



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

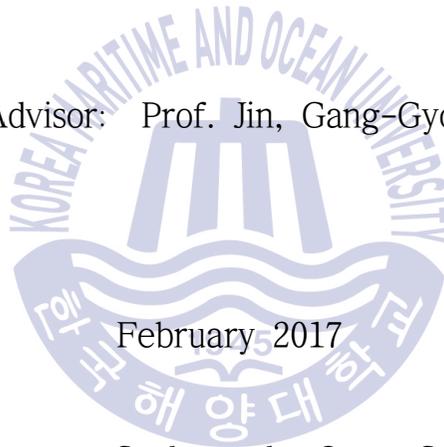
이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Thesis for a Master's Degree

Enhanced BC Algorithm Incorporating a Novel Sampling Step and a Fractional Box Count

Advisor: Prof. Jin, Gang-Gyoo



February, 2017

Department of Convergence Study on the Ocean Science and Technology
Ocean Science and Technology School
Korea Maritime and Ocean University

So, Hye-Rim

Approved by the Committee of the Ocean Science and Technology
School of Korea Maritime and Ocean University in Fulfillment of
the Requirements for the Degree of Master of Science in
Engineering

Dissertation Committee :

Prof. Cho, Seok-Je, Chair _____

Prof. Oh, Sea-Jun _____

Prof. Jin, Gang-Gyoo, Advisor _____

February 2017

Department of Convergence Study on the Ocean Science and Technology
Ocean Science and Technology School
Korea Maritime and Ocean University

Contents

List of Tables	iii
List of Figures	iv
Abstract	vi

Chapter 1. Introduction

1.1 Motivation	1
1.2 Research objectives	3
1.3 Organization of the thesis	3

Chapter 2. Overview of Fractal Theory

2.1 Definition of fractal	5
2.2 Fractal dimension	7
2.3 Fractal geometry	9
2.3.1 Mandelbrot set and Julia set	10
2.3.2 Koch snowflake (Opened)	11
2.3.3 Apollonian gasket	12
2.3.4 Vicsek fractal	13
2.3.5 Sierpinski triangle	14
2.3.6 Rand cantor	15
2.3.7 Koch curve 85°	16
2.3.8 Sierpinski carpet	17
2.3.9 Hilbert curve	18

Chapter 3. Existing Box-Counting Methods

3.1 Conventional BC method	20
3.2 Triangle BC method	25

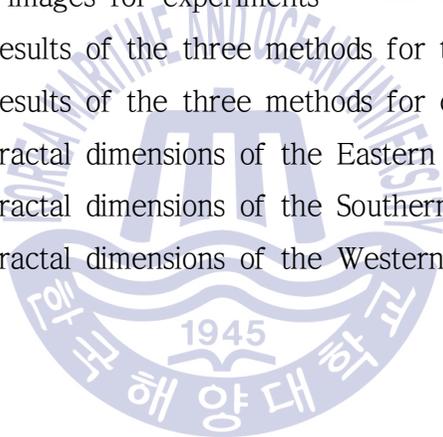
Chapter 4. Enhanced BC method

4.1 Existing sampling methods and their drawbacks	27
---	----

4.1.1 Sampling methods	27
4.1.2 Pixel utilization	30
4.1.3 Drawbacks of existing sampling methods	30
4.2 New sampling method	32
4.3 Fractional box count	35
4.4 Procedure of the enhanced BC method	38
Chapter 5. Experiments and Review	
5.1 Experiments on deterministic fractal image	41
5.1.1 $M \times M$ test image	41
5.1.2 Determination of η	43
5.1.3 Experiment with images of $M \times M$ pixels	44
5.1.4 Experiments on rotated $M \times M$ image	45
5.1.5 Experiment with images of $M \times N$ pixels	46
5.2 Experiments on non-deterministic fractal images	51
5.2.1 Converting color images to binary images	51
5.2.2 Coastline images	52
Chapter 6. Conclusion	56
References	58
Appendix	61

List of Tables

Table 2.1 Schematics of dimension	8
Table 3.1 Step size and box numbers	21
Table 3.2 Overall procedure of the BC method	24
Table 3.3 Overall procedure of the TBC method	26
Table 4.1 Procedure of the MGS method	33
Table 4.2 Sampling results of the GS and MGS methods on 191×256 image	34
Table 4.3 Step size and real number box-counting	37
Table 4.4 Overall procedure of the enhanced BC method	39
Table 5.1 $M \times M$ test images for experiments	42
Table 5.2 Estimated results of the three methods for the test images ..	44
Table 5.3 Estimated results of the three methods for curves rotated	45
Table 5.4 Estimated fractal dimensions of the Eastern coastlines	53
Table 5.5 Estimated fractal dimensions of the Southern coastlines	54
Table 5.6 Estimated fractal dimensions of the Western coastlines	54



List of Figures

Figure 2.1 Measuring the length of the KMOU's campus coastline on different scale (Google photo)	6
Figure 2.2 Images of fractal geometry (Naver photo)	9
Figure 2.3 Mandelbrot set and Julia set	10
Figure 2.4 Koch snowflake (Opened)	11
Figure 2.5 Apollonian gasket	12
Figure 2.6 Vicsek fractal	13
Figure 2.7 Sierpinski triangle	14
Figure 2.8 Cantor set	15
Figure 2.9 Koch curve 85°	17
Figure 2.10 Sierpinski carpet	18
Figure 2.11 Hilbert curve	19
Figure 3.1 Box counting on an image of 16×16 pixels	22
Figure 3.2 Data fitting using the least squares method	24
Figure 3.3 Patterns of partitioning a square box	25
Figure 3.4 Box-counting of the TBC method	25
Figure 4.1 GS sampling on an image of 8×8 pixels	28
Figure 4.2 AS sampling on an image of 8×8 pixels	29
Figure 4.3 Number of step size to changes of M	31
Figure 4.4 Coverage ratio of the GS and MGS sampling methods	34
Figure 4.5 Partition example of an image	36
Figure 4.6 Sampling of the image of 17×21	38
Figure 5.1 η versus MAE	43
Figure 5.2 $M \times N$ sub-image from $M \times M$ image	46
Figure 5.3 MAE of the three methods for $128 \times N$ images	47
Figure 5.4 Average pixel waste ratio of the GS sampling method	48
Figure 5.5 MAE of the three methods for $191 \times N$ images	49

Figure 5.6 *MAE* of the three methods for $256 \times N$ images 50
Figure 5.7 Binarization of coastline images 51
Figure 5.8 Korean coastline maps for experiment 53



ABSTRACT

The Box-Counting (BC) method is one of the most commonly used algorithms for fractal dimension calculation of binary images in the fields of Engineering, Science, Medical Science, Geology and so on due to its simplicity and reliability.

One of the issues related to fractal dimension is data sampling that involves a process where a certain size of box is taken from a given image and it has a direct effect on the precision of the fractal dimension estimation. The Geometric Step (GS) method, arithmetic step method, and divisor step method are the representative methods. The GS method is mainly used because of its efficiency. However, the GS method has some drawbacks in nature. If the image size is large, it provides insufficient data for regression analysis. It can be applied to the image of $M \times M$ pixel size for 100 [%] pixel utilization. Application of the GS method to an image of $M \times N$ may waste pixels in the calculation and degrade the estimation accuracy.

In this thesis, a novel sampling method is proposed in order to resolve the shortcomings of the GS method on the basis of the intuitive observation that an estimate may have a higher degree of precision if more pixels are utilized in each step and a sufficiently large number of fitting data are guaranteed. The proposed sampling method is an improved version of the conventional GS method, called *the modified GS (MGS) method*. The MGS method selects some additional step sizes with higher pixel utilization rate among the middle values between the integer powers of 2 to constitute the overall step set with the GS method.

Not all sampling methods including the MGS method can guarantee 100 [%] pixel utilization when the BC method is applied to images of an arbitrary size.

This study suggests a novel fractional counting method to resolve the problem of pixel waste. The proposed counting method counts pixels of fractal within a discarded box (not of $\delta \times \delta$ size) and adds its fractional count normalized by both the average pixel number of all boxes with $\delta \times \delta$ size and step size δ to integer count.

The performance of the enhanced BC method incorporating the MGS method and fractional counting method is verified on a set of deterministic fractal images whose theoretical dimensions are well known and compared it with those of the existing BC methods. The experimental results show that the proposed method outperforms the conventional BC method and triangle BC method.



Chapter 1. Introduction

1.1 Motivation

The fractal theory is an effective tool for modeling of complicated geometrical structures found in the nature, characterized by self-affinity and recursiveness [1]. When a part of a fractal is magnified and then rotated, the shape becomes equal to the original fractal entirely or partly, or has a statistically similar structure, which is called self-affinity. The never-ending repetition of a simple structure is called recursiveness. Fractal structures are classified as non-deterministic structures such as the coastline, rivers, streams, and thunders that may be seen from the surroundings and deterministic structures created by mathematical rules such as Mandelbrot set, Sierpinski triangle, and Sierpinski carpet.

Pentland [2] showed that human recognition of the degree of fractal surface texture between smoothness and coarseness, which is geometric irregularity, is closely related to the fractal dimension. Various methods of estimating the fractal dimension such as the Box-Counting (BC) method, triangular prism method, and fractional Brownian motion method have been suggested [3]. The BC method is one of the most commonly used methods to be applicable to estimate the complexity of binary fractals in the field of Engineering, Science, Medicine, and Geology because

of its simplicity and high reliability [4].

Jin et al. [5] applied the BC method to the fractal dimension estimation of the images of synthetic and natural texture. Yu et al. [6] applied the BC method to the recognition of human iris image divided into the higher part and the lower part. Shyu et al. [7] applied the BC method to measure the complexity of the cerebral cortex surface of a fetus. Lin et al. [8] applied the BC method to estimate the fractal dimension of the river water flow rate in a basin. Barakou et al. [9] applied the BC method to calculate the fractal dimension of a distributed network and to create a virtual distributed network by using an extended fractal model.

To increase the precision of the BC method, Foroutan-pour et al. [10] discussed the appropriate size of a rectangular frame surrounding a fractal figure and its threshold. Bisoi and Mishra [11] also investigated the box size threshold. Miloevic and Elston [12] suggested a method of eliminating regression analysis data having non-linear properties by adequately controlling the box size and applied the method to estimate the fractal dimension of neurons. Kaewaramsri and Woraratpanya [13] suggested the Triangle Box-Counting (TBC) method where the Geometric Step (GS) method is used by dividing the conventional square box into two triangles to increase the precision of box-counting.

Most of the fractal dimension estimation methods suggested until now has employed the GS method where the step size δ , which is used to divide an image is a power of 2. Therefore, the size is limited to $M \times M = 2^m \times 2^m$ (m is a positive integer) in the BC method so that the pixels of the handled image are not wasted. However, the shapes and sizes of the image found in actual environment, such as coastline, rivers, chains of mountains, and plants are all different. When the BC method is applied to such images, degrade of the performance is unavoidable because some pixels may not be used for fractal dimension calculation.

1.2 Research objectives

Therefore, in this thesis an enhanced BC method which employs both a modified GS (MGS) sampling method and a novel fractional counting method is presented. The MGS method is an improved version of the GS method, which provides more numbers of step size than the GS method for data fitting when the image size is large. The novel fractional counting method provides the ability to count blocks as a real number, that is, the product of the ratio of the fractal pixel numbers within the relevant block for the averaged pixel numbers within the blocks of $\delta \times \delta$ size and the ratio of the relevant block size for the block size of $\delta \times \delta$.

The performance of the suggested BC method has been verified with two deterministic fractal images with the Mean Absolute Error (MAE) as the evaluation index. The effectiveness of the suggested BC method has been verified in comparison with the conventional BC method and the TBC method proposed by Kaewaramsri et al [13]. In addition, the suggested BC method has been applied to the coastline images of the Korean Peninsula, which are non-deterministic images, to estimate the fractal dimensions and measure the complexity.

1.3 Organization of the thesis

The thesis is organized as follows:

Chapter 2 gives an overview of the fractal theory. It presents definition of fractal and fractal dimension. Two kinds of fractal geometry and their examples are shown. It also gives a brief description of how to make deterministic fractal geometries.

Chapter 3 describes the existing methods to estimate fractal dimension of

binary (black and white) images. As well known fractal dimension estimators, the Box Counting (BC) method and the TBC method are explained. As sampling methods, the Geometric Step (GS) method, Arithmetic Step (AS) method, and Divisor Step (DS) method are introduced.

