are used in this project.

2.2.2.1 Marching Cube Algorithm

MC uses linear interpolation to setup triangle models of constant value surface (iso-surface) and uses normalized gradient of original data to calculate shading [4].

First a logic cube is defined by 8 voxels: four from one slice and four from a neighboring slice and a user defined constant value C is chosen. To determine how the iso-surface passes through the volume, divide-and-conquer method is used to create triangle patches in a logic cube. If one voxel's value is higher than C, this voxel is out of the surface and if its value is smaller than C, this voxel is inside the surface. Each voxel can be an "out" or "in" state, so totally there are $2^8 = 256$ situations in a logic cube. Considering the complementary case of one situation (i.e., each vertex's state changes to its opposite), the topology of triangles in this logic cube is not changed. And considering the rotational symmetry, some cases are in fact the same one. So the 256 situations can be reduced to 15 cases [4](Figure 2.3).

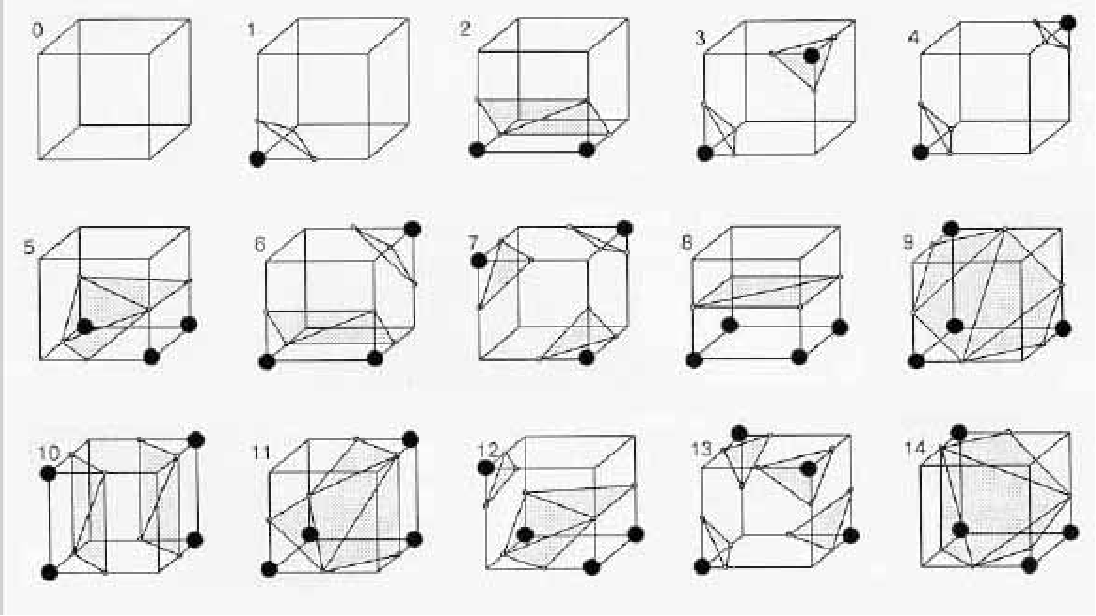


Figure 2.3 Triangulated Cubes
Courtesy of W.E. Lorensen and H.E. Cline

A look-up table is set up beforehand, which includes the 256 cases. In every entry, we can find edges which are intersected by the isosurface and these edges are output by a sequence that forms a front face. According to the inside or outside status of each vertex of a cube, an entry index is created, Index = $V_8\ V_7\ V_6\ V_5\ V_4\ V_3\ V_2\ V_1$ (V_i is 0 or 1, instead of inside or outside of isosurface). We use this index to look up the pre-calculated table and find out which edges are intersected with the isosurface. The value of one of the vertices on such an edge must be smaller than the threshold and the other must be bigger than it. Using linear interpolation, the intersecting location can be found on the edge. Then, we store all the location information in a list and store each location's normal value into another list. The normal value is the linear interpolation of normal values of the two end vertices. And each vertex's normal value is computed using central differences.

$$\begin{cases} g_x = \dfrac{V_{x+1,y,z} - V_{x-1,y,z}}{2} \\ g_y = \dfrac{V_{x,y+1,z} - V_{x,y-1,z}}{2} \\ g_z = \dfrac{V_{x,y,z+1} - V_{x,y,z-1}}{2} \end{cases}$$

Having locations and normal values of vertices of a triangle, this triangle can be shaded.

The MC method does not guarantee a continuous surface. “Holes” can be generated when two ambiguous faces meet [13]. Nielson [14] suggested the asymptotic decider method to settle this problem.

MC is such a commonly used method for generating iso-surface and lots of researches are carried on MC to improve its performance. In MC, the number of triangles generated is proportional to rendering time. Some researchers designed algorithms to reduce the number of triangles. Montani [15] used midpoint of each edge as the intersection point of the iso-surface and the logic cube instead of linear interpolation, i.e., only to keep the topology. This method can reduce the number of generated triangles by merging output facets into large coplanar polygons. Although this approach is simpler than MC, it loses some precision. Shekhar [16] presented a decimation method to downsample the volume data where the iso-surface is mostly flat and maintain high resolution at places with finer details. This approach balanced surface details and rendering time.

The disadvantages of MC are:

- It cannot represent partially transparent materials correctly.
- A binary decision has to be made on the position of the intermediate surface.
- False positives (introducing features that do not exist) and false negatives (losing tiny features) are introduced.
- Inappropriate for low-resolution sampling.

2.2.2.2 Ray Casting Method

Compared with MC algorithm, Ray Casting method does not need to convert 3D information to a surface-based intermediate structure. In RC, a volume can be viewed as making up by spherical particles. When rays cast through the volume, those particles both absorb energy and scatter light [17]. From each pixel on the image plane, a ray is cast into the volume (Figure 2.4).

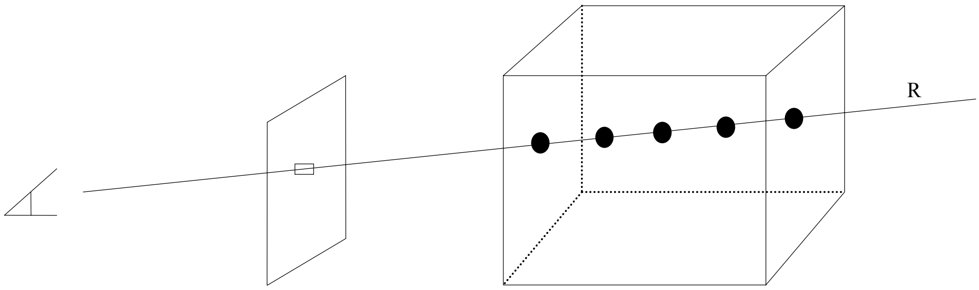


Figure 2.4 Ray Casting

By accumulating color and opacity of each voxel on this ray in back-to-front order with the background color, the result obtained is the final color of this pixel on the image plane. Levoy [5] described his algorithm in two steps: one is visualization step and the other is classification step. In visualization step, surface shading is calculated on each voxel from a reflection model using local gradient. And in classification step, each voxel's opacity is determined. The advantage of separating these two steps is that undistorted visualization result can be got. For each voxel x_i, C_{in} is the incoming intensity and C_{out} is the outgoing intensity. We have this formula $C_{out} = C_{in}(1 - \partial(x_i)) + c(x_i)\partial(x_i)$, where $\partial(x_i)$ is each voxel's opacity and $c(x_i)$ is its shade. For a whole ray, the first C_{in} is the background color and the last C_{out} is the color of a pixel on image plane.

Calculations in RC are heavy. To increase rendering speed, optimization algorithms such as Heirarchical Spatial Enumeration, Adaptive Termination methods are used [18]. Because there is no need to do any binary decision on the position of the intermediate surface, small features will not be missed. As to the smooth silhouettes and aliasing problem, Levoy used selective blurring and super-sampling to improve them. RC can display structures within other structures but not just one iso-surface. RC is now implemented in some visualization systems and visualization packages such as VolumePro, Vizard, Data Visualizer and IBM DataExplorer, etc.

2.2.2.3 Compare MC and RC

Rendering time and memory cost in MC is proportional to the number of triangles generated while memory cost in RC is related to the size of volume only. In MC, there is always a tremendous number of triangles, so the memory cost in MC is generally larger than RC. More operations are used in MC than in RC.

In general, RC is efficient in large size volume rendering while MC is appropriate to iso-surfaces with small number of triangles. RC and MC have their different application fields. They are both important volume rendering techniques.

2.3 Medical Data

2.3.1 X-ray

Wilhelm Konrad Röntgen won the first Nobel Prize in Physics in 1901 for his discovery of X-ray in 1895. Back to that year, he found an unknown radial which can penetrate his hand and left the image of hand skeleton on paper plane (Figure 2.5). X-ray's penetration power varies among different materials, because of the different density of materials. For example, bone can absorb more X-ray than soft tissues. The different characteristics of X-ray absorption give out the gray-scale image of materials combined together. From this kind of gray-scale images, different parts can be observed clearly. X-ray technique is wildly used in medicine, industry and commerce to examine material structure. Until today more than 100 years later after X-ray is found, using X-ray image to check chest's health is still a common method in medicine.

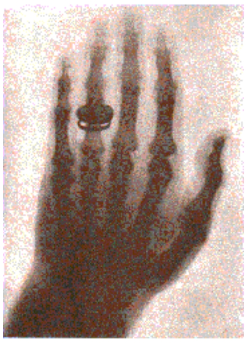


Figure 2.5 Hand skeleton of Röntgen's wife http://www.bud.org.tw/museum/s_star17.htm

The image obtained from X-ray is the accumulated effect of all the parts that the ray penetrates and it is only 2D information. If a tumor is blocked by bone, in X-ray image no abnormity can be found. In order to get information of every point in 3D, Computed Tomography is designed.

2.3.2 Computed Tomography (CT)

In 1972 the world's first CT scanner was devised by Sir Godfrey Hounsfield. Computed Tomography is also named as computerized axial tomography (CAT) scanning. CT is a specialized X-ray imaging technique. During the process of CT scanning, an X-ray source tube rotates around the scanned object for 360 degree. At the same time, at the opposite side of the X-ray source, there is a sensor which can record the energy of X-ray after penetrating the object. For each scan, a 2D X-ray image data can be obtained. After the 360 degree successive scans, the backward reconstruction process will combine all the 2D image data and reconstruct a 3D volumetric dataset such that each position has a density value related to the density of the scanned materials.

Using the volumetric dataset, both 3D and 2D information, both internal and outline profile can be obtained. From every view direction, the volume can be viewed as an array of 2D images and this enables the "cut plane" operation for volume. Further more, we can take out only part of the volume which we are most interested in for a more detailed study. Another important operation of volume data is "windowing", i.e., select a density range and only display the voxels whose values are within this density range. Using windowing method, different tissues can be displayed separately. Compared with traditional anatomical method, CT scan is a non-invasive approach which can save a lot of information that will be lost during the process of slice cutting.

Although X-ray sources are easy to get and X-ray devices are widely used, long time and excessive exposure under X-ray is harmful. When high resolution scan or the whole body scan is needed, scanning time will be quite long and using CT is not a very safe choice. In this situation, harmless MRI comes into being.

2.3.3 Magnetic Resonance Imaging (MRI)

In 1971 Raymond Damadian found that the nuclear magnetic relaxation times of tissues and tumors differed and from then on great interests are caused to study magnetic resonance in medical applications. On July 3, 1977, the first MRI examination is performed on a human being. MRI images are based on spatial variations in phase and frequency of the radio frequency energy being absorbed and emitted by objects. Because hydrogen nuclei have a nuclear magnetic resonance signal, and 63% of the composition of human body is hydrogen atoms, MRI imaging primarily images the nuclear magnetic resonance signal from the hydrogen nuclei. MRI imaging is more suitable to soft tissues and compared with X-ray, MRI imaging has no radiotoxin.

When MRI scanning is performed, a horizontal tube runs through the magnet from front to back. A 2D or 3D map of different tissue types are built up when the MRI system goes through the patient's body and after the scanning the integrated information can give out a 2D image or a 3D volume. By changing the exam parameters, the MRI system can cause tissues in the body to take on different appearances. If there is an abnormal appearance, possibly a tumor has appeared and focus will be put on the abnormal part. MRI is a growing technique and in 1993 functional MRI was developed to map the function of various regions of human brain.

2.3.4 Micro Computed Tomography (MCT)

The resolution of traditional CT scanner is not high enough to some tiny structures such as a tooth or trabecular bone. In this situation, high resolution Micro CT (uCT) scanner came into being. The basic technique of Micro CT is the same as traditional CT. Scanco Medical and SkyScan are two famous Micro CT providers. Data used in this thesis came from the

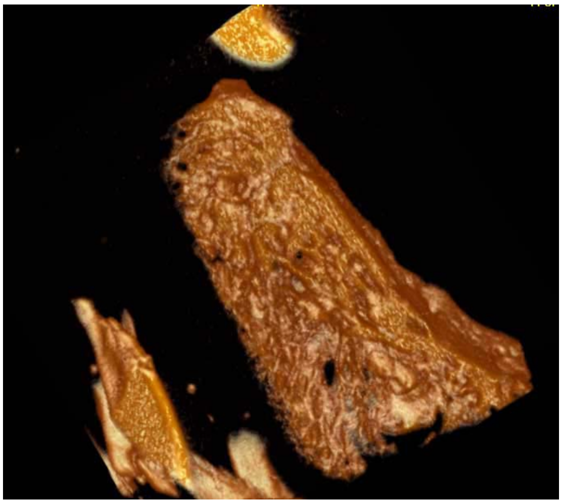
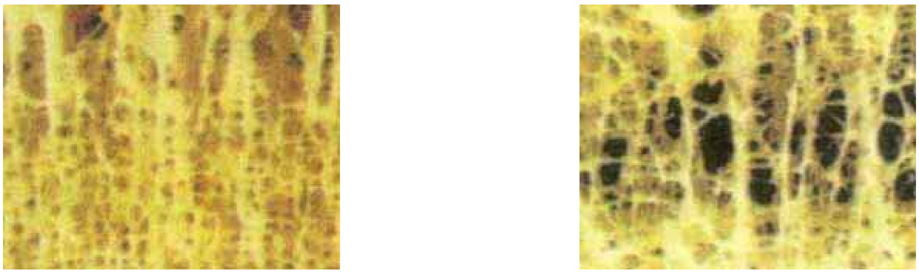


Figure 2.6
Trabecular Bone displayed by VolumPro

Scanco's uCT 40 scanner whose resolution is 20 µm. The maximum diameter it can scan is 38mm and the maximum length is 80mm. Specimen can be analyzed precisely both quantitatively and qualitatively. Using the volume data, detailed tiny structures can be presented (Figure 2.6).