Chapter 4 is the core part in this thesis and presents an enhanced BC method. To overcome the drawbacks of the existing sampling methods, a novel method, called the modified GS (MGS) method is newly introduced and a fractional box counting method to be applicable to images of an arbitrary size is proposed.

Chapter 5 carries out experiments to select the optimal value of the user-defined parameter η of the MGS sampling method, verify the performance of the proposed BC method on a set of deterministic fractal images whose theoretical dimensions are well known and compare it with those of the existing two methods. In addition, the proposed method is used to measure the complexity of coastlines of the Korean Peninsula.

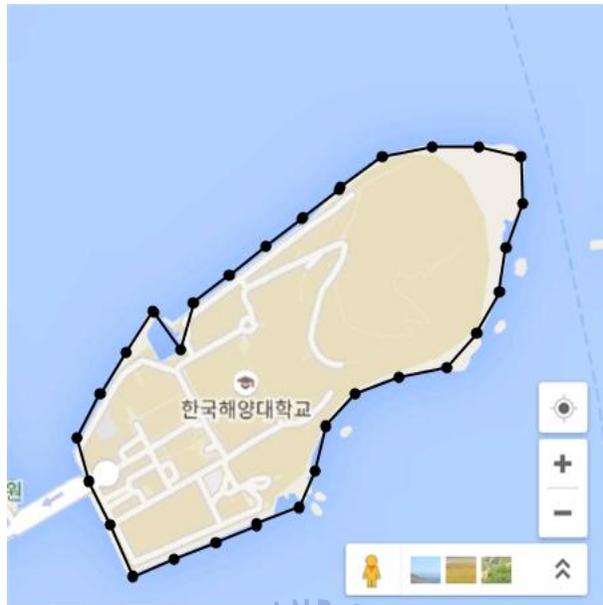
Chapter 6 summarizes the conclusion of the study.

Chapter 2. Overview of Fractal Theory

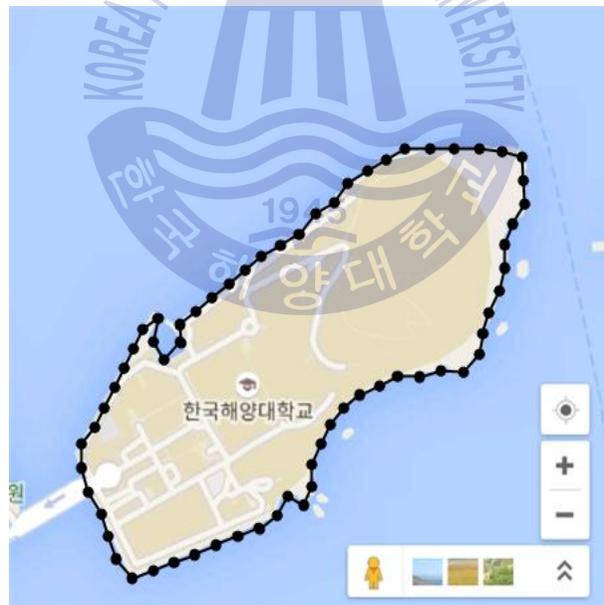
2.1. Definition of fractal

The term ‘fractal’, originated from ‘Fractus’ which is an adjective of the Latin verb ‘Frangere’ that means break, split, and fracture, has become a very important part in modern physics and mathematics since Mandelbrot [1] introduced the fractal theory by publishing a paper titled ‘How long is the coast of Britain?’ in 1975. Mandelbrot showed in his paper that the length of the coast varies according to the size of measurement unit.

For example, measuring of the length of the KMOU’s campus coastline depicted in Figure 2.1 shows that when the length is measured with a little scale ruler of 50 [m], the length is 1.50 [km], and when it is measured with a little fine scale ruler of 25 [m], the length is 1.63 [km]. In other words, the length of the coastline becomes longer when a smaller measurement unit is applied. This shows that when the measurement unit becomes infinitely smaller for measuring the length of a complex natural object other than the coast, its length may become infinitely longer. This is a new geometry that can explain the structural irregularity of nature. When we take a look at complex structures available in the nature by partially enlarging them into a geometrical shape that enables modeling, such shapes are exactly the same or statistically similar.



(a) 50 [m]



(b) 25 [m]

Figure 2.1 Measuring the length of the KMOU's campus coastline on different scale (Google photo)

The fractal geometry has a characteristic of self-similarity and prime number dimension, and this is a field of study that explains an uneven shape which is difficult to be explained in a non-linear status just as Euclidean geometry, not a smooth linear or curvilinear shape that is easy to be explained. Bracken can be considered as an example of such characteristics [14]. Bracken has a characteristic of self-similarity and recursiveness. Self-similarity means that even if a shape is scaled down in a fractal space, a part which has the spatial characteristics of the whole shape in the original scale always exists and such part is aligned in phases, and recursiveness means that two or more same or statistically similar elements are aligned, forming a successional pattern [15].

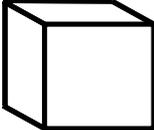
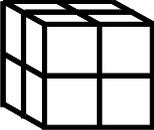
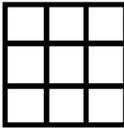
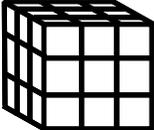
2.2 Fractal dimension

A dimension is an index showing how the general measurements are correlated with measurements such as area, length, and volume. Typically, there are integer dimensions including 0 dimension for a dot, 1 dimension for a straight line, and 2 dimensions for a plane, the dimension is a strictly absolute definition [16]. As an example of importance of dimension, when the volume of a book is measured on the assumption that it is measured on an appropriate scale and the unit of volume is constant, the same result will always be shown. However, the fractal figure on a fractional dimension cannot be measured with a scale of integer dimension. This is in the same context with the calculation of volume of line segment. The line segment is a concept in 1 dimension but the volume is a concept in 3 dimensions. Likewise, an appropriate dimension should be known in order to calculate a length and volume properly. Therefore, it is very important to know the fractal dimension which has a decimal dimension.

In mathematical approach, one side is divided by m in 1 dimension, $m(=N)$ side(s) which length is $1/m$ time(s) the original length is(are) created, and when each side of an equilateral square is divided by m in 2 dimensions,

$m^2(=N)$ square(s) which side is reduced by $1/m$ is(are) created, and each side of an equilateral hexahedron is divided by m in 3 dimensions, $m^3(=N)$ equilateral hexahedron(s) reduced by $1/m$ time(s) is(are) created. Taking a square as an example, when one side is divided by 2 in 1 dimension, two sides of which length is $1/2$ time(s) the original length are created, and when each side of an equilateral square is divided by 2 in 2 dimensions, 4 squares of which side is reduced by $1/2$ time(s) are created. This is summarized in Table 2.1.

Table 2.1 Schematics of dimension

$m \backslash D$	1	2	3
1	 $N=1$	 $N=1$	 $N=1$
2	 $N=2$	 $N=4$	 $N=8$
3	 $N=3$	 $N=9$	 $N=27$

The relational expression between the rate of reduction $r=1/m$, number of figures N and the dimension D from Table 2.1 is shown in Equation (2.1) and Equation (2.2).

$$N = 1/r^D = (1/r)^D \quad (2.1)$$

$$D = \log(N) / \log(1/r) \quad (2.2)$$

2.3 Fractal geometry

Fractal objects are divided into deterministic and non-deterministic fractal object [17]. A deterministic fractal object is created by reducing or rotating its shape recursively in phases according to the mathematical rules, and a non-deterministic fractal has a statistically similar shape with the whole shape by enlarging a part of a certain shape in phases. The examples of deterministic and non-deterministic fractal figures are shown in Figure 2.2.

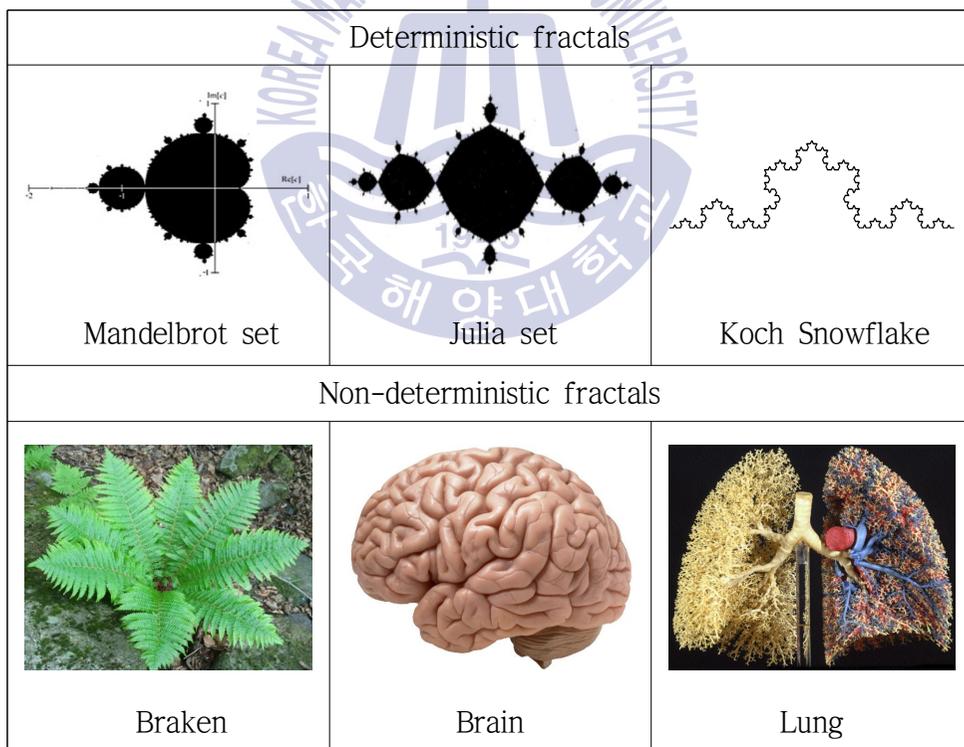


Figure 2.2 Images of fractal geometry (Naver photo)

2.3.1 Mandelbrot set and Julia set

Since there is one generator for basic fractal geometries, so the whole fractal dimension and partial fractal dimension are same [1]. However, fractal geometries appeared after Mandelbrot set or Julia set have a different value between global dimension and local dimension. The figures of Mandelbrot set and Julia set are shown in Figure 2.3.

Mandelbrot set is the set of c that does not diverge when the initial value of z is 0. It begins from Equation (2.3) which is a simple complex recurrence formula [16].

$$z_{n+1} = z_n^2 + c \quad (2.3)$$

The above equation can be explained as repetition of process to square a certain complex and add the original complex to the value. Julia set is a type of fractal designed by a French mathematician Gaston Maurice Julia and it begins from Equation (2.4) which is a simple complex recurrence formula. This is sets of z that does not diverge when c is fixed in the complex equation. In other words, the set of z varies depending on which value c is fixed to [16].

$$f_c(z) = z^2 + c \quad (2.4)$$

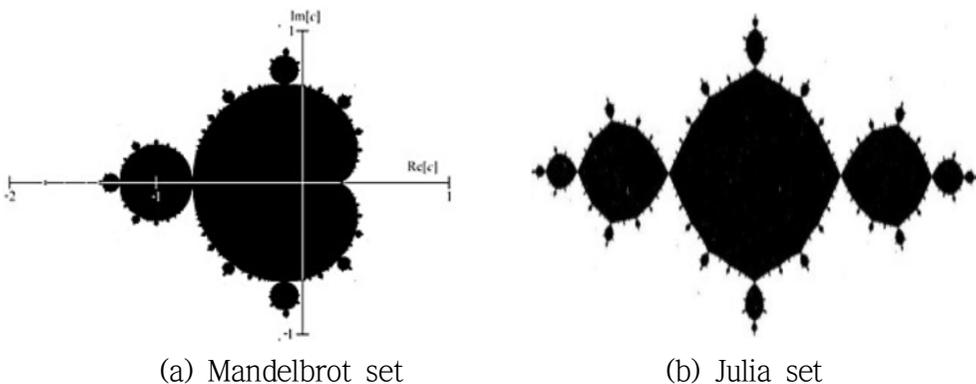


Figure 2.3 Mandelbrot set and Julia set

2.3.2 Koch snowflake (Opened)

Koch snowflake is one of fractals that were introduced first among the deterministic fractals that can be explained mathematically. It appeared first in a paper of a Swedish mathematician Helge Von Koch in 1904 [16]. The figures of Koch snowflake according to the level are shown in Figure 2.4. In order to draw Koch snowflake, a triangular shape is created by dividing one segment into three equal segments and pulling up the middle segment, and at this time, the length of two segments created from one middle segment should be same with the length of one segment at the time of dividing the original segment into three segments, and at this time, four segments of which length is the original $1/3$ are created [16]. $1:4 = 1^x : 3^x$ when substituting it for the proportional expression, and at this time, x value is the dimension. x value is $\log 4 / \log 3$ which takes the value of 1.26. We call one dimension as a line and two dimensions as a plane. 1.26 dimension is a figure in the mean dimension which is not one dimension nor two dimensions. When the above process is repeated, the length of each segment decreases by $1/3$ and the length of the whole segment increases by $1/3$. In other words, the length of one segment of n th Koch snowflake is $(1/3)^{n-1}$ and the whole length becomes $(4/3)^{n-1}$. When it is carried out infinitely, $4/3$ is larger than 1 so it becomes infinite.

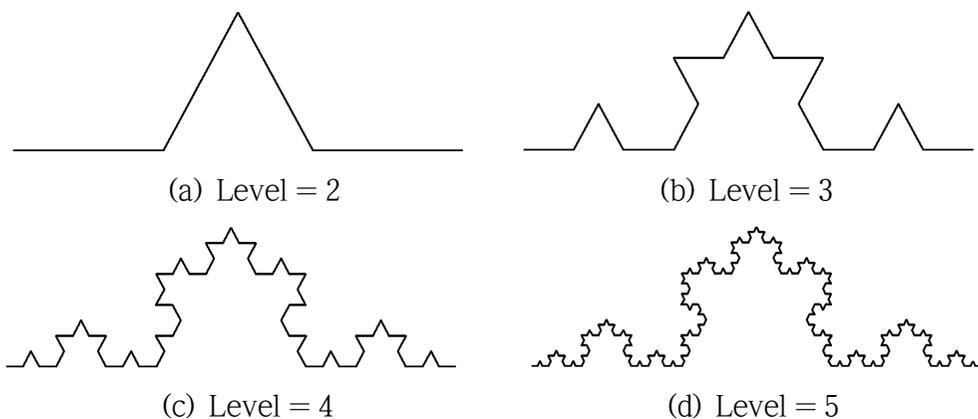


Figure 2.4 Level of Koch snowflake (Opened)

2.3.3 Apollonian gasket

An Apollonian gasket created by using a circle of Euclidean geometry is a shape frequently used in the fractal geometry. It is created by a sand artist Jim Denevan and his three associates [18]. The figures of Apollonian gasket according to the level are shown in Figure 2.5. The first figure of Apollonian gasket consists of large and small circles and which were drawn repeatedly on a flat desert to give a property. It was named after the ancient Greek mathematician 'Apollonius' who used terms including ellipse and arc for the first time in the book that is titled 'Conic Sections'. A gasket as an automotive part is a mechanical seal which fills the space between pipes or engine to prevent leakage of gas or oil. There are several kinds of methods to make Apollonian gasket figure. It depends on the way the first figure starts.

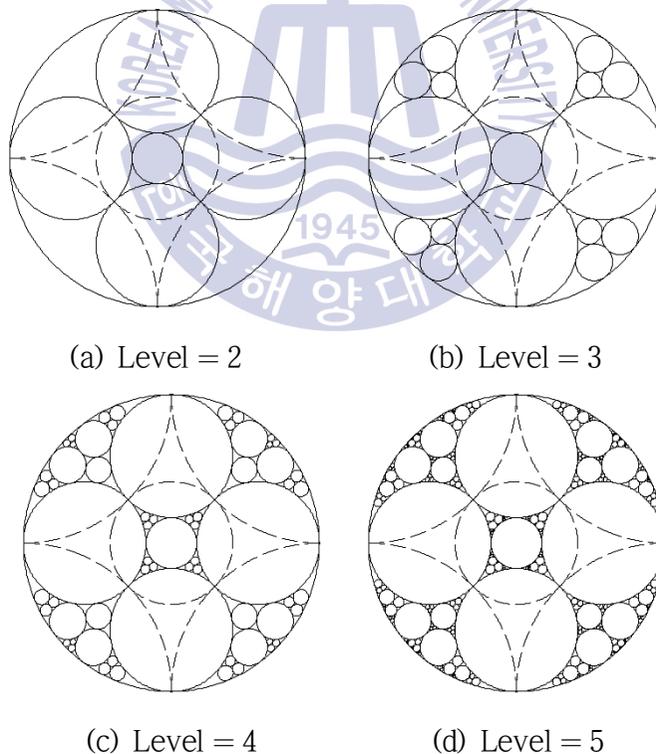


Figure 2.5 Apollonian gasket

To draw a normal Apollonian gasket, a regular triangle is drawn first. Then, three adjoining circles with a radius which equals to a half of one side of the regular triangle are drawn. Next, a circle which adjoins with three circles at the same time should be drawn, and one circle inside the circle and two circles outside the circle are drawn. Next, when a circle which adjoins three circles and two additionally drawn circles at the same time is drawn, a total of eleven circles can be obtained. When this process is repeated, a total of $3^{n-1} + 2$ circles can be obtained in n th steps. When it is drawn infinitely, the figure where a circle is completely filled but it does not become a plane. Its dimension is 1.654.

2.3.4 Vicsek fractal

Vicsek fractal is a fractal geometry arising from a construction similar to that of the Sierpinski carpet, proposed by Tamas Vicsek [19]. The figures of Vicsek fractal according to the level are shown in Figure 2.6.

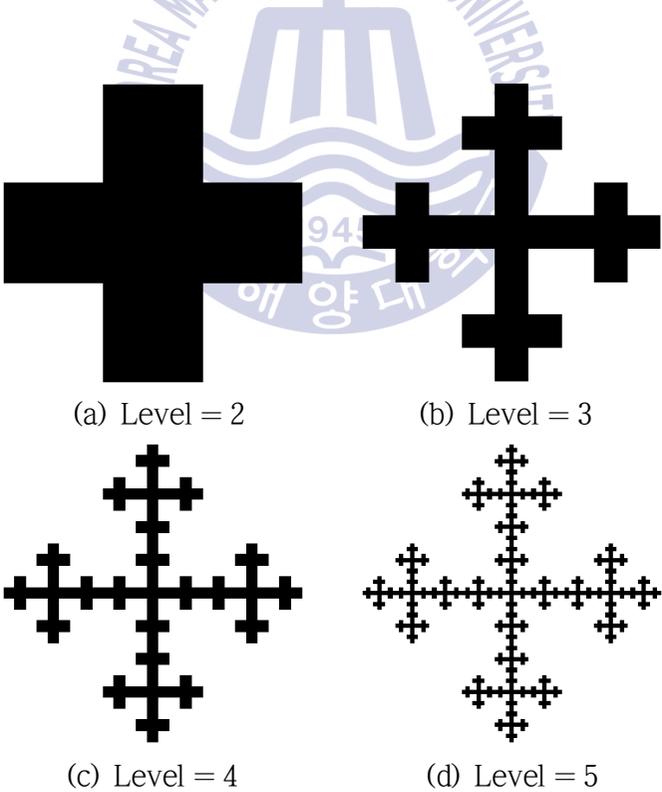


Figure 2.6 Vicsek fractal

It begins from a square figure. The square is divided into nine smaller squares in the 3-by-3 grid and then the squares at the corners are removed. Then, 5 out of 9 squares remain. In the next level, the same process is repeated for each of the five remaining squares. When the experiment according to the level is repeated by n th times, the area of the figure is $(5/9)^{n-1}$ and $5/9$ is a number smaller than 1, when it is repeated infinitely, its area is close to 0. Its dimension is $\log 5 / \log 3$ which is approximately 1.465.

2.3.5 Sierpinski triangle

Sierpinski triangle, named after the Polish mathematician Waclaw Franciszek Sierpinski in 1917, is also called the Sierpinski gasket. The figures of Sierpinski triangle according to each level are shown in Figure 2.7.

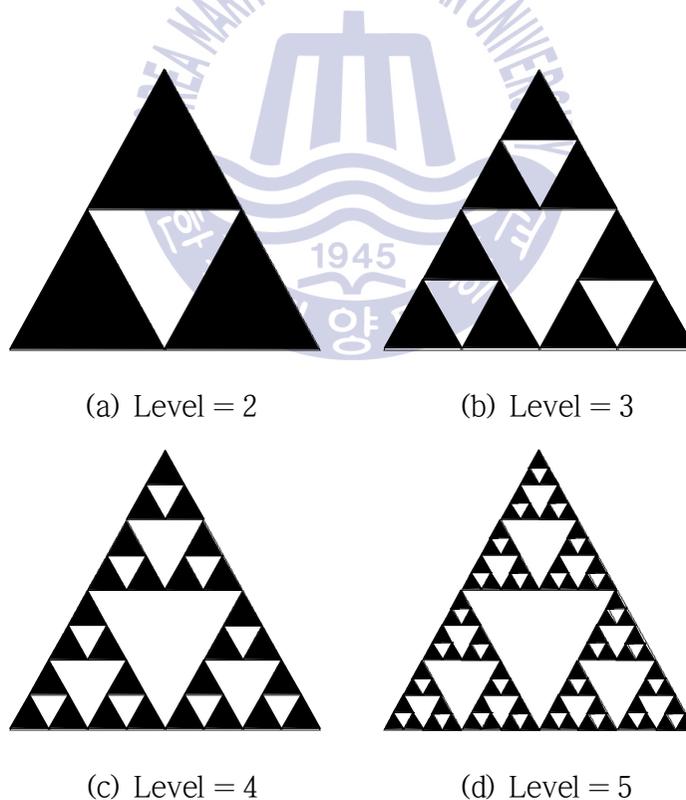


Figure 2.7 Sierpinski triangle

To draw a Sierpinski triangle, an equilateral triangle is drawn first and the center of each side of the equilateral triangle is connected to each other with a line. Then, a total of four equilateral triangles are created and the central one is removed. A figure obtained by repeating the same process for the remaining three equilateral triangles is called Sierpinski triangle. Each side is reduced by $1/2$ and the number of triangles increases threefold, so the dimension is $\log 3 / \log 2$ that is 1.585. When the above process is repeated, the sum of length of sides of equilateral triangle in n th step becomes $(3/2)^{n-1}$ and when it is carried out infinitely, the sum becomes an infinite value. The area becomes $(3/4)^{n-1}$ in n th step, and when it is carried out infinitely, the value becomes 0 [16].

2.3.6 Rand cantor

Rand cantor is a type of cantor set. The figures of Rand cantor according to each level are shown in Figure 2.8. The cantor set is a type of fractal that

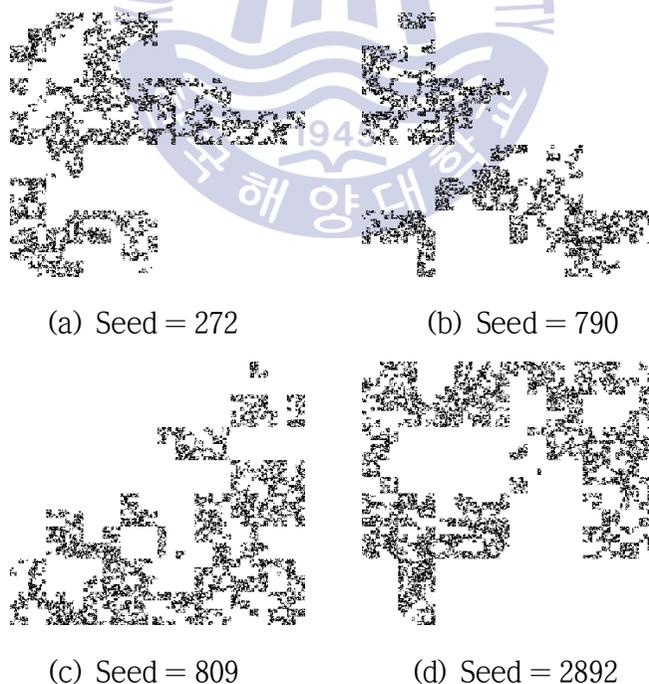


Figure 2.8 Cantor set

has the characteristics of self-similarity, and the German mathematician Georg Cantor discovered a method to count a transfinite number and introduced the concept of fractal geometry along with an example of cantor set in 1883 [16].