2.4 Osteoporosis and Trabecular Bone

When a person is born, the process of bone growing begins. Calcium, phosphate and other minerals are absorbed by bone and at the age of 25-30 years old, the density of bone reaches its maximum. From then on, the bone growing rate is slower than bone loss rate. With the process of bone mass decrease and bone structure deterioration, osteoporosis will occur (Figure 2.7). Osteoporosis is the reason why some old people have bone facture without any obvious fall or pressure. Osteoporosis has affected people's health seriously and attracted more and more attentions.



(a) Normal Bone

(b) Bone with Osteoporosis

Figure 2.7 Bones with different quality
Pictures come from Health Plus Magazine

Skeleton is composed of cortical bone and trabecular bone. Cortical bone is the cover of trabecular bone and is mainly contained in limbs. Trabecular (also named as spongy or cancellous) bone is mainly contained in the axial skeleton and the ends near the joints of the long bones, such as rib cage, femur, radius and backbone. The structure of trabecular bone is a lattice of plates and struts [19] (Figure 2.8). This kind of structure provides mechanical strength to the skeleton. Although the quantity of trabecular bone is mainly determined by the bone density, structure of the trabecular bone also contributes significantly to the strength [2]. So the study of the topology of trabecular bone is important.

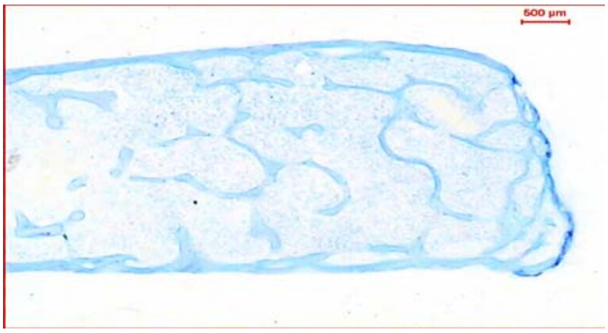
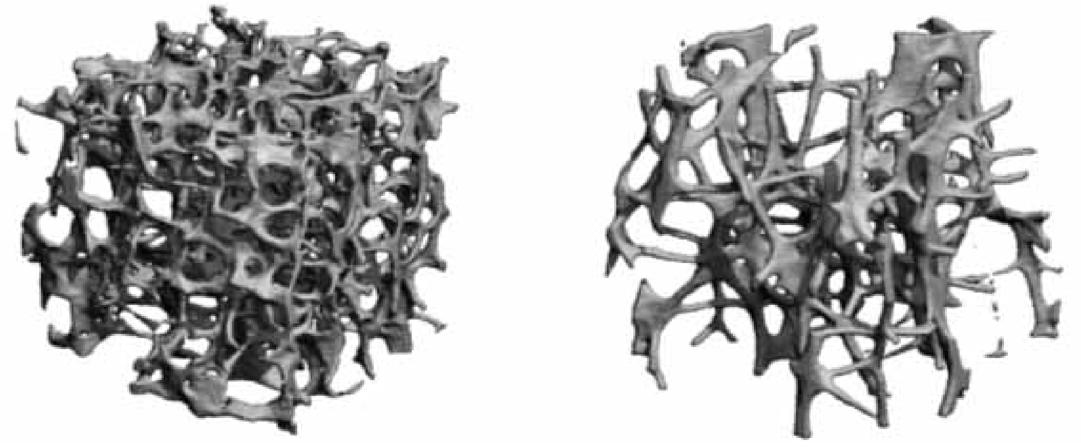


Figure 2.8 A 2D slice of trabecular bone (Courtesy of Dr. W.W.Lu, HKU Orthopedic Surgery Department)

Trabecular bone consists of a lattice of interconnected plates and rods. The bone in Figure 2.9(a) is more plate-like and that in (b) is more rod-like. Intuitively (b) is looser than (a), bone mass in (a) is more than that in (b) and (a) is stronger than (b). If examine their bone density, we can get the same conclusion.



(a) Plate-like Trabecula

(b) Rod-like Trabecula

Figure 2.9 Trabecular Bone Structure
Courtesy of SCANCO Medical AG

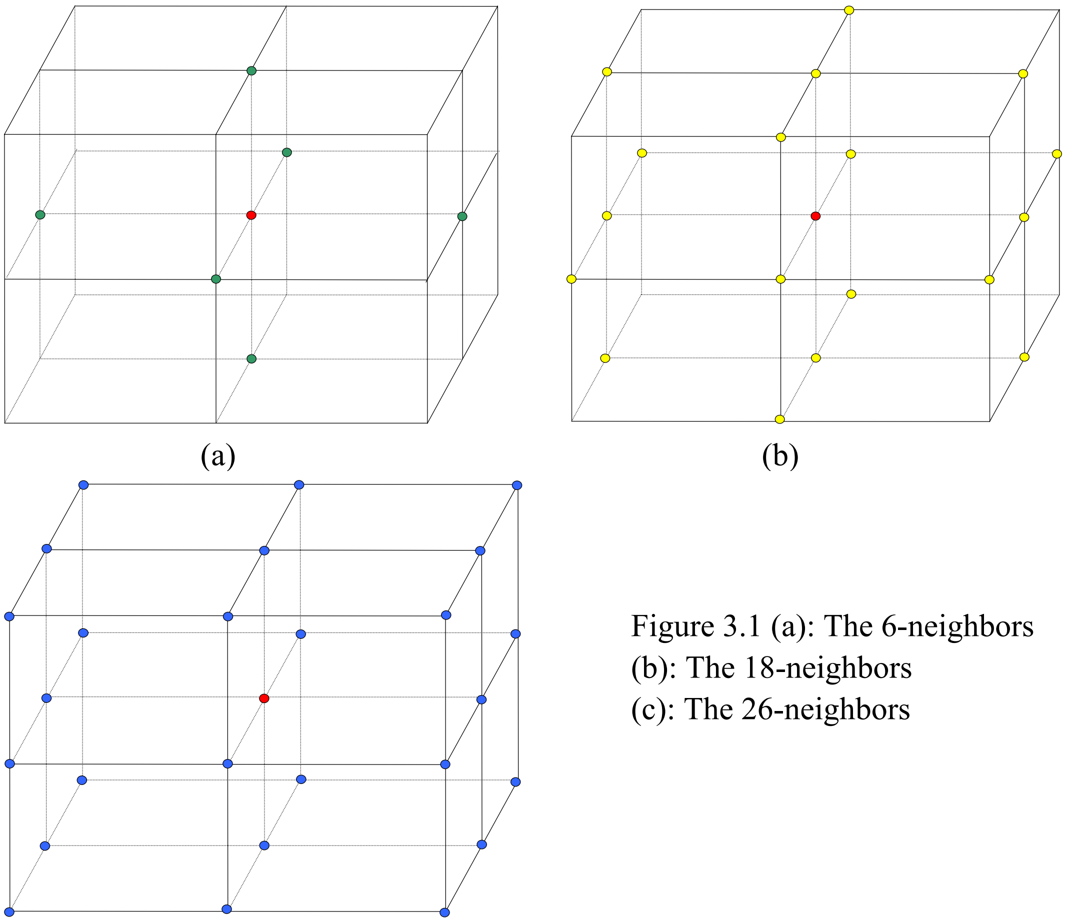
Chapter 3

Topology Related Algorithms

In order to understand the complex structure of trabecular bone better, we will focus on the topology of bones. Using topological analysis, properties related to position, connectivity etc. can be obtained in a mathematical way. Our project is set up on the basis of digital topology and a series of topology related algorithms, such as image thinning, voxel classification and components segmentation. In this chapter, basic concepts of digital topology related to these algorithms are introduced.

3.1 General Definitions

First of all, definitions of adjacency relations in 3D digital images [20] are presented: Each voxel is a lattice point in 3D with coordinates (*x,y,z*) and a scalar value. In Figure 3.1 (a) green voxels are 6-adjacent to the red voxel and exactly one coordinate element of each green voxel differs from the corresponding coordinate element of the red voxel. In (b), the yellow voxels are 18-adjacent to the red one and two of their coordinate elements can be differed by one. In (c), the blue voxels are 26-adjacent to the red one and all three coordinate elements can be different by one. If two voxels are ∂-adjacent to each other, they are ∂-neighbors.



(a)

(b)

(c)

Figure 3.1 (a): The 6-neighbors
(b): The 18-neighbors
(c): The 26-neighbors

Suppose there is a voxel sequence $p_0, p_1, ..., p_n$ in a set of voxels *S*. If p_i is ∂-adjacent to p_{i+1}, $(0 \leq i < n)$, there is a ∂-path between p_0 and p_n. If $p_0 = p_n$, it is a ∂-closed path. If a set of voxels *S* cannot be separated into two subsets S_1 and S_2, which are not ∂-adjacent (if one voxel in S_1 and one voxel in S_2 are ∂-adjacent, S_1 and S_2 are called ∂-adjacent.), *S* is ∂-connected. There must be a ∂-path between any two voxels in a ∂-connected set. A ∂-component of *S* is the maximum ∂-connected subset of *S*.

Let N(*p*) denote a voxel *p* and all its 26-neighbors (N(*p*) has 27 voxels) and N*(*p*) denotes only the 26-neighbors but not *p* (N*(*p*) has 26 voxels).

In [20], a definition of conventional digital picture is given as a quadruple (V, m, n, B), where V = Z^2 stands for a 2D image or V = Z^3 for a 3D image. B is the black subset of V. Black voxels are adjacent if they are m-adjacent and white voxels are adjacent if they are n-adjacent. Black component is a set of m-adjacent black points and white component is a set of n-adjacent white points. In this thesis, black component is 26-connected and white component is 6-connected.

In a (Z^3, 26, 6, B) digital picture, if a black voxel has no black 26-neighbors, this voxel is isolated. If a black voxel is surrounded by black ones, i.e., the 26-neighbors of a black voxel are all black, this is an *interior voxel*; otherwise it is a *border voxel*. If a white voxel is surrounded all by black voxels, this white voxel forms a *hole* (*cavity*) inside the black component.

The *Euler characteristic* of a 3D polyhedral set *S* is equal to the number of black components plus the number of cavities minus the number of tunnels in *S*. How to define a *tunnel* rigorously is a hard problem, although it can be visualized and described intuitively. If there is a solid handle in an object, there must be a tunnel through this handle and if the handle is broken, this tunnel will disappear. Saha [21] presents a computational method to calculate the number of tunnels in N*(*p*) and we will discuss it later.

A black point is a *simple point* if its deletion will not affect the topology of a digital image. The characteristics of a simple point *p* in a 3D digital image (Z^3, 26, 6, B) are [20]:

(1) *p* is a black point.

(2) *p* is adjacent to just one black component of N*(*p*).

(3) *p* is adjacent to just one white component of N*(*p*).

(4) Number of tunnels in N(*p*) and N*(*p*) are the same.

3.2 3D Thinning Algorithm

3.2.1 Basic Concepts

Thinning operation is the process of changing the original object to its "skeleton" representation by deleting object points and at the same time maintaining topology. Thinning operation is often used in pattern recognition, for the "skeleton" is the most meaningful part in recognition. For 2D images, Blum [23] proposed the medial axis transformation method which is analogous to thinning and Tsao et al. [24] also used discrete medial axis in their thinning algorithms. Hilditch's [25] thinning algorithm based on connectivity preservation when a black point is deleted. Stefanelli and Rosenfeld [26] designed rules for 2D topology preservation and the rules have become criterion to detect whether a black point is a simple point. There is not much literature that can be found on 3D image thinning. Saha [27] summarized Lobregt et al.'s Euler characteristic preserving method, Morgenthaler's end points algorithm, Srihari et al.'s path connectivity thinning, Tsao and fu's surface connectivity method and some other algorithms that extended the 2D thinning technique to 3D.

Lobregt et al. [28] designed the thinning algorithm based on the invariance of Euler number when deleting voxels. Euler number can represent the connectivity characteristic of 3D objects. Their algorithm ensures the invariance of Euler number by restricting that the Euler number in N(p) does not change when a voxel is deleted. But their algorithm has problem when this deletion adds one black component and a tunnel at the same time (Figure 3.2). In this situation, the Euler number does not change, but the topologies are not the same.

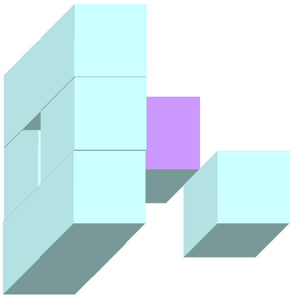


Figure 3.2
Deletion of the purple voxel will add one component and tunnel at the same time.
The Euler number does not change.

Saha in [27] proposed his thinning algorithm that not only can preserve topology, but also can maintain shape. His algorithm uses local topological information but not the global topological information. We first introduce the definitions used in this algorithm:

Suppose p is the center voxel in a 3x3x3 cube,

s-point: voxel *p*'s 6-neighors (*p* has 6 s-points in N(*p*)).

e-point: voxel *p*'s 18-neighbors but not 6-neighbors (*p* has 12 e-points in N(*p*)).

v-point: voxel *p*'s 26-neighbors but not 18-neighbors (*p* has 8 v-points in N(*p*)).

opposite s-points: two s-points in N(*p*) which are not 26-neighbors.