To draw the cantor set, a closed interval $[0,1]$ is given first, it is divided by three sections and the middle section $1/3$ is removed. The remaining two sections are divided by three sections each and the middle section is removed. Then, the sets $\{[0, 1/9], [2/9, 1/3], [2/3, 7/9], [8/9, 1]\}$ remain. The set obtained by carrying out this process recursively is called the cantor set [16]. In a normal cantor set, the middle segment is removed sequentially, but in a Rand cantor, segments are removed randomly to create a shape.

2.3.7 Koch curve 85°

Koch curve 85° is a fractal object designed from the fractal figure originated from the Koch snowflake. The figures of Koch curve 85° according to each level are shown in Figure 2.9.

In a normal Koch snowflake [16], one segment is divided by three to have approximately 60° for the angle between the 0° segment and the vertex of the triangle, and Koch curve 85° is a figure which has 85° for the angle between the segment and the vertex of the triangle.

First, an isosceles triangle with the vertex angle of 10° is drawn, and the base side is erased and one side of length which is same with the size of the isosceles triangle is drawn on each side. When this process is carried out repeatedly, a fractal figure as shown in the figure below is obtained. A theoretical fractal dimension of Koch curve 85° is 1.785.

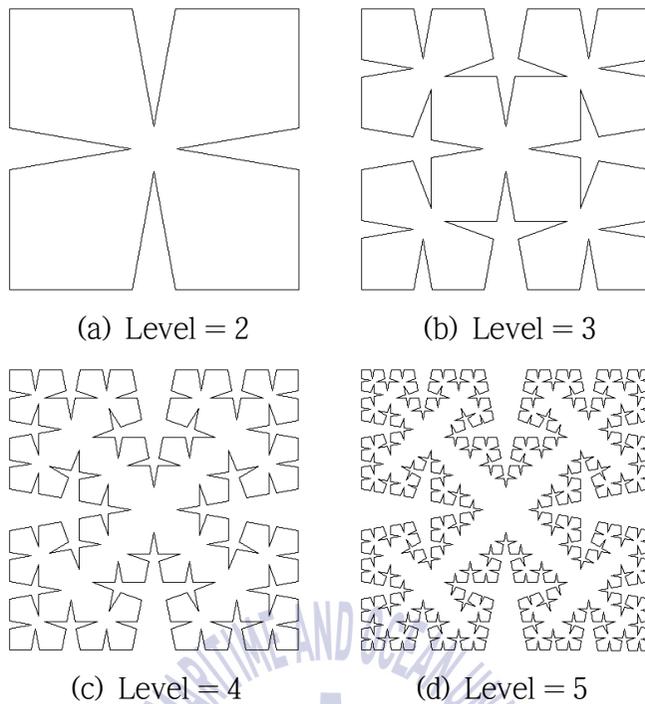


Figure 2.9 Koch curve 85°

2.3.8 Sierpinski carpet

Sierpinski carpet was named after the Polish mathematician W. F. Sierpinski in 1916. The figures of Sierpinski carpet according to each level are shown in Figure 2.10.

To create this, a square is first necessary. At first, the square is cut into 9 congruent sub-squares in a 3-by-3 grid and the central sub-square is removed. To move to the next level, each of the 8 remaining sub-squares are cut into congruent sub-squares in a 3-by-3 grid and each of central sub-square is removed. The Sierpinski carpet can be obtained by repeating the same procedure infinitely [17]. When it is shown in a formula, the area of Sierpinski carpet in n th step is $(8/9)^n$ and by taking the limit, becoming the same to the principle of Sierpinski triangle.

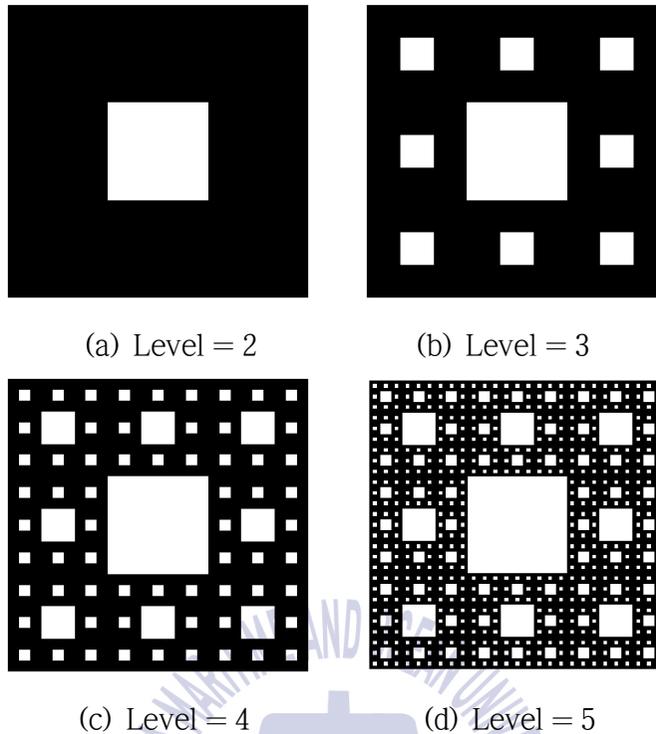
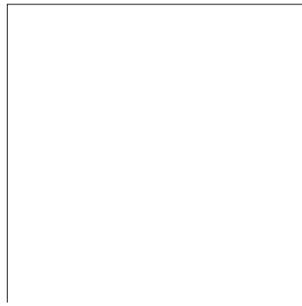


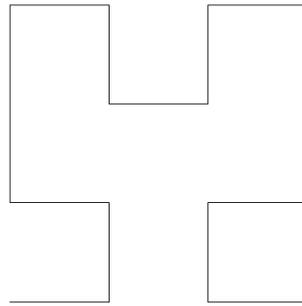
Figure 2.10 Sierpinski carpet

2.3.9 Hilbert curve

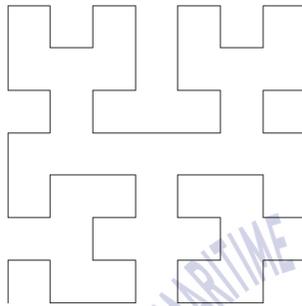
A Hilbert curve is a space-filling curve that was described by the German mathematician David Hilbert in 19th century. The figures of Hilbert curve according to each level are shown in Figure 2.11. This is a variant curve of the Peano curves discovered by Giuseppe Peano [20]. As the Peano curve, the Hilbert curve begins from a plane line, creates and adds a line infinitely, filling the plane completely and becoming a figure in two dimensions, and its dimension is 2. The Hilbert curve is a fractal figure which is created in the unit square. Level 1 begins as three segments by connecting 4 dots of the unit square $(1/4, 3/4)$, $(1/4, 1/4)$, $(3/4, 1/4)$, $(3/4, 3/4)$. In Level 2, all coordinates of the Level 1 are reduced by a half and a curve rotated counterclockwise based on $(0, 1/2)$ is added. Next, symmetrical lines based on $x = 1/2$ are added infinitely to obtain the Hilbert curve.



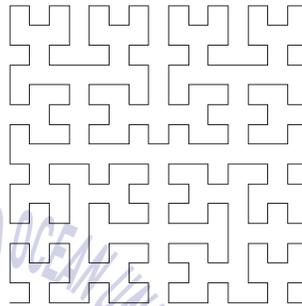
(a) Level = 1



(b) Level = 2

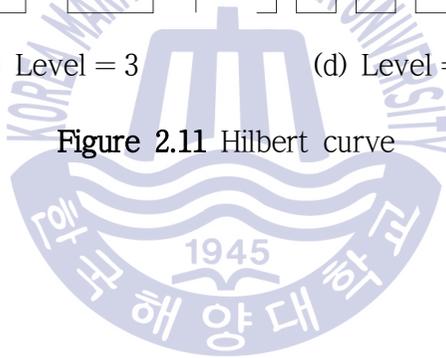


(c) Level = 3



(d) Level = 4

Figure 2.11 Hilbert curve



Chapter 3. Existing Box-Counting Methods

The concept of fractal dimension plays an important role in the fractal study. There have been several methods for estimating a fractal dimension include the BC method, triangular prism method and Fractional Brownian motion method [17]. Among these methods, the BC method suitable for two dimensional binary (black and white) images is used frequently because of its simplicity and high reliability. Therefore, the relevant study was carried out.

3.1 Conventional box-counting method

In the BC method, when a binary image is given, the fractal dimension is estimated by repeating an estimation process where a reference square (of which length of a side is 1) incorporating the image is created and then divided into grids by reducing the length of a side in a scale of r , and then the boxes including fractal figures are counted [1]. Therefore, the minimum number of boxes required to completely cover the fractal image, $N(r)$, is a function of r , expressed as follows:

$$N(r) = \left(\frac{1}{r}\right)^D \quad (3.1)$$

Not all fractals exhibit deterministic self-similarity but some statistical self-similarity.

In order to obtain approximate fractal dimension, we can use one of the commonly adopted algorithms, called the BC algorithm, introduced by Gangnepain and Roques-Carnes [21]. Given a binary (black and white) image I of $M \times M$ pixels, where $M=2^m$ and m is an integer number, Equation (3.1) can be rewritten from the relation between box (step) size δ and M , that is, $\delta=rM$ ($0 < r < 1$) as

$$N(\delta) = \left(\frac{1}{r}\right)^D = \left(\frac{M}{\delta}\right)^D = M^D \left(\frac{1}{\delta}\right)^D \quad (3.2)$$

Since M^D is a constant, Equation (3.2) may be rewritten as the following Equation (3.3):

$$N(\delta) \propto \left(\frac{1}{\delta}\right)^D \quad (3.3)$$

Step size δ needed to calculate $N(\delta)$ sampling has a direct effect on the precision of the fractal dimension estimate, the sampling method should be carefully decided. Representative sampling methods are the GS method, the AS method, and the DS method.

The GS method used with BC method is employed by most fractal dimension estimation methods as in the present study. To make the data intervals on the log-log coordinates become equal, the GS method employs geometric numbers increasing in the multiples of 2, which are the powers of 2. Figure 3.1 shows an example of calculating $N(\delta)$ while sampling an image of 16×16 pixels representing Koch snowflake curve. Table 3.1 shows the results.

Table 3.1 Step size and box numbers

δ	$N(\delta)$	$\log(1/\delta)$	$\log(N(\delta))$
1	46	0	1.6628
2	20	-0.3010	1.3010
4	6	-0.6021	0.7782
8	4	-0.9031	0.6021

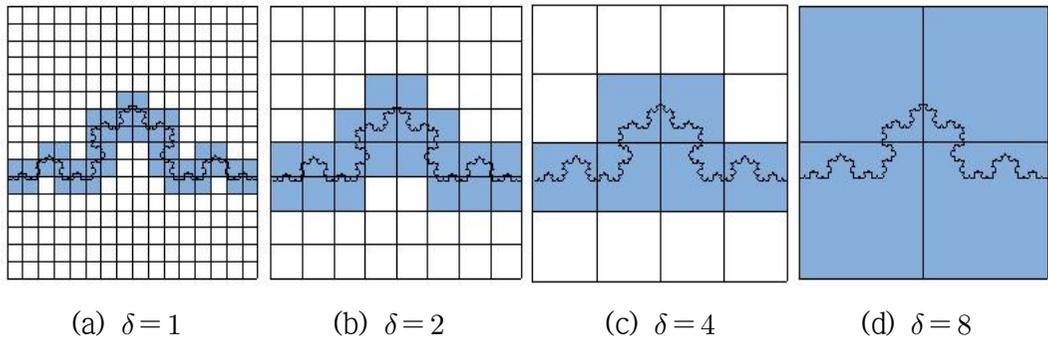


Figure 3.1 Box counting on an image of 16×16 pixels

Hence, $N(\delta)$ is obtained by repeating the process of dividing the image while changing δ and counting the boxes including fractal figures. The fractal dimension is the gradient of the $\log(1/\delta) - \log(N(\delta))$ plot which is expressed as:

$$D = \frac{\log(N(\delta))}{\log(1/\delta)} \quad (3.4)$$

If $\{\delta_k, N(\delta_k)\}$ ($1 \leq k \leq m$) is given as the number of boxes corresponding to the given step size, it can be obtained also through least square technique by the regression analysis of linear model. The relationship between δ and $N(\delta)$ can be expressed as a linear equation as shown in Equation (3.5)

$$\log(N(\delta)) = D \cdot \log(1/\delta) + a + \epsilon \quad (3.5)$$

and in Equation (3.5), a is an intersection point between the linear equation and y-axis, and ϵ is an error between the linear equation and data.

When substituting data $\{\delta_k, N(\delta_k)\}$ ($1 \leq k \leq m$) for Equation (3.5), it can simply be expressed by Equation (3.6).

$$\mathbf{b} = \mathbf{Ax} + \boldsymbol{\epsilon} \quad (3.6a)$$

$$\mathbf{b} = \begin{bmatrix} \log(N(\delta_1)) \\ \log(N(\delta_2)) \\ \vdots \\ \log(N(\delta_m)) \end{bmatrix} \in R^m, \quad \mathbf{x} = \begin{bmatrix} D \\ a \end{bmatrix} \in R^2, \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{bmatrix} \in R^m, \quad \mathbf{A} = \begin{bmatrix} \log(1/\delta_1) & 1 \\ \log(1/\delta_2) & 1 \\ \vdots & \vdots \\ \log(1/\delta_m) & 1 \end{bmatrix} \quad (3.6b)$$

When the sum of error square as the estimation function is J

$$J(\mathbf{x}) = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = (\mathbf{b} - \mathbf{Ax})^T (\mathbf{b} - \mathbf{Ax}) \quad (3.7)$$

$\hat{\mathbf{x}}$ estimated from the necessary condition $\partial J(\mathbf{x}) / \partial \mathbf{x} = 0$ for this value to be the minimum is given as Equation (3.8).

$$\hat{\mathbf{x}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (3.8)$$

Therefore, the fractal dimension is obtained from Equation (3.8) as $D = \hat{x}_1$. When the approximate fractal dimension calculated using the data of boundary value on both sides from Table 3.1 can be obtained with the least square method in Equation (3.5) as

$$D = \frac{\log N(1) - \log N(8)}{\log 1 - \log \frac{1}{8}} = \frac{1.6628 - 0.6021}{0 + 0.9031} = 1.17, \text{ and } D = 1.23$$

It can be known that this is approximate to $D = 1.262$, theoretical dimension of Koch snowflake curve. Also, Figure 3.2 shows that the gradient of the graph means fractal dimension of koch snowflake curve.

Table 3.2 shows overall algorithm of the BC method with $M \times M$ images to calculate fractal dimension.

Table 3.2 Overall procedure of the BC method

The conventional BC algorithm
<pre> Get an image I of $M \times M$; $m = \log_2(M)$; for $k=1$ to m $\delta_k = 2^{k-1}$; Blocks = M/δ_k; $N(k) = 0$; for $i=1$ to Blocks for $j=1$ to Blocks if sub-image $I(i,j)$ is a non-empty box $n_{ij} = 1$; else $n_{ij} = 0$; end if $N(k) = N(k) + n_{ij}$; end for end for end for Estimate D using data set $\{\delta_k, N(k)\}$ ($1 \leq k \leq m$) of m and the least squares method; </pre>

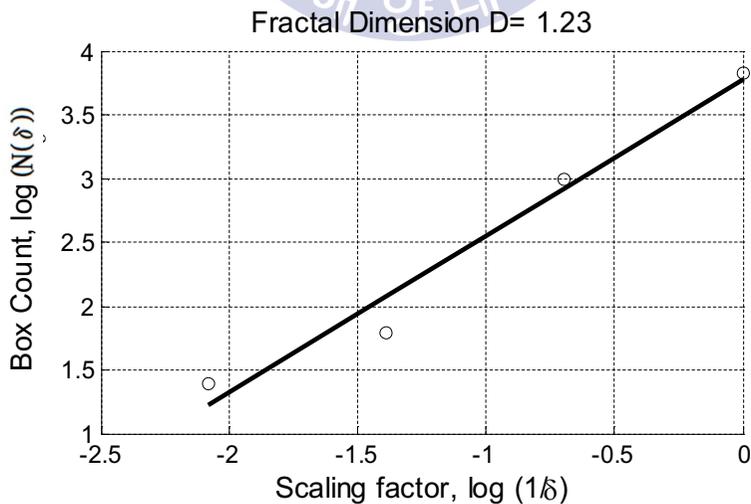


Figure 3.2 Data fitting using the least squares method

3.2 Triangle box-counting method

The procedure of the TBC method, suggested by Woraratpanya et al. [13], is similar with that of the BC method shown in Table 3.1 and the major difference is that the box of each $\delta \times \delta$ pixel size is counted separately into two pattern areas as shown in Figure 3.3. Therefore, it is counted as $n_{ij} = 2$ in case of Figure 3.4(a) and $n_{ij} = 1$ in case of Figure 3.4(b).

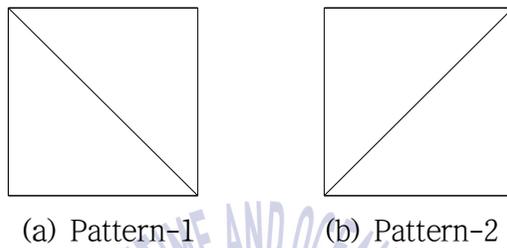


Figure 3.3 Patterns of partitioning a square box

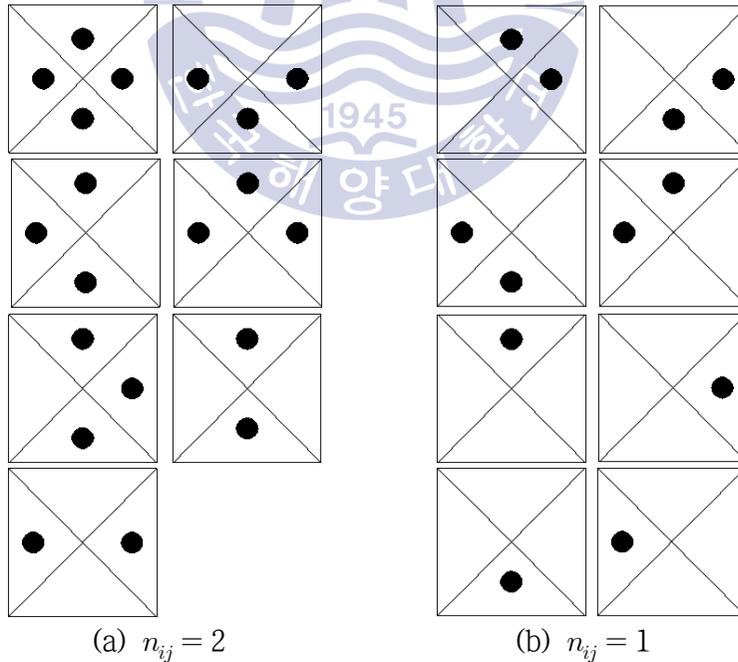


Figure 3.4 Box-counting of the TBC method

Given an image of size $M \times M$, the overall algorithm of the TBC method is shown below in Table 3.3.

Table 3.3 Overall procedure of the TBC method

The TBC algorithm
<pre> Get an image I of $M \times M$; $m = \log_2(M)$; for $k=1$ to m $\delta_k = 2^{k-1}$; Blocks = M/δ_k; $N(k) = 0$; for $i=1$ to Blocks for $j=1$ to Blocks Split sub-image $I(i,j)$ into two equally triangle boxes as shown in Figure 3.3 Count non-empty boxes in both patterns, such that C_1 and C_2 denote counter variables for Pattern-1 and Pattern-2, respectively if C_1 and C_2 are equal to 2 $n_{ij} = 2$; else if C_1 and C_2 are equal to 1 $n_{ij} = 1$; else if C_1 is not equal to C_2, such that C_1 and C_2 are greater than 0 $n_{ij} = \min[C_1, C_2]$; else if $n_{ij} = \max[C_1, C_2]$; else $n_{ij} = 0$; end if $N(k) = N(k) + n_{ij}$; end for end for end for Estimate D using data set $\{\delta_k, N(k)\}$ ($1 \leq k \leq m$) of m and the least squares method; </pre>

Chapter 4. Enhanced BC Method

In the previous chapter, three data sampling methods were mentioned. Their principles and drawbacks are in this chapter reviewed and a new sampling method and fractional counting method are suggested to improve the BC method.

4.1 Existing sampling methods and their drawbacks

4.1.1 Sampling methods

In the GS method, which is employed by most fractal estimation methods, sampling is performed in geometric numbers increasing in the multiples of 2 to make the data intervals on the log coordinate become equal. The step set and step size are shown below.

For example, applying the GS method to an 8×8 image gives $\Delta_{GS} = \{1, 2, 4\}$ and $n_{GS} = 3$.

$$\Delta_{GS} = \{\delta | 2^{k-1}, k=1, 2, 3, \dots, \log_2(M)\} \quad (4.1)$$

$$n_{GS} = |\Delta_{GS}| = \log_2(M) \quad (4.2)$$

Figure 4.1 shows the results of applying the GS method.

In the AS method, which has been proposed for the purpose of obtaining sufficient data for regression analysis of a small size of image, δ is arithmetically increased by 1 at each time from 1 to $M/2$ in the sampling process. The step set and step size are shown as:

$$\Delta_{AS} = \{\delta \mid \delta = 1, 2, 3, \dots, M/2\} \quad (4.3)$$

$$n_{AS} = |\Delta_{AS}| = M/2 \quad (4.4)$$

Application of the AS method to the previously mentioned image of 8×8 pixels gives $\Delta_{AS} = \{1, 2, 3, 4\}$, and $n_{AS} = 4$. Figure 4.2 shows the results of applying the AS method. When $\delta = 3$, 28 pixels are wasted.

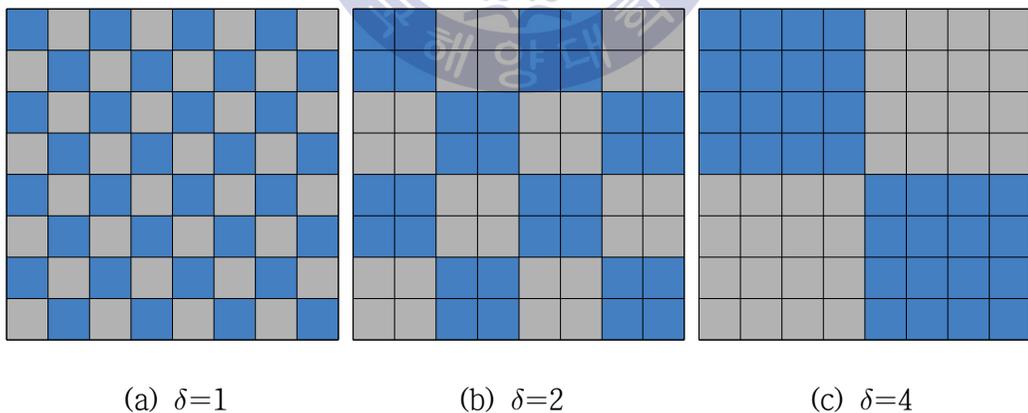


Figure 4.1 GS sampling on an image of 8×8 pixels

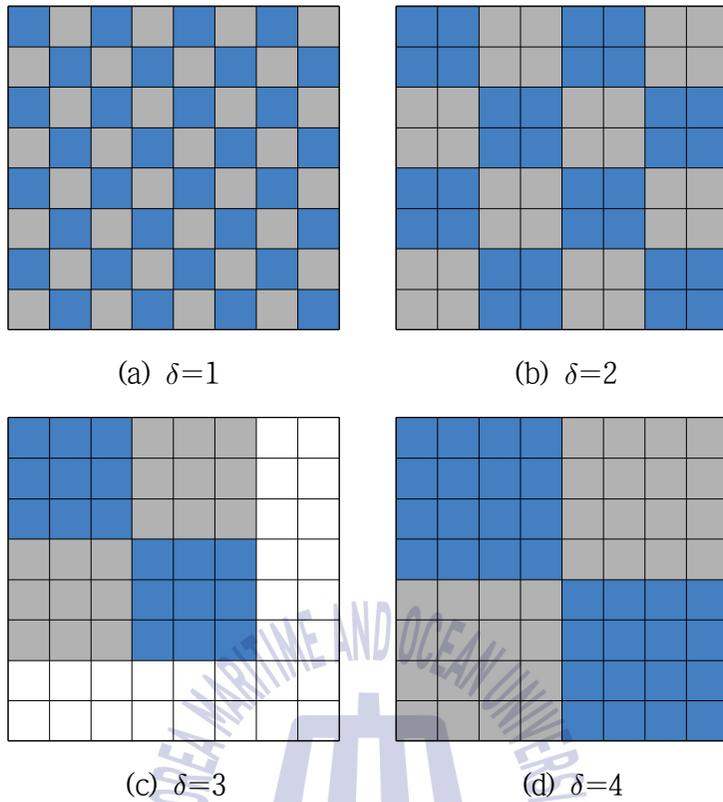


Figure 4.2 AS sampling on an image of 8×8 pixels

In the DS method, which is a method designed to accomplish 100 [%] of pixel utilization rate at any δ , δ is a natural number divisor (except M). The step set and step size are given by

$$\Delta_{DS} = \{ \delta \mid \delta \text{ is a natural number divisor of } M \text{ and } \delta \neq M \} \quad (4.5)$$

$$n_{DS} = |\Delta_{DS}| \quad (4.6)$$

If M is a geometric number, the DS method becomes identical to the GS method. Application of the DS method to the same image of 8×8 pixels gives $\Delta_{DS} = \{1, 2, 4\}$ and $n_{DS} = 3$. The sampling result is equal to that shown in Figure 4.1.