Suppose *a,b,c,d,e,f* are the s-points in N(*p*). We assume the arrangement such that (*a*,*d*), (*b*,*e*) and (*c*,*f*) are three pairs of opposite s-points.

e(a,b,p) = *q*, then *q* is the voxel that is both 6-adjacent to *a* and 6-adjacent to *b*. Hence, *q* is an e-point.

v(a,b,c,p) = *q*, then *q* is 6-adjacent to *e(a,b,p)*, *e(b,c,p)* and *e(a,c,p)*. Hence, *q* is a v-point.

$f_1(a,p)$ = *q*, then *q* is 6-adjacent to *a* and *q* is not in N(*p*). Five of *a*'s 6-neighbors are in N(*p*), so *q* is unique.

$f_2(a,b,p)$ = *q*, then *q* is 6-adjacent to $f_1(a,p)$ and *e(a,b,p)*.

$f_3(a,b,p)$ = *q*, then *q* is 6-adjacent to $f_2(a,b,p)$ and $f_2(b,a,p)$.

surface(a,p) is the 3x3 voxels which are in a plane passing through *a* but not *p*.

From Figure 3.3, we can get a clear understanding of these definitions.

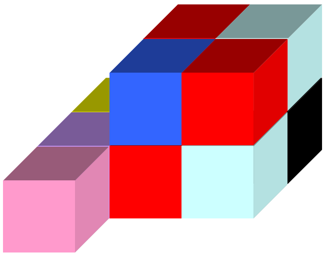


Figure 3.3
Black voxel is *p* and three light blue voxels are s-point *a*,*b*,*c* (a is blocked; the front one is b; the top one is c). Then three red voxles are *e(a,b,p)*, *e(a,c,p)*, *e(c,b,p)*. Dark blue one is *v(a,b,c,p)*. Yellow voxel is $f_1(a,p)$. The purple one is $f_2(a,b,p)$. pink voxel is $f_3(a,b,p)$.

Middle plane *M(a,d,p)* is the surface exclude *p* which passes through *p* and perpendicular to the line *ad* (contains 8 voxels). An extended middle plane *EM(a,d,p)* is a surface passing through *p* but not including *p* which contains both the middle plane *M(a,d,p)* and some extended voxels. The extension rules are as follows: if the middle plane contains the bottom and west voxels of *p*, the extended voxels are in the bottom and west direction of the middle plane (Figure 3.4 (a)); In Figure 3.4 (b), *M(a,d,p)* contains the bottom and south voxels of *p* and the extended voxels are in the bottom and south direction of the middle plane; In Figure 3.4 (c), extended voxels are in the south and west direction of the middle plane because *M(a,d,p)* contains the south and west voxels of *p*. There are 15 voxels in *EM(a,b,p)*.

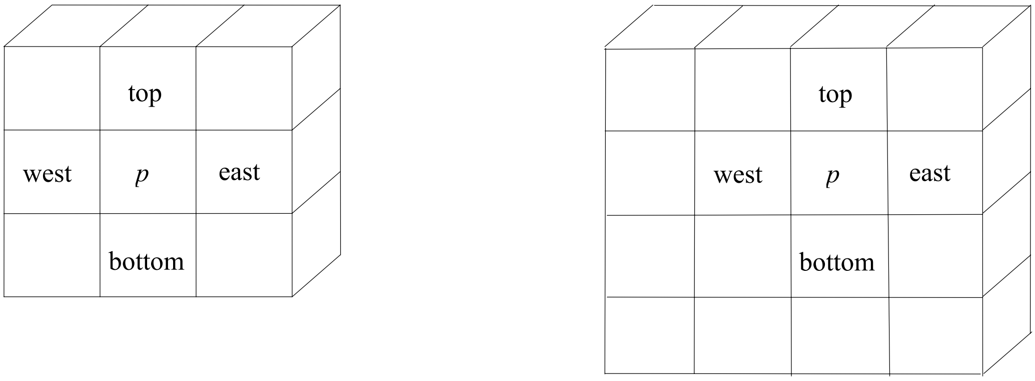


Figure 3.4 (a) Middle Plane and Extended Middle Plane

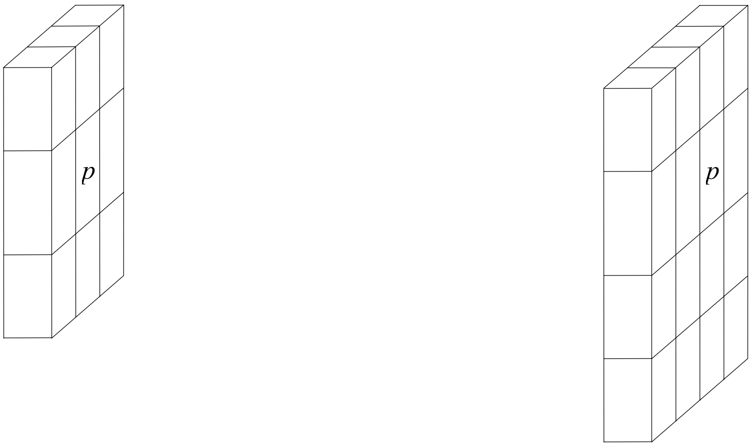


Figure 3.4 (b) Middle Plane and Extended Middle Plane

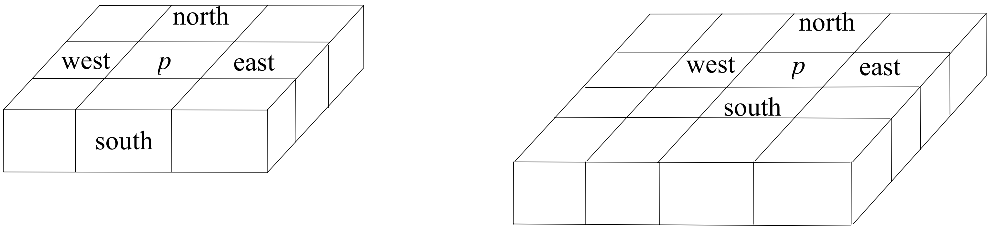


Figure 3.4 (c) Middle Plane and Extended Middle Plane

Saha proposed the definition of a *simple point* in N(*p*):

(1) *p* is a black voxel.

(2) *p* has at least one black 26-neighbor.

(3) *p* has at least one white 6-neighbor.

(4) all the black neighbors of *p* are connected.

(5) the set of white 6-neighbors of *p* is 6-connected in the set of white 18-neighbors of *p*.

s-open point: if a black voxel *p* has at least one white 6-neighor, *p* is an s-open point.

e-open point: should satisfy

(1) *p* is not an s-open point.

(2) *p* has an *e(a,b,p)* which is white.

(3) $f_1(a,p)$ and $f_1(b,p)$ are black.

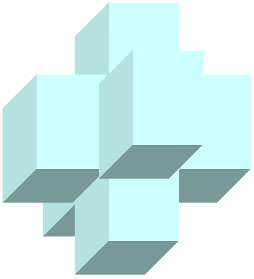


Figure 3.5 illustration of *e-open point*

For example in Figure 3.5, the center point is an *e-open point*.

v-open point: (1) *p* is neither an s-open point nor an e-open point.

(2) *p* has a white *v(a,b,c,p)* point.

(3) $f_1(a,p)$, $f_1(b,p)$ and $f_1(c,p)$ are black.

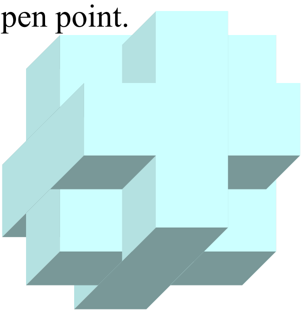


Figure 3.6 illustration of *v-open point*

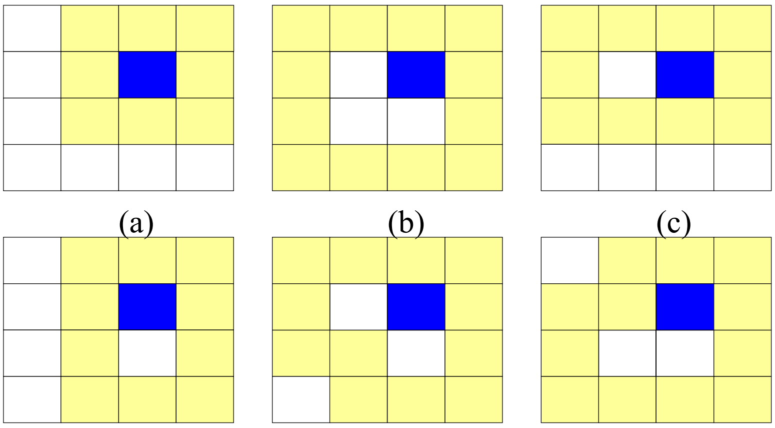
In Figure 3.6, the center point is a *v-open point*.

Arc-like point: (1) *p* is a black voxel.

(2) In N(*p*) there is a pair of opposite s-points (*a*,*d*), surface(*a*,*p*) and surface(*d*,*p*) both at least contain one black voxel. *p* is 26-adjacent

to all the voxels in surface(*a,p*) and surface(*d,p*), so this condition satisfied that the arc is at least 3 voxels long.

(3) The extended middle surface *EM(a,d,p)* contains a white 6-closed path that surrounds *p*, this path can be of any one case in Figure 3.7 (blue voxel is p, yellow voxels forms the path and other voxels in Figure 3.7 can be either black or white point). The definition of arc-like point tells us that the diameter of an arc is at most 2 voxels and the length of an arc is at least 3 voxels.


(a) (b) (c)

(d) (e) (f)

Figure 3.7 white 6-closed path surrounded *p*

surface-like point: (1) *p* is a black voxel.

(2) There are a pair of opposite s-open points (*a,d*) in N(*p*) and suppose *d* is a voxel on the bottom or west or south of *p*. *a* is white, *d* or $f_1(d,p)$ is white.

(3) Each voxel *q* in the middle plane *M(a,d,p)* and two of *q*'s opposite s-points which out of *M(a,d,p)* forms a column. Each column at least has one black voxel.

From the definition of surface-like point, we know that the "surface" in this algorithm is at least 3x3 voxels and the levels of voxels in a surface are at most two.

Both arc-like point and surface-like point are called *shape point*.

thick(a,d,p) = 1 if *a* and $f_1(d,p)$ are white while *b*,*c*,*d* are black before an iteration;

otherwise *thick(a,d,p)*= 0;

erodable point p is a simple point and should satisfy one of the three conditions:

(*a,d*), (*b,e*) and (*c,f*) are pairs of opposite s-points and *d,e,f* are in the bottom, south and west directions of voxel *p*.

(1) *thick(a,d,p)* is 1 and black points in *M(b,e,p)* and *M(c,f,p)* are one black component without tunnel.

(2) *thick(a,d,p)* and *thick(b,e,p)* are 1 and black points in *M(c,f,p)* forms one black component without tunnel.

(3) *thick(a,d,p)*, *thick(b,e,p)* and *thick(c,f,p)* are 1.

After introduction of concepts used in this 3D thinning algorithm, the thinning process is described as follows:

3.2.2 Thinning Process

This thinning algorithm is an iterative process that simple points are deleted and shape points are maintained in the outer-layer during each iteration. It includes two steps: primary-thinning and final-thinning.

The process of primary thinning includes three steps:

```
Step 1: for every unmarked s-open point in volume
            if it is a shape point, mark this point;
                else if it is a simple point, delete it;
                    else do nothing;
Step 2: for every unmarked e-open point
            if it is a simple point and deletion of this point will not effect the 2D
            topology in the three axis-parallel planes passing through this point,
            delete this point;
Step 3: for every unmarked v-open point
            if it is a simple point, delete this point;
```

If in step 1, 2, 3 there are still points being deleted, then repeat step 1, 2, 3.

At last in final-thinning, a black point is deleted if it is an erodable point.

A medial surface representation of the original image can be got after the thinning process and this "skeleton" representation will be used in the following classification phase.

3.3 Classification

Generally, classification often refers to the classification of different tissues, such as bone and soft tissues, normal organ and abnormal organ etc. This kind of classification often can be realized by their densities [29]. However, for those materials with densities whose differences are not very obvious, classification cannot be done well only by densities. Shape and architecture of different materials are often useful aids. In our project, the focus is trabecular bone and it is composed by rod-like and plate-like structures. Although soft tissue is also contained in the trabecular bone Micro CT data, they can be easily distinguished by densities. Classification mentioned in this thesis refers to the classification of different shapes, that is to distinguish the rod-like and plate-like architectures.

Saha [21] proposes a method that classifies each black voxel in the surface skeleton image to one kind of topological type. This method is applied to Trabecular Bone MR Images [22]. Firstly, the 3D volume data is converted into a surface skeleton representation (the thinning process). Now each black voxel either belongs to a surface or a curve structure. Secondly, a classification method is applied to the surface skeleton image based on local topological (in a voxel's 3x3x3 neighborhood) analysis. After this step, each voxel is give

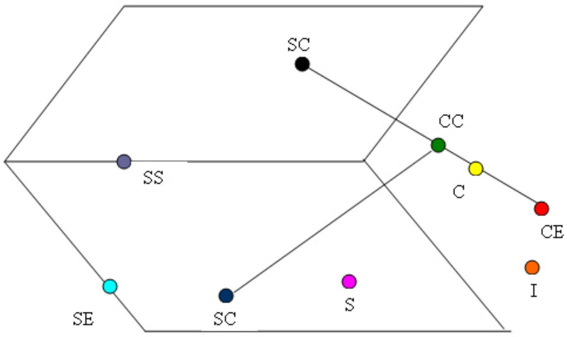


Figure 3.8 Illustration of voxel types

one topological type (Figure 3.8) of I-type (isolated), C-type (curve), CE-type (curve edge), S-type (surface), SE-type (surface edge), CC-type (curve-curve joint), SS-type (surface-surface joint) or SC-type (surface-curve joint).

Tunnels, cavities and components are characteristics of both local and global topology. If an empty space surrounded by black voxels, there is a cavity. In each place there is a handle, a tunnel must be created. To calculate the number of tunnels is not an intuitive job and the following section gives out the calculation method.