4.1.2 Pixel utilization

The coverage ratio $CR(\delta)$, which is the pixel utilization ratio to measure how many pixels out of the whole pixels are used or deleted according to δ when the sampling method is applied to an image of $M \times N$ pixels, is defined as follows:

$$CR(\delta) = \frac{\text{int}(\frac{M}{\delta})\delta \times \text{int}(\frac{N}{\delta})\delta}{M \times N}, \quad (\delta \in \Delta) \quad (4.7)$$

where $\text{int}(\frac{M}{\delta})\delta \times \text{int}(\frac{N}{\delta})\delta$ denotes the number of pixels with a given step size of δ , and Δ denotes the step set. In addition, with respect to all step sizes, the average coverage ratio \overline{CR} is defined as:

$$\overline{CR} = \frac{1}{|\Delta|} \sum_{\delta \in \Delta} CR(\delta) \quad (4.8)$$

Then, average wasting ratio \overline{WR} is defined as follows:

$$\overline{WR} = 1 - \overline{CR} \quad (4.9)$$

4.1.3 Drawbacks of the existing sampling methods

In the case of the BC method or the Ruler's method, the degree of accuracy is improved when measuring a fractal dimension as generally the scale r is larger, in other words, the step size δ is smaller. The fact that when M becomes larger, the GS method has a disadvantage that there are less data in high precision scale, and the AS method has a large step size so that enough data for regression analysis can be obtained but a block with low pixel utilization rate is

sampled several times so that the degree of accuracy is not improved while only increasing the burden of calculation. The DS method secures 100 [%] pixel utilization rate for $M=2^m$ (m is a positive integer) image but when $M \neq 2^m$ and M is a prime number, the number of data is insufficient.

Next, the number of steps was obtained with the GS method, the AS method, and the DS method by increasing M with an increment of 1 from 10 to 50 and the graph was drawn. According to Figure 4.3, the GS method and the AS method tend to show a monotonous increase as M increases, but the GS method has too narrow margin of increase, and in case of the DS method, the number of steps increases or decreases repeatedly according to M and the number of steps which is 1 occurs frequently, and in case of the AS methods, the number of steps increases in proportion to M but the pixel utilization rate is low and too many small boxes are created so that the estimated value is skewed accordingly.

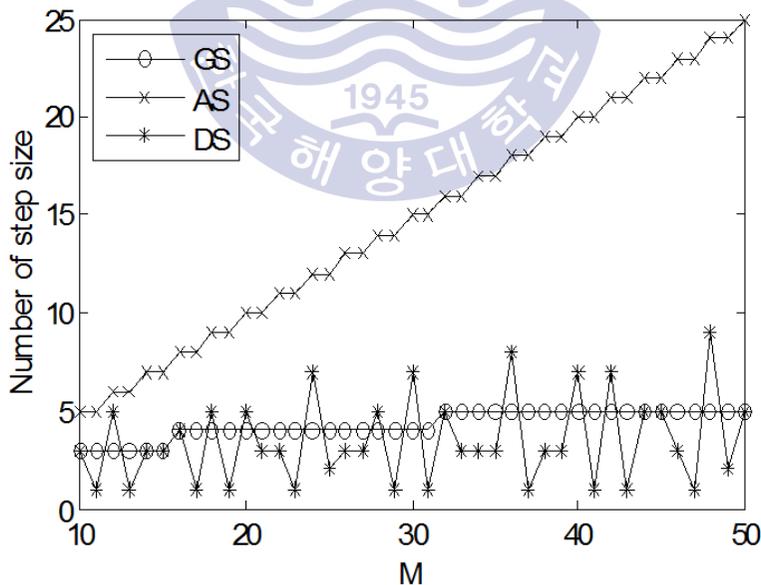


Figure 4.3 Number of step size to changes of M

4.2 New sampling method

It is critical in the BC method to determine the step size and the limits of the step size. In the GS method, the data in the scale of which precision may be increased (i.e., small δ) are insufficient if M is large. Also, when it is applied to $M \times N$ image, some pixels cannot be utilized. Therefore, in this study, a novel sampling method is herein proposed to resolve these problems on the basis of the intuitive observation that an estimate may have a higher degree of precision if more pixels are utilized in each step and sufficiently large quantity of regression analysis data are secured.

The suggested method is an improved version of the previous GS method, called the modified GS (MGS) method. The MGS method takes the middle value from the integer section on the log-log coordinates for the step set of the GS method to combine δ where \overline{CR} is better.

When the image size is $M \times N$, partition should be done based on the length of a short side, so if $L = \min(M, N)$, the step set of the GS method will become Equation (4.10).

$$\Delta_{GS} = \{\delta \mid \delta = 2^{k-1}, k = 1, 2, 3, \dots, \ell\} \quad (4.10)$$

Here, $\ell = \text{int}(\log_2(L)) = n_{GS}$. The step set which takes the middle value from the integer section on the log-log coordinates becomes Equation (4.11). Here $\text{round}()$ means the round-off operator.

$$\Delta_{MID} = \{\delta \mid \delta = \text{round}(2^{(k-\frac{1}{2})}) \leq \frac{L}{2}, k = 1, 2, 3, \dots, \ell\} \quad (4.11)$$

The remainder between Δ_{MID} and Δ_{GS} to eliminate the overlapping step size due to real exponent and rounding off operation can be obtained by Equation (4.12).

$$\Delta_{DIFF} = \Delta_{MID} - \Delta_{GS} = \{\delta | \delta \in \Delta_{MID} \text{ and } \delta \notin \Delta_{GS}\} \quad (4.12)$$

$CR(\delta)$ of $\delta \in \Delta_{DIFF}$ is obtained and lined up in descending order in order to include a large number of δ which has high pixel utilization rate while securing more data than the GS method, and the step set Δ_{ADD} is composed by taking the corresponding δ as many as n_{ADD} according to size.

$$n_{ADD} = \min(n_{DIFF}, \text{round}(\eta \cdot \ell)) \quad (4.13)$$

where η is a real positive value between 0 and 1 and a user-defined variable. Therefore, the suggested sampling method is the sum of Δ_{GS} and Δ_{ADD} .

$$\Delta_{MGS} = \Delta_{GS} \cup \Delta_{ADD} \quad (4.14)$$

The overall procedure of the proposed MGS method is shown in Table 4.1.

Table 4.1 Procedure of the MGS method

The MGS method
Get M, N and η ;
Set $L = \min(M, N)$;
Set $\ell = \text{int}(\log_2(L))$;
Calculate the set Δ_{GS} using Equation (4.10);
Calculate the additional set Δ_{ADD} using Equations (4.11)~(4.13);
Set $\Delta_{MGS} = \Delta_{GS} \cup \Delta_{ADD}$;

For example, when the MGS method is applied to 17×21 image with $\eta = 0.6$, $L = \min(M, N) = \min(17, 21) = 17$, $\ell = \text{int}(\log_2(L)) = 4$, $\Delta_{GS} = \{1, 2, 4, 8\}$ and $\Delta_{MID} = \{1, 3, 6\}$. Therefore, $\Delta_{DIFF} = \Delta_{MID} - \Delta_{GS} = \{3, 6\}$ and $n_{ADD} = \min(n_{DIFF}, \text{round}(\eta \cdot \ell))$

$= \min(2, \text{round}(0.6 \times 4)) = 2$, $\Delta_{ADD} = \{3, 6\}$, $\Delta_{MGS} = \Delta_{GS} \cup \Delta_{ADD} = \{1, 2, 3, 4, 6, 8\}$ in the end. As another example, the result of applying the GS method and the MGS method to a 191×256 image is shown in Table 4.2 and Figure 4.4 shows a drawing of $CR(\delta)$.

Table 4.2 Sampling results of the GS and MGS methods on 191×256 image

Sampling method	Step size δ							Average coverage ratio (\overline{CR}) (Standard deviation)
GS	1	2	4	8	16	32	64	0.910 (0.111)
MGS	1	2	3	4	6	8	11	0.903 (0.111)
	16	23	32	45	64	91		

According to the table and the figure, the average coverage ratio \overline{CR} of the MGS method and the GS method is similar but the number of steps of the MGS method is 13 and the number of steps of the GS method is 7, indicating that the MGS method can secure more regression analysis data.

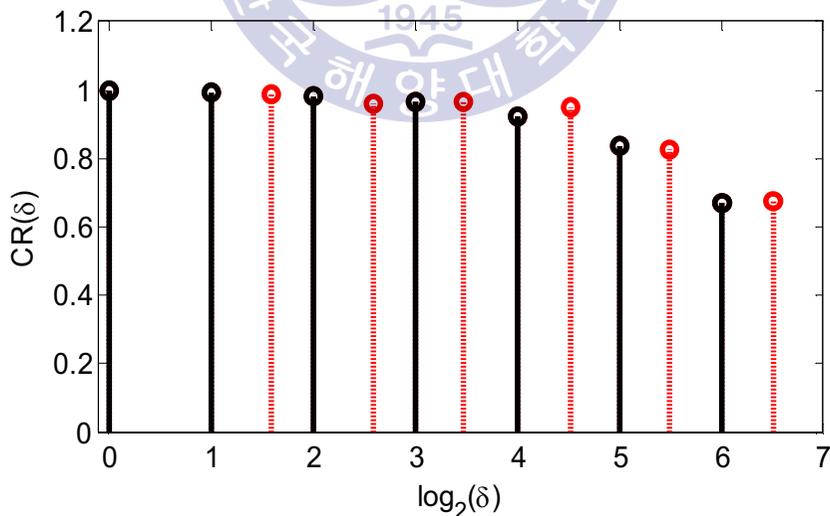


Figure 4.4 Coverage ratio of the GS and MGS sampling methods
(Solid: GS method, Solid+Dotted: MGS method)

4.3 Fractional box count

As described above, the conventional BC method and most of the methods which can estimate fractal dimension, where the sampling is performed by the GS method, limits that the image size is $M \times M = 2^m \times 2^m$ (m is a positive integer) to prevent waste of pixels. Also, when large images are treated to calculate fractal dimension, the data for regression analysis are not enough.

On the average the size of images acquired from actual environment, such as the shapes of coastline, rivers, chains of mountains, is actually various. Therefore, this study suggests a real number counting method to resolve the problem of pixel waste which occurs when the BC method is applied to images of an arbitrary size.

Once δ is determined, and $m = \text{int}(M/\delta)$ and $n = \text{int}(N/\delta)$, the image is divided as shown below to make each block to be a square or a rectangle.

- If $M = m\delta$ and $N = n\delta$, the image is divided into m, n blocks of $\delta \times \delta$ pixels.
- If $M > m\delta$ and $N = n\delta$, the image is divided into m, n blocks of $\delta \times \delta$ pixels and n blocks of $(M - m\delta) \times \delta$ pixels.
- If $M = m\delta$ and $N > n\delta$, the image is divided into m, n blocks of $\delta \times \delta$ pixels and m blocks of $\delta \times (N - n\delta)$ pixels.
- If $M > m\delta$ and $N > n\delta$, the image is divided into m, n blocks of $\delta \times \delta$ pixels, n blocks of $(M - m\delta) \times \delta$ pixels, m blocks of $\delta \times (N - n\delta)$ pixels, and one block of $(M - m\delta) \times (N - n\delta)$ pixels.

Figure 4.5 shows an example of a image divided with step size δ .

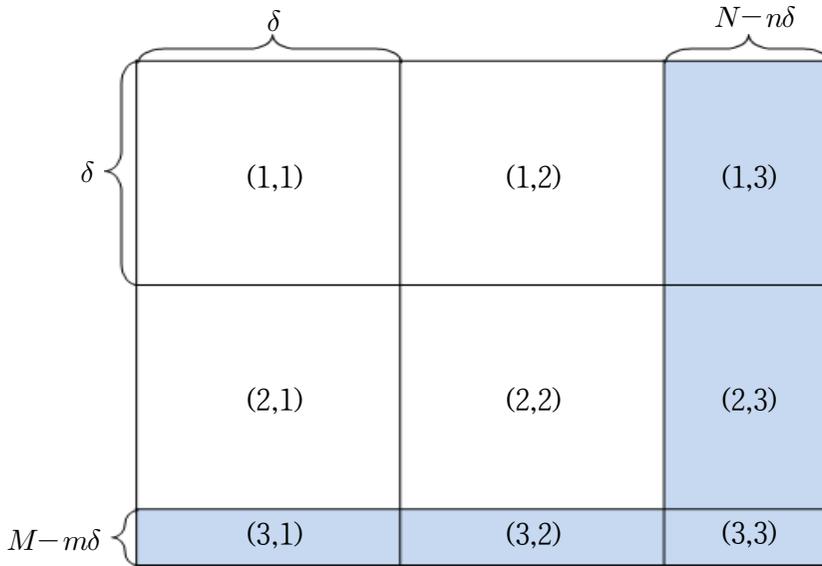


Figure 4.5 Partition example of an image

p_{ij} denotes the number of pixels of the fractal figure included in the (i,j) block and \bar{p}_δ denotes the average pixel number of the fractal figure included in the block of a $\delta \times \delta$ size which is counted as 1.

- If the block size is $(M - m\delta) \times \delta$, the block are counted by the real number counting method. If any fractal figure is included in the (i,j) block, the count is done by Equation (4.15); otherwise $n_{ij} = 0$:

$$n_{ij} = \min\left(\frac{p_{ij}}{\bar{p}_\delta}, 1\right) \cdot \frac{(M - m\delta)}{\delta} \quad (4.15)$$

- If the block size is $\delta \times (N - n\delta)$, the block are counted by the real number counting method. If any fractal figure is included in the (i,j) block, the count is done by Equation (4.16); otherwise $n_{ij} = 0$:

$$n_{ij} = \min\left(\frac{p_{ij}}{\bar{p}_\delta}, 1\right) \cdot \frac{(N - n\delta)}{\delta} \quad (4.16)$$

- If the block size is $(M-m\delta)\times(N-n\delta)$, the block are counted by the real number counting method. If any fractal figure is included in the (i,j) block, the count is done by Equation (4.17); otherwise $n_{ij}=0$;

$$n_{ij} = \min\left(\frac{p_{ij}}{p_\delta}, 1\right) \cdot \frac{(M-m\delta)(N-n\delta)}{\delta^2} \quad (4.17)$$

For instance, application of the proposed MGS method and the fractional counting method to the 17×21 image, $L = \min(M,N) = \min(17,21) = 17$, $\ell = \text{int}(\log_2(L)) = 4$, $m = \text{int}(17/8) = 2$, $n = \text{int}(21/8) = 2$, $\Delta_{MGS} = \{1,2,3,4,6,8\}$.

Figure 4.6 shows how the image can be divided and Table 4.3 shows the result of $N(\delta)$.

Table 4.3 Step size and real number box-counting

Step size δ	1	2	3	4	6	8
Box Count $N(\delta)$	56	21.19	10	8.04	4.24	4.61

When we see Figure 4.6(f) where $\delta=8$, 4 blocks sized 8×8 , that is, (1,1), (1,2), (2,1), (2,2)th blocks are counted as integers and the (2,3)th block sized 8×5 that includes the fractal figure is counted as a real number. The integer counting result is 4 and the (2,3)th block is counted as follows.

By referring to Figure 4.6(a) and Figure 4.6(f), the average pixel number of the fractal figures included in 4 blocks of 8×8 is $\bar{p}_\delta = 11.25$ and also $p_{2,3} = 11$, so

$$n_{2,3} = \min\left(\frac{p_{2,3}}{\bar{p}_\delta}, 1\right) \times \frac{(N-n\delta)}{\delta} = \frac{11}{11.25} \times \frac{5}{8} = 0.61. \text{ Therefore, } N(\delta) = 4 + 0.61 = 4.61. \text{ When}$$

$N(\delta)$ is obtained for each δ , the fractal dimension is obtained through the least square technique, and in case of this example, $D=1.279$.

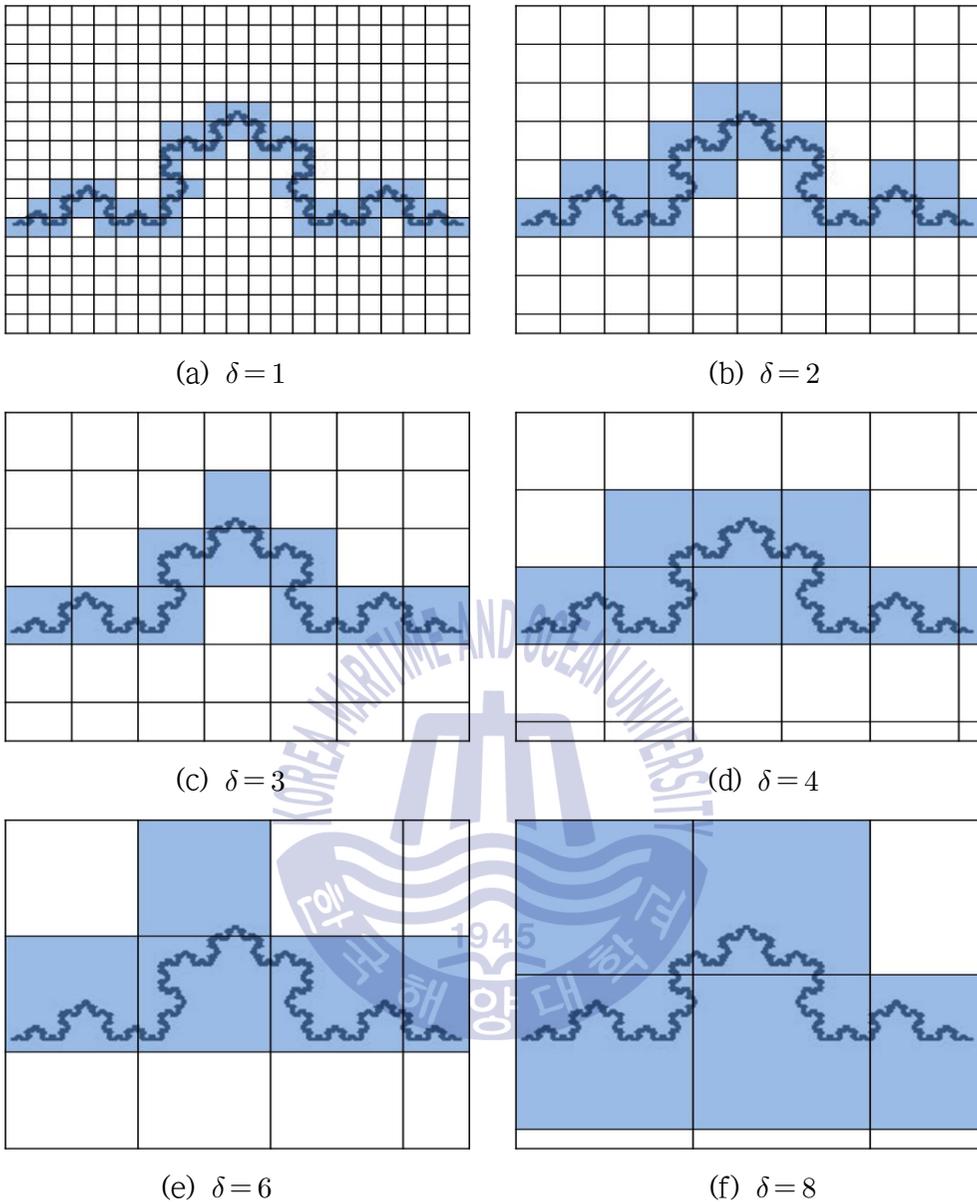


Figure 4.6 Sampling of the image of 17×21

4.4 Procedure of the enhanced BC method

The overall algorithm of the proposed BC method in this study is shown in Table 4.4.

Table 4.4 Overall procedure of the enhanced BC method

The enhanced BC algorithm
<pre> Get an image I of $M \times N$ and η; Calculate Δ_{MGS} using Equations (4.10)~(4.14) and $\ell_{MGS} = \Delta_{MGS}$; for $k=1$ to ℓ_{MGS} rBlocks=$int(M/\delta_k)$; cBlocks=$int(N/\delta_k)$; $N(k) = 0$; for $i=1$ to rBlocks for $j=1$ to cBlocks if sub-image $I(i,j)$ is a non-empty box $n_{ij} = 1$; else $n_{ij} = 0$; end if $N(k) = N(k) + n_{ij}$; end for end for end for if $M > rBlocks \times \delta_k$ and $N = cBlocks \times \delta_k$ for $jj=1:cBlocks$ if sub-image $I(rBlocks+1,jj)$ is a non-empty box Calculate n_{ij} using Equation (4.15); else $n_{ij} = 0$; end if $N(k) = N(k) + n_{ij}$; end for end if if $N > cBlocks \times \delta_k$ and $M = rBlocks \times \delta_k$ for $ii=1:rBlocks$ </pre>

```

if sub-image  $I(ii, cBlocks + 1)$  is a non-empty box
    Calculate  $n_{ij}$  using Equation (4.16);
else
     $n_{ij} = 0$ ;
end if
 $N(k) = N(k) + n_{ij}$ ;
end for
end if
if  $M > rBlocks \times \delta_k$  and  $N > cBlocks \times \delta_k$ 
    if sub-image  $I(rBlocks + 1, cBlocks + 1)$  is a non-empty box
        Calculate  $n_{ij}$  using Equation (4.17);
    else
         $n_{ij} = 0$ ;
    end if
     $N(k) = N(k) + n_{ij}$ ;
end if
end for
Estimate  $D$  using data set  $\{\delta_k, N(k)\}$  ( $1 \leq k \leq m$ ) of  $m$  and the least squares
method;

```



Chapter 5. Experiments and Review

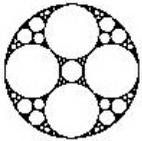
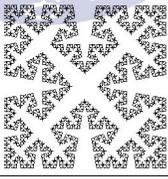
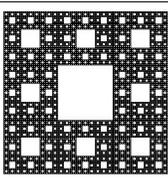
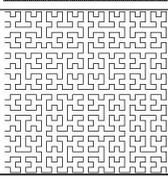
In this chapter, a set of simulation works are performed with deterministic fractal figures of which theoretical dimensions are well known to evaluate the estimation performance of the proposed BC method. The result of using the proposed BC method was compared with that of the conventional BC method and the TBC method. The proposed BC method also was applied to the non-deterministic images of coastlines of South Korea to estimate its complexity and compared with the other two methods.

5.1 Experiments on deterministic fractal image

5.1.1 $M \times M$ test image

For these experiments, eight fractal figures having a fractal dimension of 1 to 2 were selected and then drawn by using mathematical formulas. The resulting drawings were converted into images having pixel sizes of 128×128 , 256×256 , and 512×512 . Since the estimate of a fractal dimension is slightly dependent on the levels of the figures whose complexity are different, the images of individual figures drawn at five levels were used. Table 5.1 summarizes the figures used in the experiment and the theoretical dimensions as well as the levels.