3.3.1 Jordan's Curve Theorem

Jordan's Curve Theorem describes a fundamental topological property. The Jordan's Curve Theorem [30] states that a simple closed curve in a 2D plane separates the plane into two parts. The interior part is bounded by the closed curve and the exterior part is unbounded (Figure 3.9). In 3D, if a black subset of a topological sphere forms a simple loop, this black subset separates the sphere into two parts and creates one tunnel. The number of tunnels equals to the number of separated parts minus one. In the local topological definition, N*(p) is analogous to the sphere, **B** is black voxels in N*(p), X(p) is the 6-adjacent white voxels in N*(p) and Y(p) is the 18-adjacent white voxels in N*(p). Saha gave out the theorem of calculating tunnel numbers as follows: *Thoerem. If X(p) is nonempty, the number of tunnels, is one less than the number of white components of Y(p) that intersects X(p) (intersect X(p) means a white component contains a voxel that is 6-adjcent to p), or zero otherwise.*

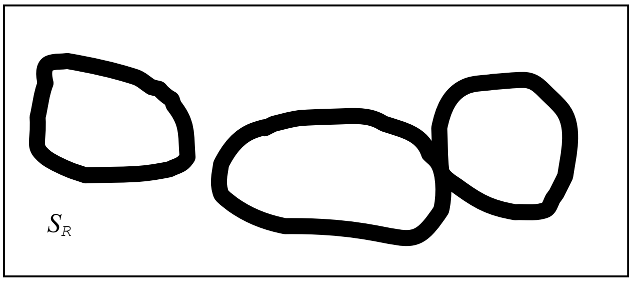


Figure 3.9
The black components form 3 simple loop and separate S_R into 4 parts. 3 tunnels are created. Analogize to 3D.

3.3.2 Classification Process

Let *ξ(p)*, *η(p)* and *δ(p)* represent the number of black components, tunnels and cavities in N*(*p*), Saha's classification method [21] can be described as follows:

In N*(*p*), *p* has six 6-neighbors and totally has $2^6 = 64$ configurations of 6-neighbors. Due to symmetry, these 64 configurations can be grouped into 10 classes

Class 0: all 6-neighbors are black.

Class 1: five 6-neighbors are black.

Class 2: two pairs of opposite 6-neighbors are black.

Class 3: one pair of opposite 6-neighbors and two non-opposite 6-neighors are black.

Class 4: one pair of opposite 6-neighbors and another 6-neighbor are black.

Class 5: three non-opposite 6-neighbors are black.

Class 6: one pair of opposite 6-neighbors is black.

Class 7: two non-opposite 6-neighbors are black.

Class 8: only one 6-neighbor is black.

Class 9: no 6-neighbor is black.

In each class, *ξ(p)*, *η(p)* and *δ(p)* is calculated separately. Using table 3.1, each voxel *p* is primarily classified.

ξ(p)	*η(p)*	*δ(p)*	Initial Type	Possible Classification
0	0	0	T_1	I-type
1	0	0	T_2	SE-type or CE-type
2	0	0	T_3	C-type
>2	0	0	T_4	CC-type
1	1	0	T_5	S-type or CC-type
>1	>=1	0	T_6	SS-type, SC-type or CC-type
1	>1	0	T_7	SS-type, SC-type or CC-type
1	0	1	T_8	SS-type, SC-type or CC-type

Table 3.1 Initial Classification.

After the initial classification, further decision should be made to uniquely determine each voxel's type. In this process, the type of 26-neighbors of a voxel is consulted. Using table 3.2, the type of *p* is uniquely determined.

Initial type	26-neighborhood type	Initial classification
T_2	Only one black neighbor	CE-type
T_2	More than one black neighbor	SE-type
T_5	All black neighbors are T_3 or T_4	CC-type
T_5	Not all black neighbors are T_3 or T_4	S-type
T_6	All black neighbors are T_3 or T_4	CC-type
T_6	Not all black neighbors are T_3 or T_4	SC-type
T_6	No black neighbors is T_3 or T_4	SS-type
T_7	All black neighbors are T_3 or T_4	CC-type
T_7	Not all black neighbors are T_3 or T_4	SC-type
T_7	No black neighbors is T_3 or T_4	SS-type
T_8	All black neighbors are T_3 or T_4	CC-type
T_8	Not all black neighbors are T_3 or T_4	SC-type
T_8	No black neighbors is T_3 or T_4	SS-type

Table 3.2 Second Classification

3.3.3 Successive Corrections

After the initial and second classification, each voxel's type is uniquely determined. And in the final step some successive corrections are made.

3.3.3.1 Extension of SS Lines

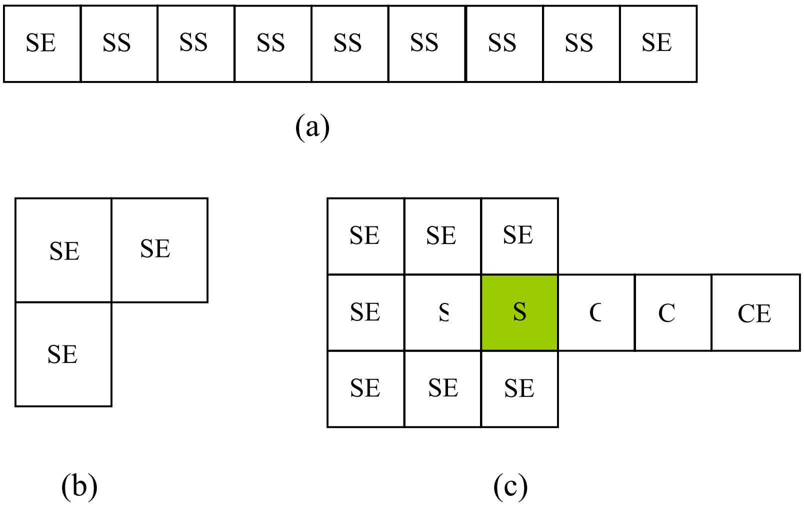
When two surfaces are connected, a serial of SS-type voxel are identified. But the two ends of the junction line are still SE-type (Figure 3.10(a)). In the final step, these end points are reclassified as SS-type points.

3.3.3.2 Profile Type Detection

In Figure 3.10(b), each voxel is identified as SE-type, but intuitively they are not the edge of a surface. SE-type voxels without S-type, SC-type or SS-type 26-neighbors are changed to P-type (profile-type). P-type voxels always appear in pair and a pair of P-type voxels can be taken as one C-type voxel.

3.3.3.3 SC Type Reclassification

In Figure 3.10(c), obviously the green voxel is the joint of a surface and a curve, but this voxel is classified as an S-type voxel. In order to resolve this problem, the neighbors of a C-type or CC-type voxel *p* is examined, putting all the S-type, SC-type and SS-type voxels in N*(*p*) (do not care about other types in N*(*p*)) into a set *P* and let $P_1, P_2, \ldots P_n$ be the disconnected 26-components of *P*. If P_i does not have any SC-type or SS-type voxel, the nearest voxel in P_i is identified as SC-type (the green one).
(The distance of two voxels here is defined as the distance of the centers of these two voxels.)



(a)

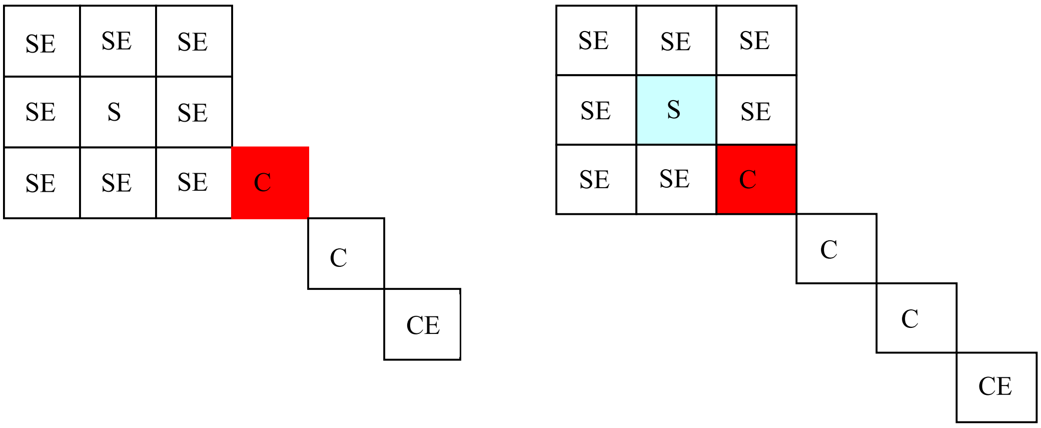
(b)

(c)

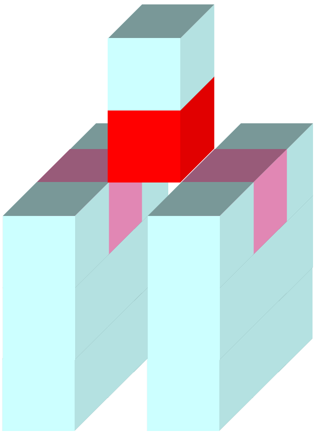
Figure 3.10 Successive Correction

3.3.3.4 Other Modification of Mine

During the process of surface-curve junction identification (Section 3.3.3.3), I find there are still some situations which need to be modified. In the structure topologically equals to Figure 3.11(a), using Saha's method, no voxel can be identified as SC-type and in the structure topologically equals to Figure 3.11(b), the blue voxel with initial type S is changed to SC-type as a result, but intuitively I think the voxel is not appropriate to be determined as a surface-curve junction and the red one seems more like a SC type voxel. In Figure 3.11(c), the middle-top voxels (pink ones) on each surface are identified as SC-type by Saha's method, but I think if the red voxel is classified as SC-type and the two pink voxels are maintain SE-type are more suitable. So I modified Saha's method: examining the neighbors of each C-type or CC-type voxel *p*, if there is any S-type or SE-type voxels, the voxel *p* is reclassified as SC-type. So red voxels in Figure 3.11(a), (b) and (c) are reclassified as SC-type. This change will not affect other voxel's type or the following segmentation operation and is more appropriate.



(a) (b)



(c)

Figure 3.11 Modification of SC-type

3.4 Segmentation Operation

After the thinning operation and the classification operation, every black voxel on the skeletonized surface representation is associated with a topological type. And using the type information, segmentation operation can be applied to the skeletonized bone structure. This operation can be used to identify every rod-like and every plate-like structure of trabecular bone. After the thinning operation, the rod-like part is changed to curve in the surface skeleton representation and plate-like part is changed to surface. Mentioned in 3.3, the topological type includes I, C, CE, S, SE, CC, SS, SC, P types. When an I-type is encountered, we can just discard it. When a C-type voxel is encountered, two 26-adjacent C-type voxels are contained in N*(p), so we extend the C-type voxels until a CE, CC or SC type point is detected. Two end voxels and the interior C-type voxels form a curve. In the same way, a surface can be formed. Although we can use this seed-tracing method to separate every curve and surface, this iteration method is time-consuming.

Saha [31] proposed the segmentation method based on his classification algorithm[21]. Let us first introduce the symbols used in the method before describing the algorithm.

For a set of black points *S*, the 26-envelope of *S* $\varepsilon(S)$ is defined as $\varepsilon(S) = \bigcup_{p \in S} N(p) - S$. Let *S* be the set of all black voxels in the surface skeleton representation and *J* be the set of all junction points which include SS-type, SC-type and CC-type voxels in *S*.

The segmentation method is a 3-steps process:

First step: let $S^{'} = S - (\varepsilon(J) \bigcup J)$, then each 26-component in $S^{'}$ is an isolated voxel, a curve or a surface. Curves or surfaces are separated from the junction place. If only the set *J* is deleted, the connected parts are remained connected in some situations. Removing not only the junction points, but also the envelope of all the junction points ensures that the connected parts can break completely, although this deletion removes some points which belong to the curve or surface. So the next two steps will extend the separated parts to their original shape as much as possible.

Second step: add the 26-envelope voxels, which are originally black, of each separated component. In this step, the excessively deleted part in step 1 is made up.

Third step: add the junction points which are 26-adjacent to the subset got from step2.

3.5 Euler Characteristic

The Euler number of 3D object equals to the number of object components pluses the number of cavities minus the number of tunnels. One minus the Euler number means the minimum number of connections that a connected part can break before it separates into two parts. When a black point is deleted or a white point is added, the Euler number in N(*p*) will change accordingly and this change will affect the Euler number in the whole object. So the Euler number of a volume dataset can be calculated form each voxel's local topological property. Saha [32] gave out a method to calculate the Euler number. When a black voxel *p* is deleted, the change of Euler number in its 3x3x3 neighborhood equals to E(N(*p*))-E(N*(*p*)) (E stands for Euler number). Repeating the deletion process until there is no black voxel left, the Euler number of the empty volume is 0. Back to the original volume, the Euler number is the sum of the change of every deletion. And in N(*p*), the number of object is always 1 with no tunnel and no cavity. So E(N(*p*)) = 1 – 0 + 0 = 1. We only need to calculate E(N*(*p*)), i.e., the number of objects $\xi(p)$, the number of tunnels $\eta(p)$ and the number of cavities $\delta(p)$ in N*(*p*). Using the knowledge in 3.3, these numbers can be computed. The algorithm is as follows:

```
A 3D digital image P = (V,26,6,B)
At first E(P) = 0;
For (each black voxel p in P)
{
    E(P) = E(P) + 1 - $\xi(p)$ + $\eta(p)$ - $\delta(p)$;
    B = B - p;
}
```

Bischoff [33] introduces another algorithm to calculate the Euler number. A voxel can be viewed as a cell complex which contains 1 cube (3-cell), 6 surfaces (2-cell), 12 edges (1-cell) and 8 vertices (0-cell). A volume dataset can be viewed as the sum of all the object voxels in it and the overall cell complex is the sum of each object voxel's cell complex. When overlap n-cell elements are encountered, they are only counted once. For example in Figure 3.12, this volume dataset contains 2 cubes, 11 surfaces (one surface overlaps), 20 edges (four edges overlap) and 12 vertices (four vertices overlap).

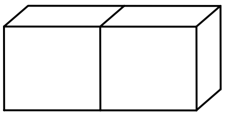


Figure 3.12 Cell Complex

Euler number also can be calculated based on cell complex theory as the number of even dimensional cells minus the number of odd dimensional cells. Using N_i to represent the number of i-cell, the formula of Euler number can be written as this: $E(S) = N_0 - N_1 + N_2 - N_3$, i.e., the number of vertices in this dataset minus the number of edges plus the number of surfaces minus the number of object voxels. In Figure 3.12 $E(S) = 12 - 20 + 11 - 2 =1$. The result coincides with the calculation method using the number of objects minus the number of tunnels plus the number of cavities.

How to calculate N_i? Counting directly is straight-forward but not very easy, because the overlap elements should be counted only once. In [33] the method used is: Scan the volume dataset row by row, line by line, slice by slice. A voxel is defined as a solid cube and at every vertex of the cube we can suppose a 2x2x2 voxels pattern (Figure 3.13) is placed. The pattern's center and the vertex are overlapped. Suppose there is a red cube located at the center of the pattern and with the same size of a voxel. Then we will count how many cubes, surfaces, edges and vertices the red cube intersects with the black voxels in the 2x2x2 pattern. Totally there are $2^8 = 256$ cases in this pattern (each voxel can be either a black voxel or a white one). For example in Figure 3.14, at the vertex that the two voxels share, N_0, N_1, N_2, N_3 are 1, 6, 6, 2 and at any other vertices, N_0, N_1, N_2, N_3 are 1, 3, 3, 1. At every vertex, the cell complex fractionally contributes to the overall Euler number, i.e., $N_0/2^0$, $N_1/2^1$, $N_2/2^2$, $N_3/2^3$ are used. So the Euler number in Figure 3.14 is $1/2^0 - 6/2^1 + 6/2^2 - 2/2^3 + 14(1/2^0 – 3/2^1 + 3/2^2 - 1/2^3) = 1$. It is the same as the number of objects – the number of tunnels + the number of cavities. A look-up table can be setup in advance for reference and during the scanning process the voxel complex can be looked up from the table using the configuration in the current 2x2x2 pattern as index.

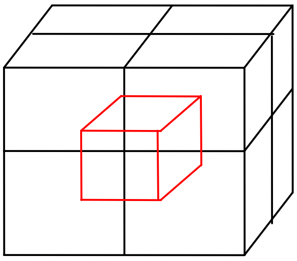


Figure 3.13 the 2x2x2 Pattern

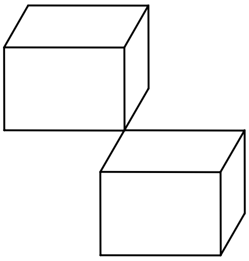


Figure 3.14 Calculating the Euler Number

In theory, these two methods of Euler number calculation should get the same result. In this project, the second method is adopted because it is easier to compute and it is efficient.

Chapter 4

Trabecular Bone Study System

Originally, the study of bone architecture is based on electronic microscopy and stereological theorems of 2D sections [34]. Obviously, there are two limitations to these methods: 1. Sections got in this way are destructed when the bone is cut into slices. 2. Connectivity in 3D cannot be rigorously inferred from isolated 2D sections. With the introduction of 3D scanning devices, non-destructive volumetric dataset can be obtained. And with knowledge of 3D topology, connectivity property can be studied directly. Using techniques in computer visualization, image processing, digital topology and medicine, a trabecular bone study system is being developed. It can be used as an aid to study the trabecular bone properties and osteoporosis. The primary purpose of our system is to extract and display rod-like structures of trabecular bone. We present a new method in this chapter for identifying the rod-like parts. The new method employed the following three techniques: main direction determination, short curves merging and thin surface detection.

4.1 System Overview

This system contains two parts: the visualization system and the analysis system and includes four phases: the data preparation phase, original data visualization phase, data analysis phase and rod-like structure extraction phase. An overview of various processes in this system is as follows:

- Obtaining the volume dataset from Micro CT scanner and convert the data to appropriate data format that can be read by some commercial hardware and software;
- Preprocessing the original data, including noise filtering and data compression;
- Displaying the volume data after the preprocessing;
- Perform thinning to the volume data and display its skeleton;
- Calculating the Euler number of the thinned data and do analysis related to osteoporosis;
- Segmenting the thinned data and recognize the rod-like part of the trabecular bone;
- Extending the rod-like part from the thinned data to original data and show the visualization result.

Detailed explanation will be given in the following sections.

4.2 Data Preparation

The first phase of this project is data preparation. Trabecular bone volume dataset comes from Micro CT scanning. The Micro CT scanner used here is Scanco Medical's uCT 40 machine. After scanning, we can get an ISQ type file which is the image data (slice sequence) and an RSQ file which is the raw data (sinogram sequence). ISQ file is reconstructed from RSQ file and contains a 512-byte file header and a file body. The header contains useful information about this data, which includes its name, index, scanner's type, thickness of each slice, distance between two slices, etc. And the most important information is the data's *x*, *y*, *z* dimensions, the maximum and minimum value in this data. Following the 512-byte header, it is the data body which includes each voxel's scalar value row by row, line by line, and slice by slice. A total of 16 bits are used to store one value, but only positive values are useful, while negative values are due to the reconstruction algorithm (shepp and logan filter) and noise in the raw data. The negative numbers are "cut" at -1000 (no values lower than -1000).

In order to display the volume data using our existing VolumePro 500 system, The ISQ data is converted to a file type that can be read by VolumePro. One kind of file types suitable to VolumePro is VOX type. This kind of structure also contains a file header and a file body. The file header also includes the useful dimension information, but 8 bits are used to store one voxel's value. So the first thing we need to do is to convert the 16-bits data to 8-bits data. To take full advantage of useful data, the following method is adopted:

(1) Noise removal: using the average value of 26 neighbors of a negative voxel to replace this noise value. Sharp negative shapes of the data distribution will be removed (Figure 4.1).

(2) After the noise removing operation, the minimum value in this dataset will change. Then we will change the minimum~muximum value to 0~255 linearly.

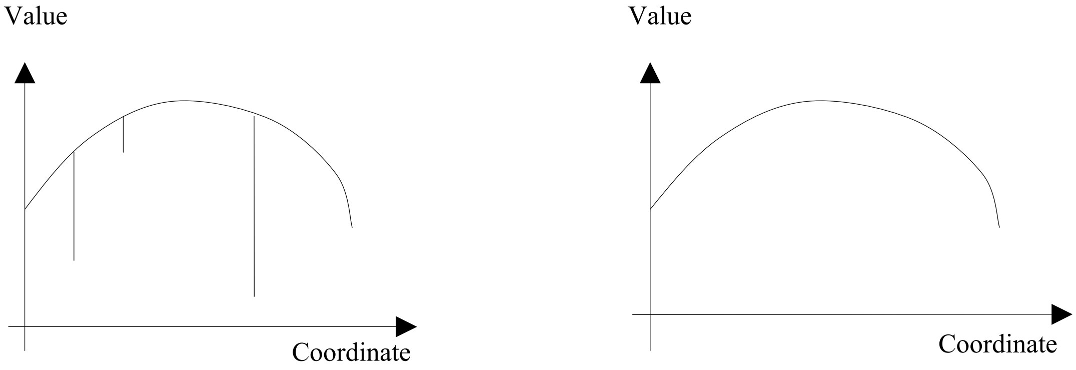
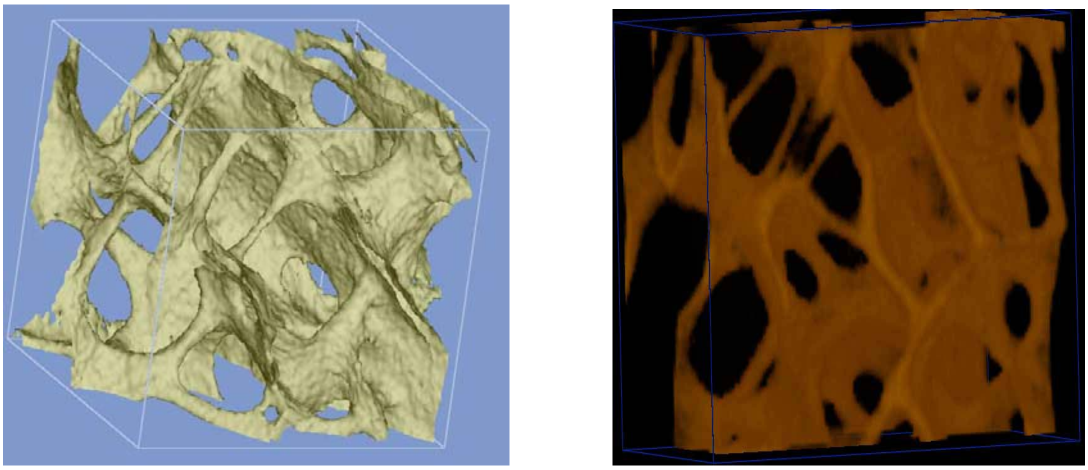


Figure 4.1 Noise Remove

4.3 Original Data Visualization

The visualization algorithm used here is the MC algorithm (mentioned in 2.2.2.1). Since we have displayed the volume data in VolumePro in which the RC algorithm is implemented, we will use MC in our project to do some visualization comparison. According to the volume data's histogram, a user-defined threshold is chosen and this threshold can define an isosurface in the volume. In the manipulation of the MC algorithm, we do not need to read the whole volume dataset into memory for sometimes the data is too big to be stored in memory. At first, four slices are read and cubes are created from two neighbor slices. After one layer of cubes are manipulated, the top slice is discarded, the other three slices are moved one layer up and a new slice is read into memory. The 4-slices structure is always kept during the MC implementation. Considering the visualization efficiency, triangles are not displayed when they are created, but are stored in a display list. When the whole volume data is scanned, this list is directly displayed. Figure 4.2 is a picture of a 100x100x127 trabecular bone displayed by MC.The number of triangles created in this data is 245400 and 7297 ms is used to generate these triangles (CPU 2.26GHz).



(a) Using MC Algorithm (b) Using RC Algorithm

Figure 4.2 Trabecular Bone Visualization

In the visualization module, several kinds of file types can be read and displayed, such as VOX, ISQ, TXT type. Subvolume extraction and saving operations can be implemented conveniently. Rotation, translation and zooming functions make observations easy and more detailed. The original dataset and the thinned dataset can be displayed interactively with the same orientation which facilitates the compare between the same place before and after the thinning operation.

4.4 Data Thinning Module

4.4.1 Close Operation

Small apertures in original dataset will be enlarged after the thinning operation (Figure 4.3).

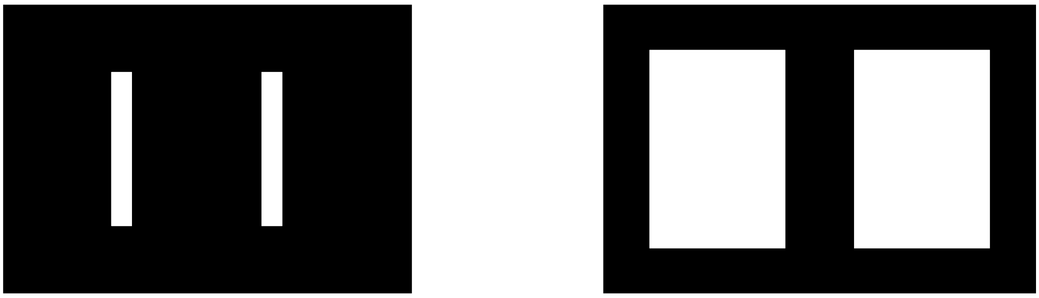
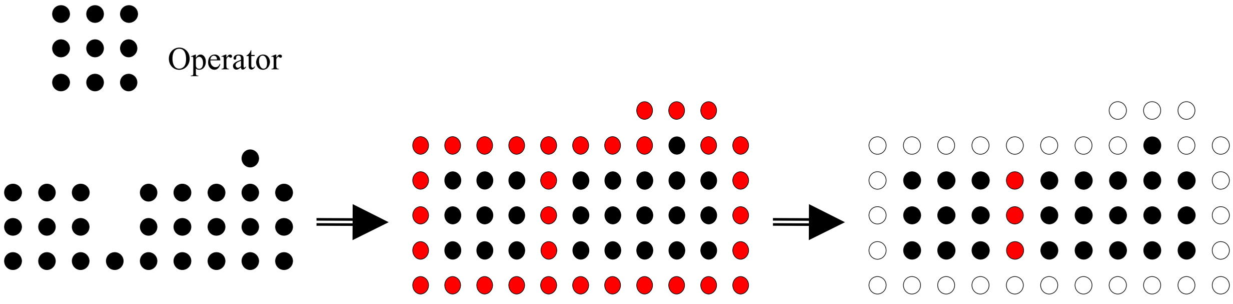


Figure 4.3 (a) Apertures in Original Data (b) Extended Apertures After Thinning

The structure in Figure 4.3 (a) is a surface-like structure although two apertures exist. After thinning, (a) changes to (b) and if the remaining bone voxels are too thin, it will be classified as curve-like structure. In order to avoid this kind of situation, image processing algorithms will be used to remove such kind of small apertures in the original dataset.

The close operation [35] includes a dilation process and an erosion process. Close operation is used to fill small breaks. Dilation operation will add pixels around edges of features. Erosion operation will remove pixels around the periphery of features in the image. Erosion is the converse of dilation. In 2D, if a 3x3 operator is used, the process of closing is illustrated in Figure 4.4. (a) is the original data and using the operator to do dilation process first. Operator's center is placed on each point (either black or white) in a digital image, and if any black point in the operator has intersection with black points in (a), the center point is marked. After the dilation process, all the red and black points in (b) are marked. Black points in (b) come from the original data and red ones are new added. Next we will use (b) to apply for the erosion process. Still place the center of the operator on every point in (b). If all the black points in the operator can be overlapped by red or black points in (b), the center is marked. (c) illustrates the final result of close operation. Only the red and black points are kept and the empty points are removed during the erosion process. We can see small leak in the original data is filled during close operation. In 3D, the theory is the same and we only need to select an appropriate 3D operator that suits the data.



(a) Original Data (b) After Dilation (c) After Erosion

Figure 4.4 Close Operation

4.4.2 Thinning Operation

The thinning method is introduced in Section 3.2. Figure 4.5 is the thinning result of a simple test example and a complicated real data.

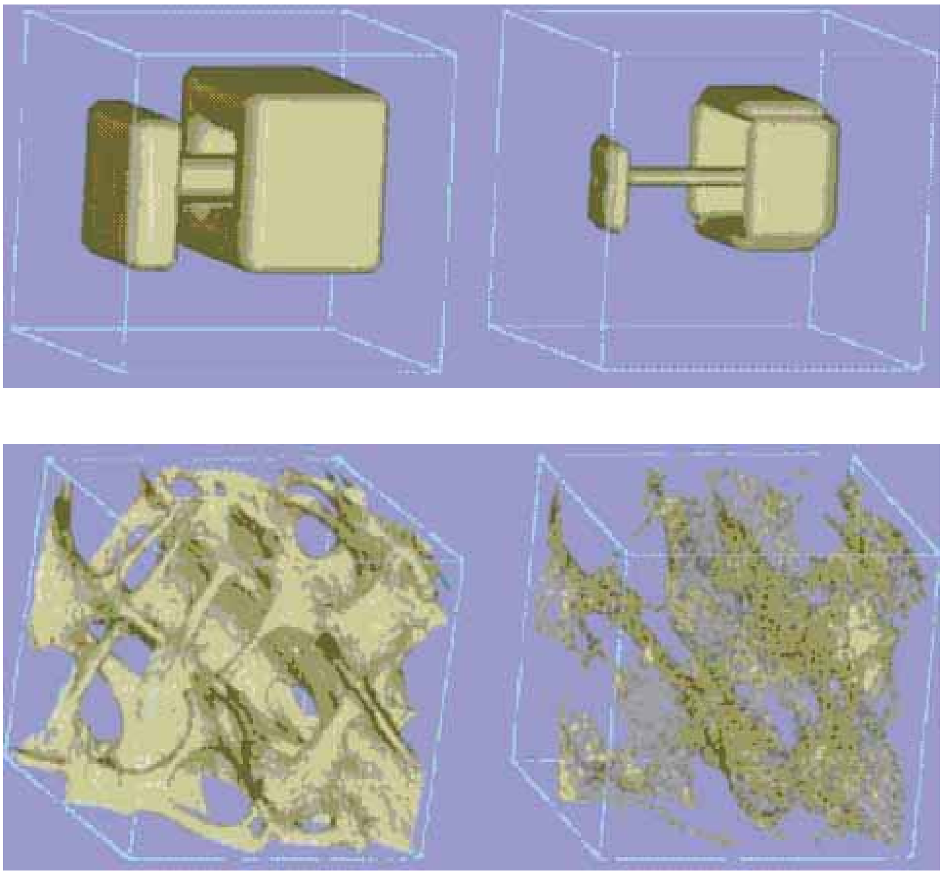


Figure 4.5 Original Data vs Thinned Data

The thinning process is just like the process of decorticating an onion which includes several iterations. During one iteration, the current out-layer is removed. When a bone voxel is deleted or kept as a skeleton point during the nth iteration, the iteration index is stored. Using this information, original dataset can be restored from the skeleton representation.

4.5 Data Analysis

In this module, a classification operation (mentioned in Section 3.3) is applied to the thinned data. Each bone voxel in the thinned volume is associated with a topological type. A visualization function is added in this module to display the classified skeleton representation (Figure 4.6).

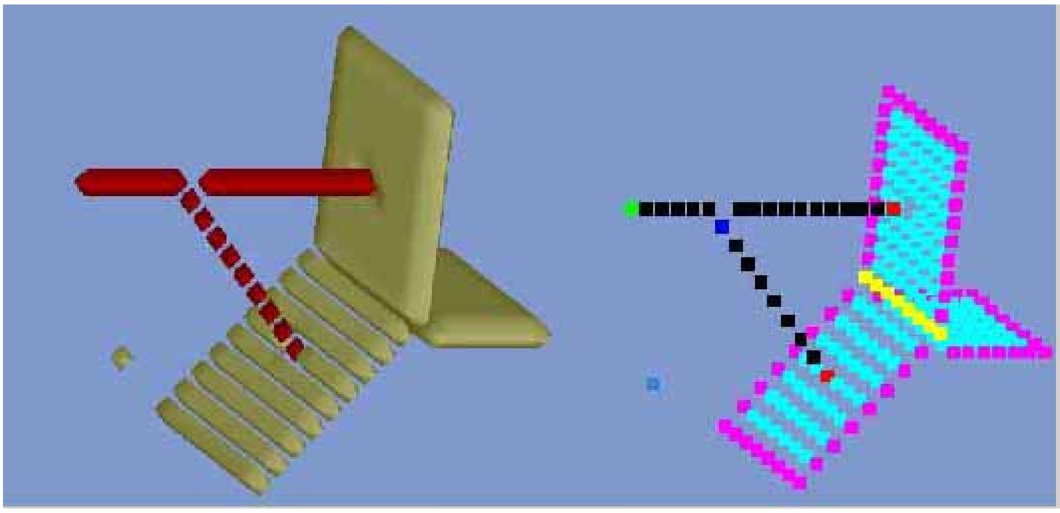


Figure 4.6 (a) Skeleton representation (b) Voxels with topological type

After classification, Wehrli et al. [36] designed two parameters to describe the quality of trabecular bone: (# means the number of voxels)

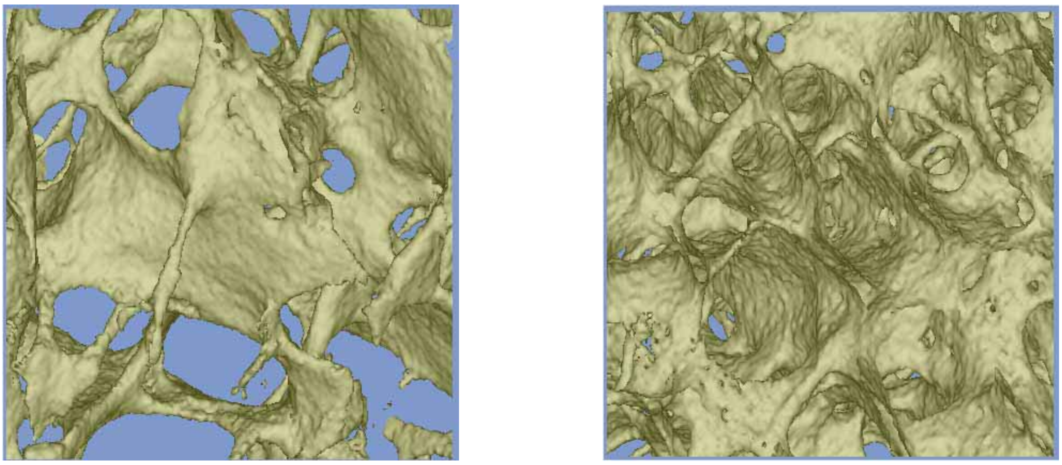
SCR (surface to curve ratio) = (#S+#SE+#SS)/(#C+#CC+#P/2)

SCR is the ratio of the number of voxels on surfaces to the number of voxels belonging to curves. In the classification phase, flat ribbonlike structures are defined as P-type and they appear in pairs. Two P-type voxels can be treated as one C-type voxel, so half of the P-type voxels are added to the denominator. To compare two trabecular bones, the one with bigger SCR has better quality and this agrees with the conclusion that comes from bone density.

EI (topological erosion index) = (#C+#CE+#SE+#P+#CC)/(#S+#SS)

EI means the ratio of parameters expected to increase divided by those expected to decrease in bone erosion process. During ageing, more and more plate-like structures convert to rod-like structure and the thickness becomes thin. The numerator of EI is the parameters that will increase during ageing and the denominator stands for the parts that will decrease. EI can be looked as a trend of osteoporosis process and a bigger EI value means a more serious osteoporosis situation.

SCR and EI parameters in [36] only consider the number of voxels in the thinned volume dataset, but ignore the original thickness of each plate-like and rod-like structure. We know that after thinning, a thick surface/curve and a thin surface/curve are both converted to a surface/curve with one voxel thick. Obviously the thickness of trabecula dominates the bone quality. Trabecula in normal bone is always thicker than bone with osteoporosis. So I modified the SCR and EI to let them associate with the thickness information. We have known that during the thinning process, several iterations are needed and the iteration index is stored in the dataset. To each bone voxel which is marked in the thinning process as a final skeleton voxel, an iteration index is associated with it. A bigger iteration value means that there are more layers surround it before the thinning process than a smaller one. We can make use of this information when we calculate SCR. Instead of calculating the sum of voxels on surface/curve, the iteration index is added when a bone voxel is encountered in the thinned volume. Then a "weighted" sum of surface/curve voxels can be obtained and used to calculate the "weighted" SCR. As to EI, it represents the trend of bone deterioration, so we do not associate it with the iteration index but the maximum iteration value minus each voxel's iteration index (Figure 4.7). Here we use H to represent this value. The bigger H means there are more layers will be removed before the current voxle will in an outer most layer than a smaller H.



SCR = 1.11
EI = 2.38
Weighted SCR = 0.52
Weighted EI = 4.27
Euler number = -4447

SCR = 1.35
EI = 2.07
Weighted SCR = 0.65
Weighted EI = 3.56
Euler number = -9291

Figure 4.7 Data Analysis

4.6 Rod-like Trabecula Visualization

4.6.1 Thinned Data Extension

In the visualization system, we use different colors to display the rod-like parts and plate-like parts. After thinning and classification operations, original data is converted to a surface representation and every voxel in the thinned data is associated with a topological type. Use the segmentation method mentioned in Section 3.4, thinned data is separated into disconnected parts and each part is a curve segment, a surface patch or an isolated voxel. By detecting the topological type of the voxels on each part, we can recognize a separated part belonging to a rod-like, a plate-like or an isolated part. This detection is done on the thinned data and makes visualization possible. We restore the thinned rod-like parts to the original data. During the thinning process, we store the information in which iteration a voxel is deleted or kept. Now we can make full use of this information to restore the original data.

For example, in Figure 4.8, black voxels represent a thinned curve after the thinning-classification-segmentation process and white voxels are deleted during the process. White voxels with index *i* are deleted during the *i*th iteration and using the following extension algorithm, we can restore the thinned data in Figure 4.8 to all the black voxels and white ones.

```
For (i = maximum iteration; i > 0; i--)
{
        Restore the neighbors of rod and voxels restored in previous "for"
        loops whose index is i
 }
```

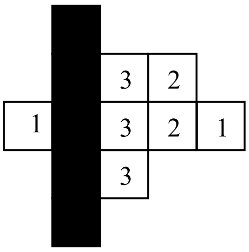


Figure 4.8 Restore to Origin

Figure 4.9 is the final result of the rod-like structure visualization which includes the extension method described above and the improvements that will be mentioned in the following sections.

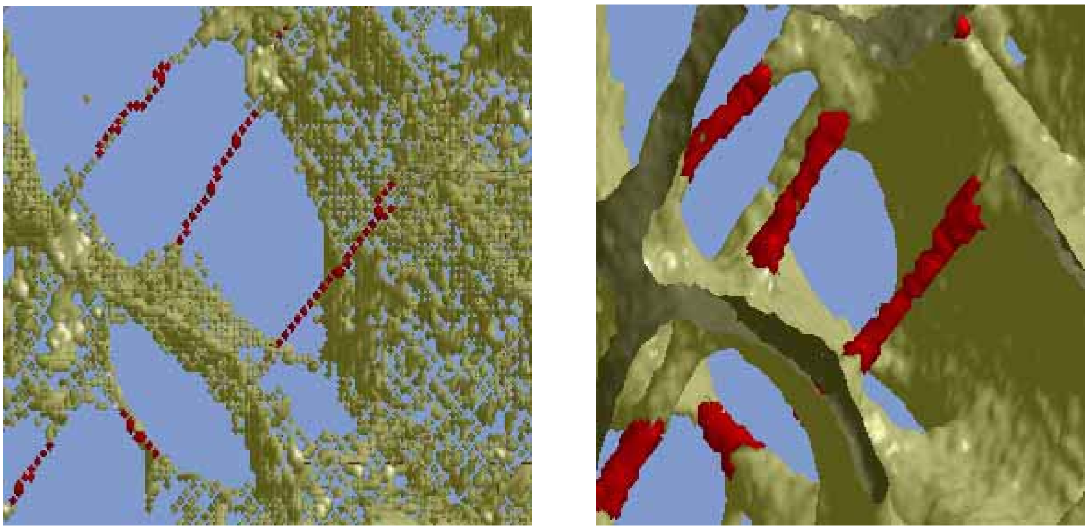


Figure 4.9 (a) Rod-like Display in Thinned Data (b) Rod-like Display in Original Data

4.6.2 Improvement of Visualization Result

Trabecular bone data is much more complicated than synthesized testing data that we can construct. Although the thinning-classification-segmentation method can get satisfactory result on synthesized data, when real trabecular data is used, a lot of possibilities arise. During the process of rod-like parts recognition, some unsatisfactory situations occurred. Some parts that look like rods are not recognized and some parts that displayed red are obviously not on rods. Three main causes can be identified that lead to the incorrectness, and corresponding improvements are given in the following three sections.

4.6.2.1 Main Direction Testing

A kind of incorrectness is that some red patches are displayed on plate-like structures and obviously we should get rid of them. Further inspection shows that some voxels in this part are classified as C-type or CC-type voxels and their distribution is not linear. Hence, main direction can be used to exclude this kind of separated structure which is classified as a curve but does not look like a curve. If a separated part has a main direction, it can be taken as a curve; otherwise, get rid of it. Covariance matrix is used to get the main direction. Firstly, let me introduce the basic theory of covariance matrix [37].

Each voxel can be written in the form of $P_i = (a_i, b_i, c_i)^T = \begin{pmatrix} a_i \\ b_i \\ c_i \end{pmatrix}, \quad i = 1,2,...,n$

Let *M* be the mean of a cluster of voxels: $M = \sum P_i / n$

and

$Q_i = P_i - M = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$ is the vector of each voxel P_i relative to *M*.

The covariance matrix of the voxels is defined by

$$S=(Q_1Q_2...Q_n)\begin{pmatrix}Q_1^T\\Q_2^T\\\vdots\\Q_n^T\end{pmatrix}=\begin{pmatrix}x_1\,x_2 & & x_n\\y_1\,y_2\cdots y_n\\z_1\,z_2 & & z_n\end{pmatrix}\begin{pmatrix}x_1 & y_1 & z_1\\x_2 & y_2 & z_2\\ & \vdots & \\x_n & y_n & z_n\end{pmatrix}=\begin{pmatrix}\sum x_i^2 & \sum x_iy_i & \sum x_iz_i\\\sum y_ix_i & \sum y_i^2 & \sum y_iz_i\\\sum z_ix_i & \sum z_iy_i & \sum z_i^2\end{pmatrix}$$

$$=\begin{pmatrix}S_{11} & S_{12} & S_{13}\\S_{21} & S_{22} & S_{23}\\S_{31} & S_{32} & S_{33}\end{pmatrix}$$

Covariance matrix S is a real symmetric matrix and has three real eigenvalues. S can be reduced to a diagonal matrix by a similarity transformation. Suppose λ_0, λ_1, λ_2 are the three eigenvalues and v_0, v_1, v_2 are the three corresponding orthogonal eigenvectors.

From $\begin{cases}Sv_0=\lambda_0v_0\\Sv_1=\lambda_1v_1\\Sv_2=\lambda_2v_2\end{cases}$, we can get

$$S(v_0v_1v_2)=(v_0v_1v_2)\begin{pmatrix}\lambda_0 & & \\ & \lambda_1 & \\ & & \lambda_2\end{pmatrix}\Rightarrow(v_0v_1v_2)^{-1}S(v_0v_1v_2)=\begin{pmatrix}\lambda_0 & & \\ & \lambda_1 & \\ & & \lambda_2\end{pmatrix}$$

The eigenvalues and their corresponding eigenvectors can represent the orientation of such a series of voxels. They define an ellipsoid that approximately reflects the main direction of these voxels. The three eigenvalues are the length of the three axes of the ellipsoid and their corresponding eigenvectors are the directions of the three axes (Figure 4.10).

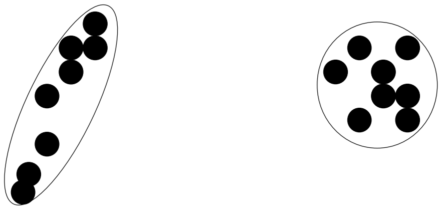


Figure 4.10 A cluster of voxels surrounded by an ellipsoid

If one eigenvalue in λ_0, λ_1, λ_2 is much larger than the other two, there must be a main direction given by the corresponding eigenvector. That is the reason why the covariance matrix can be used to determine the main direction. To calculate the eigenvalues and eigenvectors, use

$$|\lambda I - \mathrm{S}| = 0 \Rightarrow \left| \begin{pmatrix} \lambda & & \\ & \lambda & \\ & & \lambda \end{pmatrix} - \begin{pmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{pmatrix} \right| = 0$$

$$\Rightarrow \begin{vmatrix} \lambda - S_{11} & -S_{12} & -S_{13} \\ -S_{21} & \lambda - S_{22} & -S_{23} \\ -S_{31} & -S_{32} & \lambda - S_{33} \end{vmatrix} = 0$$

λ_0, λ_1, λ_2 can be got and then v_0, v_1, v_2 are easy to calculate.

By comparing the three eigenvalues, we can determine whether a cluster of voxels has a main direction. So in our program, when a curve structure is detected, main direction restriction is applied to it to see whether this structure has an obvious main direction. We discard those parts that have no main direction and only display parts that obviously look like rods. An example is given in Figure 4.11.

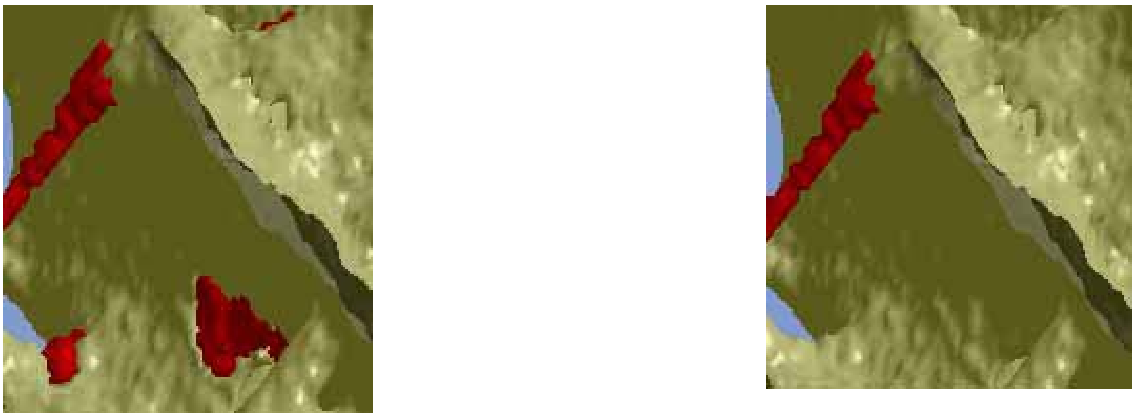


Figure 4.11 Effect of Main Direction Testing

Main direction determination is a very useful technique for the rod-like structure visualization. Statistics got from 7 datasets of trabecular bone samples are listed in table 4.1. The second row gives the number of rods without using any improvement and the third row lists the number of deleted rods using the main direction determination. The fourth row is the ratio of the result with the main direction method to that with the original method. On average 42% of the structures that are first taken as rod-like structures are reserved after applying the main direction determination.

	data1	data2	data3	data4	data5	data6	data7
No modification	100	75	85	62	86	56	41
Have main direction determination	59	37	43	44	46	30	29
Ratio of kept rods	41%	51%	49%	29%	47%	46%	29%

Table 4.1

4.6.2.2 Short Curves Combination

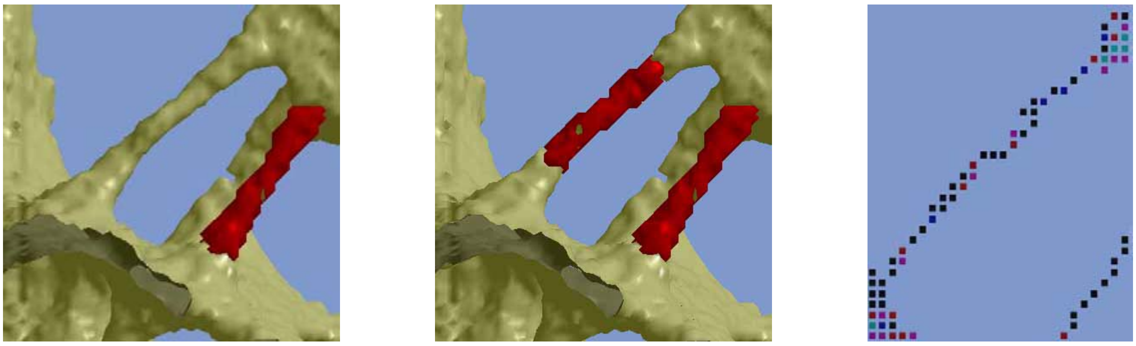
In segmentation process, a separated part which is detected as a curve should satisfy three conditions:

- Number of voxels in this part should bigger than a user defined threshold.
- C-type voxels are contained in this part.
- This part should have a main direction.

The reason why the first condition is needed is: if there is no length limitation, some very tiny parts which are classified as curve are in fact not on rods. If those tiny parts are extended to the original data, red color appears on plate-like structures and this is the thing that we do not want to see. In order to avoid this kind of problem, we only identify the curve whose length is greater than a threshold (rod-like structure in trabecular bone should not be very short). By applying the threshold, some incorrect parts are removed. But on the other hand, some structures obvious look like rods are not identified. By studying the thinned data and classification type of this kind, it is found that a rod can contain several very short curves which are separated by CC-type voxels. Because we use the length restriction to exclude very short curves, this kind of structure is missed. The following steps are used to combine all the short curves on one rod:

- Store all short curves during the segmentation process.
- Found out the short curves which are nearby and combine them into a cluster.
- Determine whether this cluster of voxels has a main direction. If yes, then this cluster can be identified as a rod-like structure.

In Figure 4.12, at first the left rod-like structure in (a) cannot be detected and voxels on it are most C-type (black points in (c)). After applying short curves combination, we can get the result in (b).



(a) (b) (c)

Figure 4.12 Short Curves Combination

Statistics with/without the short curve merging are listed in table 4.2. The second row has the same meaning as in table 4.1 and the third row shows the newly added number of rods after the merging technique is applied. The third row is the ratio of new added to the original number. On average 7% new curves are added.

	data1	data2	data3	data4	data5	data6	data7
No modification	100	75	85	62	86	56	41
Have merging short curves	6	5	6	6	10	1	3
Ratio of new added curves	6%	7%	7%	10%	12%	2%	7%

Table 4.2

4.6.2.3 Thin Surface Recognition

After thinning, some rod-like parts may be classified as thin surfaces, so they are missed when we display the curve parts. In order to add this kind of structure into our display list, for every separated part which is classified as a surface, we will determine whether it is a thin surface. If it a thin surface in our definition, we can take it as a curve. Find the center of such a cluster of voxels and then place a ball with a user defined radius at the center. The diameter is the thin surface's width that we can tolerate, i.e., if the surface is wider than the tolerance, we take it as a real surface, otherwise it is a very thin surface and we can take it as a curve (Figure 4.13).

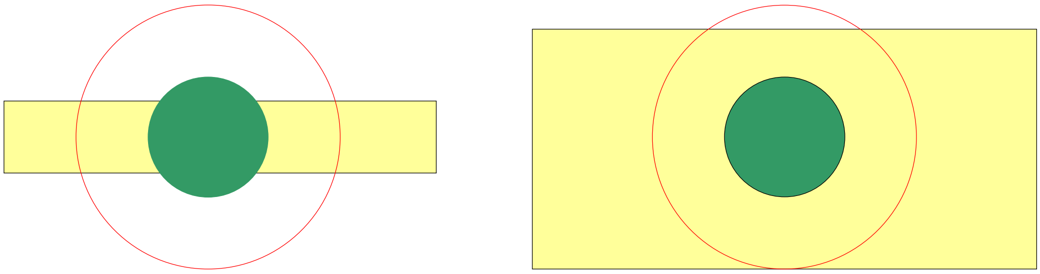


Figure 4.13 Thin Surface Detection

Suppose that S is a cluster of voxels and C is the center of S.

The detailed method is:

- Place a small ball (green ball) at C and remove voxels that contained in it. Now the remaining voxels named $S^{'}$.
- If $S^{'}$ is disconnected, we can say that S is a thin surface; otherwise, this surface is not very thin and can be viewed as a real surface.

Here in order to improve efficiency, we do not need to use all voxels in this cluster. First place a big ball (red ball) at the center and only need to care about voxels both in this ball and on the surface. The radius of the big ball is selected by users and it can be several times of the radius of the small ball. Voxels inside the big ball is used and this set of voxels is named $S^{''}$. Now the input of the above method becomes $S^{''}$ and the remaining steps are not changed.

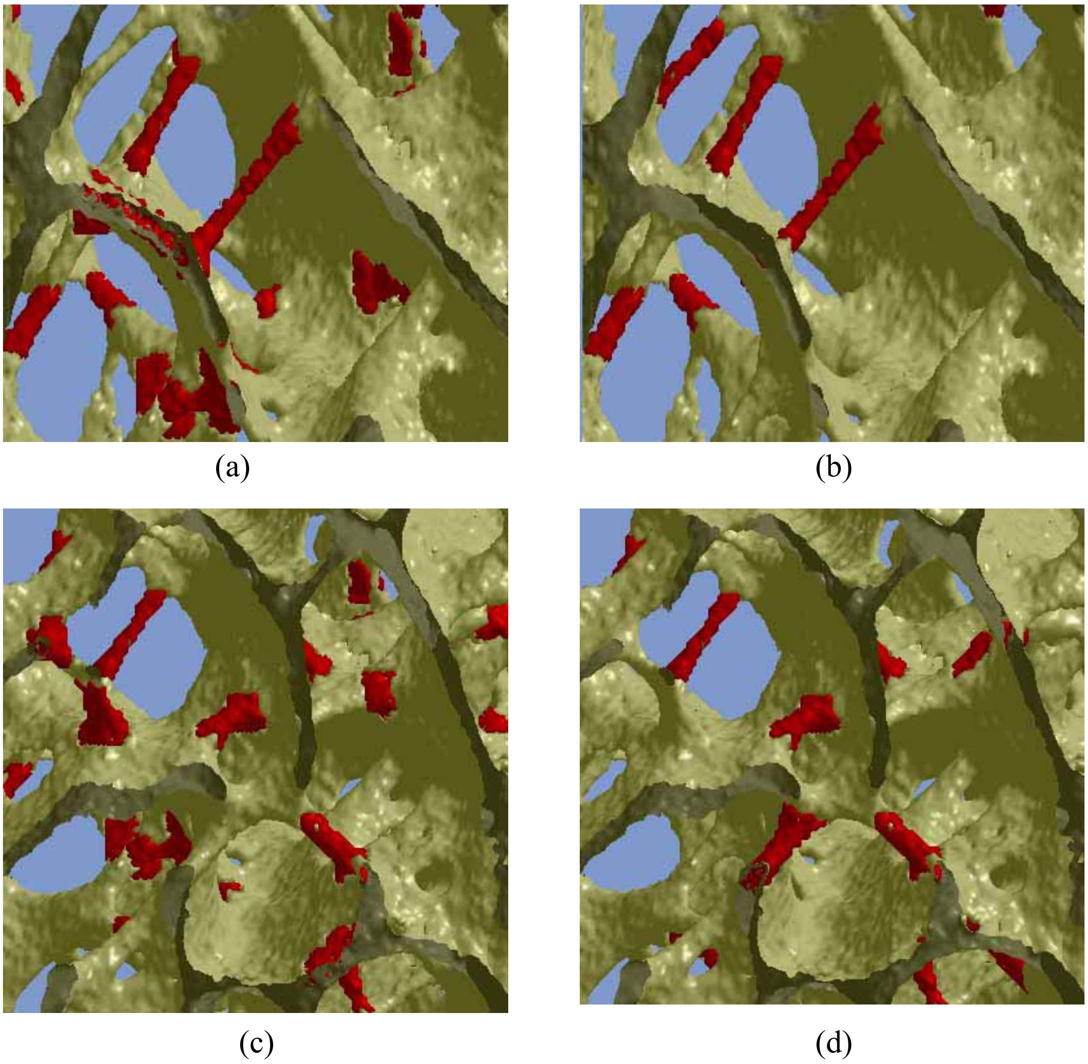
The phenomenon that a rod-like structure is determined as a surface type is not very common in real dataset, but it does exist. In table 4.3, the third row shows the number of thin surface detected. Only about 1% thin surfaces are reclassified as curves.

	data1	data2	data3	data4	data5	data6	data7
No modification	100	75	85	62	86	56	41
New added thin surfaces	0	0	1	1	1	0	1
Ratio of new added thin surfaces	0%	0%	1.2%	1.6%	1.2%	0%	2.4%

Table 4.3

4.6.2.4 Evaluation

By applying the above three improvements, the extraction and visualization of rod-like structures are obviously enhanced (Figure 4.14, (a) and (b) came from the same dataset. (b) is applied the improvements and (a) without. (c) and (d) came from the same dataset. (d) is applied the improvements and (c) without).



(a) (b)

(c) (d)

Figure 4.14 Poor vs Good Visualization

We evaluate our method on seven datasets and list the results in table 4.4. The second row gives out the number of rods without using our method and the third row shows the number of rods using our method. The number of rod-like structure changed a lot after the improvements are used.

	data1	data2	data3	data4	data5	data6	data7
No modification	100	75	85	62	86	56	41
Improved results	47	43	48	24	50	27	15

Table 4.4

There is still no accurate evaluation method to validate my method, only from careful observation and intuitive determination of our eyes.

Chapter 5

Remaining Problems and Future Work

5.1 Remaining Problems

After applying the method mentioned in chapter 4, there are still two kinds of problems that cannot be settled well:

- A structure is obviously a rod but cannot be recognized and voxels on it are C-type or CC-type.

  In last chapter, we mentioned that a rod-like structure can be a combination of a lot of short curves which point to about the same direction. In this situation, we can find out these short curves, connect them together and form the long curve. But when there are too many CC-type voxels on the rod and their distances are very near, errors will occur.

  During segmentation process, in order to separate each part, we disconnect the thinned data at the location of joint voxels. If we only remove the joint voxels, the remaining voxels still can connect together, so we remove both the joint voxels and their 3x3x3 neighborhood. If too many joint voxels along the rod-like part, when we remove voxels, all the neighbor C-type voxels may be removed also (they are contained in the 3x3x3 neighborhood). When we try to restore the rod-like part from the remaining voxels to the original part, nothing can be restored from an empty dataset.

Although this kind of problem is not very common, it does exist indeed. Method should be found out to detect this situation.

- Another problem is described in Figure 5.1: Applying Saha's [27] thinning algorithm to Figure 5.1(a), before the iteration, all voxels are s-open points. So in the first scan (for s-open points), all voxels in Figure 5.1(a) are used. There are no e-open or v-open voxels in Figure 5.1(a), so no point is kept or deleted in the next two scans. After thinning, the result is Figure 5.1(b). The topology is preserved but obviously a drilling is created.

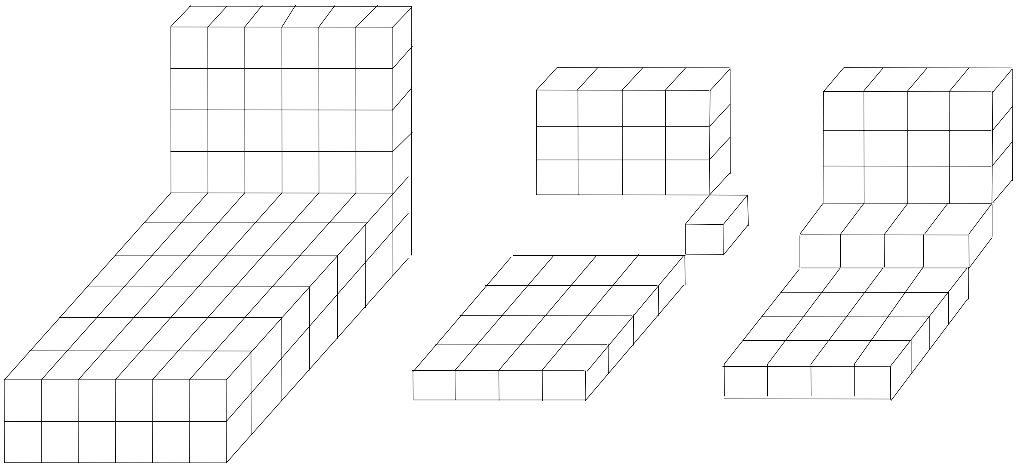


Figure 5.1 (a) (b) (C)

In Saha's [27] thinning algorithm, he mentioned that after the first scan (only care about s-open points), Figure 5.2(a) becomes Figure 5.2(b), and after the second scan (e-open points are used), the result becomes Figure 5.2(c). Although topology is maintained, there appears a drilling. This shape seems a little strange, so Saha designed a condition to check 2D topology preservation in each co-ordinate plane which passes through the candidate e-open point and contains at least one white e-open point before the iteration. After applying this condition, drilling will not be created in Figure 5.2 (c).

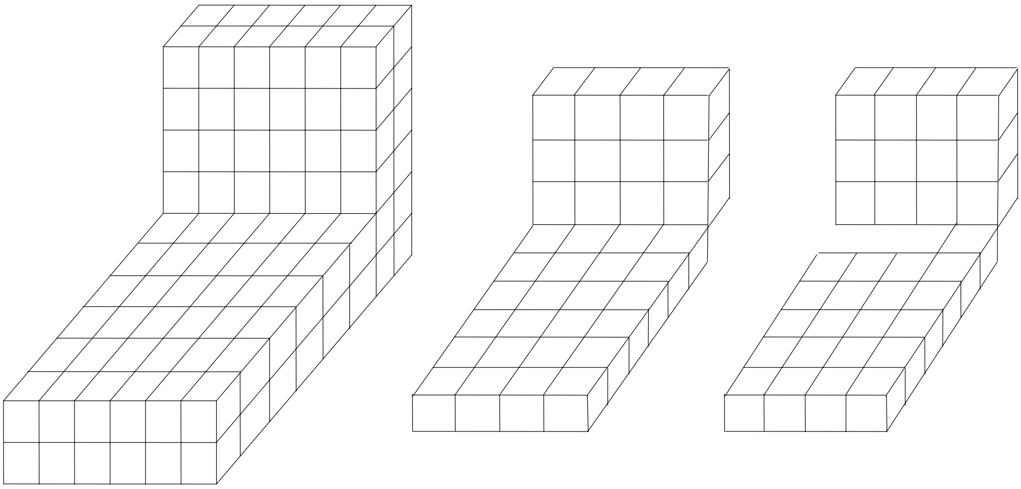


Figure 5.2 (a) (b) (c)

Prompt by the above condition, I also want to enforce some restriction that can lead the thinning result to Figure 5.1(c) instead of (b). But I still cannot find the efficient method that can only deal with this problem while do not effect the other parts.

5.2 Future Work

In the next step, I will try to improve the efficiency of the system. In some functions, I use brute-force method and in the future, try to design more efficient method. Parallel manipulation can also be applied to the thinning algorithm if parallel environment is available. A parallel thinning algorithm is mentioned in [27].

Bibliography

[1] http://www.osteoporosis.ca

[2] R. T. deHoff, E. H. Aigeltinger and K. R. Craig, Experimental determination of the topological properties of three-dimensional microstructures, J Microsc 95, 1972, pp. 69-91.

[3] J. D. Foley, A. V. Dam, S. K. Feiner and J. F. Hughes, Computer Graphics: principles and practice: *Addison-Sesley*, pp. 1034,72,649.

[4] W. E. Lorensen and H. E. Cline, Marching Cubes: A High Resolution 3D Surface Construction Algorithm, Computer Graphics, 21(4): 163-169, July 1987.

[5] M. Levoy, Display of surfaces from volume data. IEEE Computer Graphics and Applications, 8(3): 29-37, 1988.

[6] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer and L. Seiler, The VolumePro Real-Time Ray-Casting System, ACM Proceedings of SIGGRAPH, pp. 251-260, August 1999.

[7] A. Kaufman, Volume Visualization, IEEE Computer Society Press Tutorial, Los Alamitos, CA, 1990.

[8] A. Kaufman, D. Cohen and R. Yagel, Volume Graphics, IEEE Computer, 26(7): 51-64, July 1993.

[9] D. Bartz and M. Meiβner, Volume Graphics: *Springer Book*, pp. 171-183.

[10] J. Wilhelms, A. V. Gelder, A coherent projection approach for direct volume rendering, Computer Graphics, 25(4): 275-284, August 1991.

[11] L. Westover, Footprint evaluation for volume rendering, Computer Graphics, 24(4): 367-376, August 1990.

[12] Cabral B, Cam N and Foran J, Accelerated volume rendering and tomographic reconstruction using texture mapping hardware, Proceedings 1994, ACM/IEEE Symposium on Volume Visualization, pp.91-98.

[13] M. J. Dürst, “Letters: Additional reference to “Marching Cubes””, Computer Graphics, 22(2): 72-73, July 1988.

[14] G. M. Nielson and B. Hamann, The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes, IEEE visualization '91, pp. 83-91, October 1991.

[15] C. Montani, R. Scateni and R. Scopigno, Discretized Marching Cubes, IEEE Visualization '94, pp. 281-287, October 1994.

[16] R. Shekhar, E. Fayyad, R. Yagel and J. F. Cornhill, Octree-Based Decimation of Marching Cubes Surfaces, IEEE Visualization '96, pp. 335-342, October 1996.

[17] A. Watt and M. Watt, Advanced Animation and Rendering Techniques: *Addison-Sesley*, pp. 305-313.

[18] Volume Visualization With Ray Casting, John Pawasauskas, http://www.cs.wpi.edu/~matt/courses/cs563/talks/powwie/p1/ray-cast.htm

[19] J.Wolff, Das Gesetz der Transformation der Knochen. Berlin: A. Hirschwald, 1892.

[20] T. Y. Kong and A. Rosenfeld, Digital topology: Introduction and survey, Computer Vision, Graphics and Image Process 48, pp. 357-393, 1989.

[21] P. K. Saha, B. R. Gomberg and F. W. Wehrli, Three-Dimensional digital topological characterization of cancellous bone architecture, International Journal of Imaging Systems and Technology 11, pp. 81-90, 2000.

[22] B. R. Gomberg , P. K. Saha, H. K. Song, S. N. Hwang and F. W. Wehrli, Topological Analysis of Trabecular Bone MR Images, IEEE Transaction of Medical Imaging, 9(3), March 2000.

[23] H. Blum, A transformation for extracting new descriptors of shape, Models for the Perception of Speech and Visual Form, pp. 362-380, 1967.

[24] Y. F. Tsao and K. S. Fu, A general scheme for constructing skeleton models, Inform. Sci. 27, pp. 53-87, 1982.

[25] C. J. Hilditch, Linear skeletons from square cupboards, Machine Intelligence, Vol. 4, 1969.

[26] R. Stefaneli and A. Rosenfeld, Some parallel thinning algorithms for digital pictures, J. Assoc. Comput. Mach., 18: 255-264, 1971.

[27] P. K. Saha, B. B. Chaudhuri and D. D. Majumder, A new shape preserving parallel thinning algorithm for 3D digital images, Pattern Recognition, 30(12): 1939-1955, 1997.

[28] S. Lobregt, P. W. Verbeek, and F. C. A. Groen, Three-dimensional skeleton: Principle and algorithm, IEEE Transaction on Pattern Analysis and Machine Intelligence, PAMI-2, pp. 75-77, 1980.

[29] Y. Sato, C. F. Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Tamura and R. Kikinis, Tissue classification based on 3D local intensity structures for volume rendering, IEEE Transactions on Visualization and Computer Graphics, 6(2), 2000.

[30] U. Eckhardt and L. Latechi, Digital topology, In Current Topics in Pattern Recognition Research, Research Trends, Council of Scientific Information, Vilayil Gardens, Trivandrum, India, 1994.

[31] P. K. Saha and B. B. Chaudhuri, 3D digital topology under binary transformation with applications, Computer Vision and Image Understanding, 63(3): 418-429, 1996.

[32] P. K. Saha and B. B. Chaudhuri, A new approach to computing the Euler characteristic, Pattern Recognition, 28(12): 1955-1963, 1995.

[33] S. Bischoff and L. P. Kobbelt, Isosurface reconstruction with topology control, 10th Pacific Conference on Computer Graphics and Applications, pp. 246-255, October 2002.

[34] A. M. Parfitt, Stereologic basis of bone histomorphometry: Theory of quantitative microscopy and reconstruction of the third dimension, Bone Histomorphometry: Techniques and Interpretation. RR Recker, editor. CRC Press, Boca Raton, FL. 53-89.

[35] A. K. Jain, Fundamentals of digital image processing, *Prentice Hall*, pp.342-430.

[36] F. W. Wehrli, B. R. Gomberg, P. K. Saha, H. K. Song, S. N. Hwang, and P. J. Snyder, Digital topological analysis of in vivo magnetic resonance microimages of trabecular bone reveals structural implications of osteoporosis, J. of Bone and Mineral Research, 16(8): 1520-1531, 2001.

[37] A. S. Milman, Mathematical principles of remote sensing: making inferences from noisy data, *Sleeping Bear Press*, 1999, pp. 99-142.