Table 5.1 $M \times M$ test images for experiments

No.	Name of figure	Figure	Level	Dimension
1	Koch snowflake		4~8	1.262
2	Apollonian gasket		3~7	1.328
3	Vicsek fractal		3~7	1.465
4	Sierpinski triangle		5~9	1.585
5	Rand cantor		5 kinds of seeds	1.678
6	Koch curve 85°		5~9	1.785
7	Sierpinski carpet		3~7	1.893
8	Hilbert curve		6~10	2

5.1.2 Determination of η

In the proposed MGS method, η , the parameter determining how many step sizes will be selected from Δ_{DIFF} , was determined through the experiment performed with the images shown in Table 5.1. The evaluation function used to measure the estimation performance was the *MAE* between the estimated dimension, D_i , and the theoretical dimension, D_t , as expressed as follows:

$$MAE = \frac{1}{w} \sum_{i=1}^w |D_i - D_t| \quad (5.1)$$

where w denotes the total number of images used in the experiment.

Figure 5.1 shows the *MAE* calculated and drawn with respect to a total of 120 images of eight figures at five levels with 3 sizes by varying η from 0 to 1. As shown in Figure 5.1, when $0 < \eta \leq 0.6$, the performance of the proposed method was better than that of the GS method ($\eta = 0$), the best performance was obtained when $\eta = 0.6$, and the performance was worse than that of the GS method when $\eta > 0.6$, since many step sizes having a low pixel utilization rate were included.

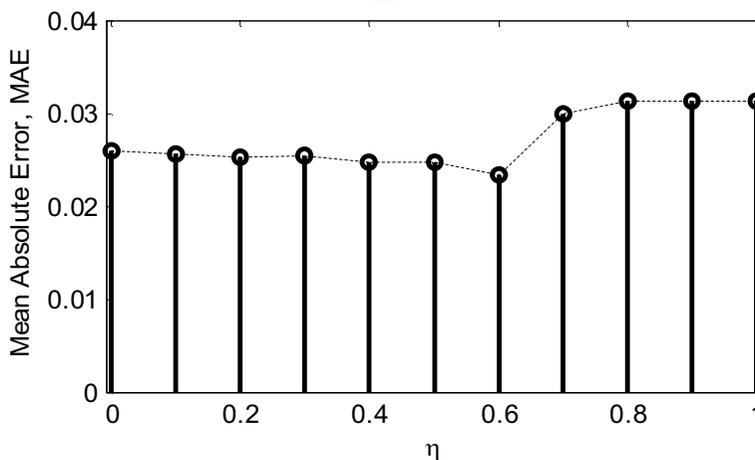


Figure 5.1 η versus *MAE*

The experimental result indicate that, although 100 [%] of pixel utilization rate is secured in the conventional estimation methods employing the GS method for an image of $M \times M$ pixel size (M is a geometric number), maintaining the number of steps at a value 1.6 more than n_{GS} is more effective even though the pixel utilization rate may be slightly lower.

5.1.3 Experiment with images of $M \times M$ pixels

The experiment was performed with the fractal figures shown in Table 5.1 to verify the performance of the BC method employing the MGS method. The fifteen images (= Level \times Size) were used for each image to calculate the fractal dimensions, and the mean and the standard deviations of the dimensions were also calculated. Table 5.2 summarizes the theoretical dimensions and the dimensions estimated by the three methods.

Table 5.2 Estimated results of the three methods for the test images

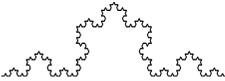
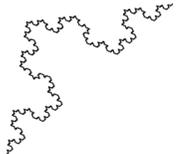
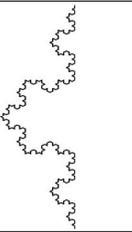
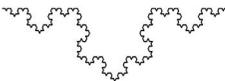
No.	Geometry	Dimension (Standard deviation)			
		Theoretic	BC	TBC	Proposed
1	Koch snowflake (Opened form)	1.262	1.250 (0.019)	1.354 (0.036)	1.267 (0.018)
2	Apollonian gasket	1.328	1.378 (0.015)	1.446 (0.042)	1.369 (0.015)
3	Vicsek fractal	1.465	1.458 (0.031)	1.585 (0.053)	1.465 (0.029)
4	Sierpinski triangle	1.585	1.583 (0.021)	1.623 (0.030)	1.583 (0.018)
5	Rand cantor	1.678	1.664 (0.027)	1.662 (0.025)	1.677 (0.022)
6	Koch curve 85°	1.785	1.792 (0.038)	1.761 (0.039)	1.788 (0.041)
7	Sierpinski Carpet	1.893	1.899 (0.018)	1.896 (0.022)	1.901 (0.016)
8	Hilbert curve	2.000	1.969 (0.062)	1.961 (0.066)	1.974 (0.063)

As can be seen from Table 5.2, the result from the proposed method was generally closer to the theoretical dimensions when compared with the other two methods, and the standard deviation was also smaller. In particular, the severe fluctuation (represented by standard deviation) of the estimates by the TBC method [13] depending on the level indicates that the triangular pattern is not effective.

5.1.4 Experiments on rotated $M \times M$ image

The next experiment was performed to evaluate the rigidity of the proposed method in the case where a fractal figure has been rotated with the Koch snowflake in Table 5.1 having a size of 256×256 (Levels 4 to 8). The figures rotated counterclockwise on the left-end tip by 45° , 90° , and 180° were drawn for each level. Table 5.3 summarizes Koch snowflake images according to different angle of them and the average and standard deviation of estimates obtained targeting 5 images (4-8 levels) used for each rotation angle.

Table 5.3 Estimated results of the three methods for curves rotated

Figure	Angle of rotation	Fractal dimension (Standard deviation)		
		BC	TBC	Proposed
	0°	1.227 (0.011)	1.350 (0.010)	1.247 (0.005)
	45°	1.259 (0.024)	1.271 (0.019)	1.248 (0.019)
	90°	1.227 (0.011)	1.350 (0.009)	1.246 (0.004)
	180°	1.227 (0.011)	1.349 (0.010)	1.254 (0.010)
Overall		1.235 (0.020)	1.330 (0.036)	1.249 (0.012)

5.1.5 Experiment with images of $M \times N$ pixels

Next, the experiment to estimate a fractal dimension by taking $M \times N$ sub-image from $M \times M$ image discussed in Table 5.1 was carried out. For the experiment, three scenarios were used when taking $M \times N$ sub-image with the previously obtained 256×256 image as the original.

In Scenario A, M is fixed to 128 and $M \times N$ pixels is taken based on the top left side while increasing N with an increment of 2 from 128 to 256, and in Scenario B, M is fixed to 191 and, in Scenario C, M is fixed to 256 and N is changed using the same method. Figure 5.2 shows this process schematically.

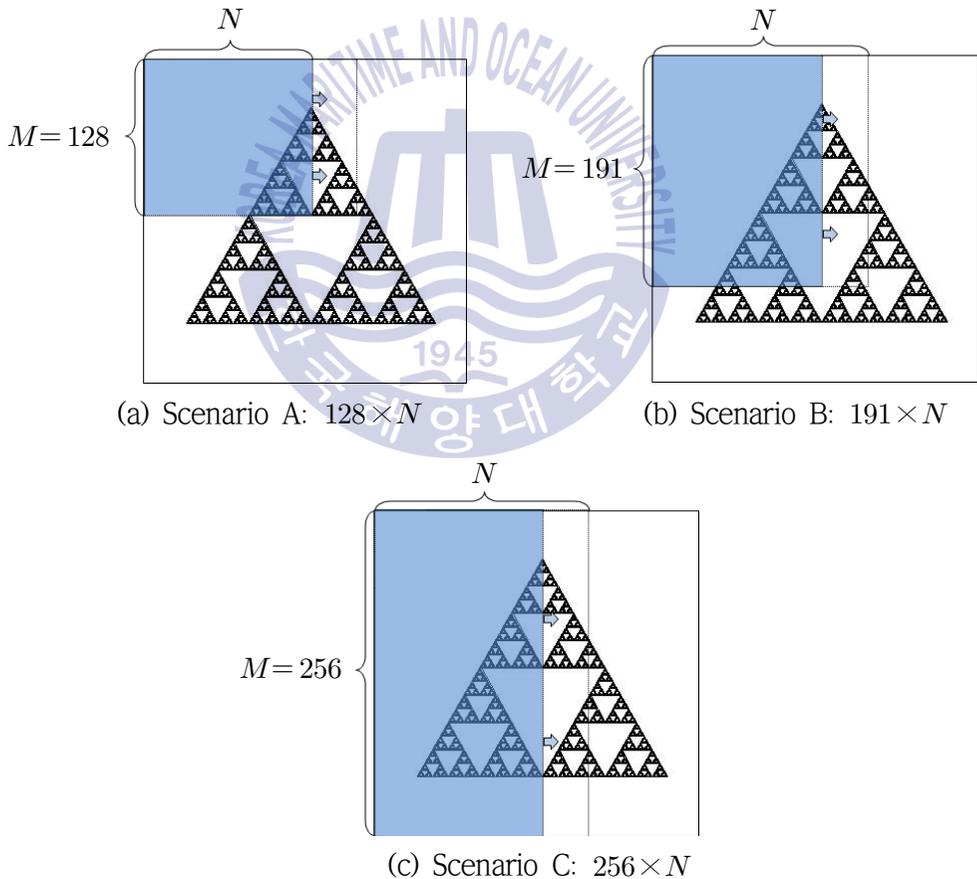
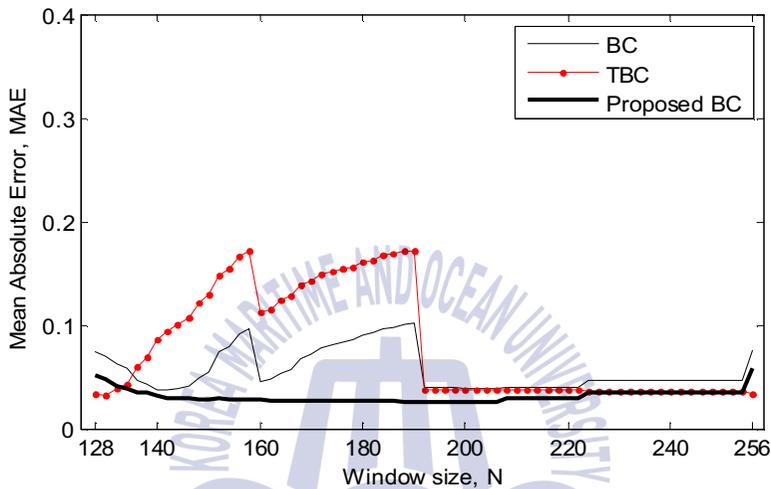


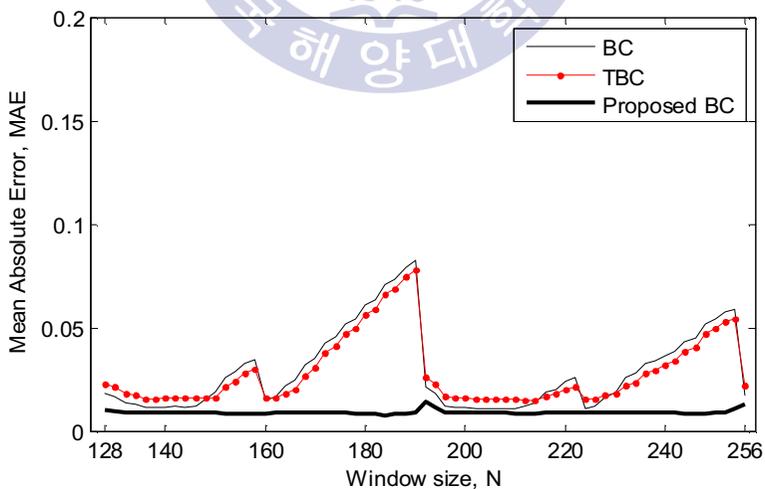
Figure 5.2 $M \times N$ sub-image from $M \times M$ image

An experiment according to the scenario A was performed with Sierpinski triangle (Levels 5 to 9) and Sierpinski carpet (Levels 3 to 7). M was fixed at a constant value, as $M=128$ for the Sierpinski triangle and the Sierpinski carpet.

The MAE was calculated while N was changed to start with sub-images of the sizes of $1/4$. Figure 5.3 show the process.



(a) Sierpinski triangle



(b) Sierpinski carpet

Figure 5.3 MAE of the three methods for $128 \times N$ images

As shown in Figure 5.2(a), in case of Sierpinski triangle, it was a difficult environment for the experiment since only white background and 1/2 size triangle are included in the initial sub-image, but in Figure 5.3(a), N increases and MAE decreases until the step where a large triangle (top figure among three figures) is completed, and then white background is added and MAE increases slightly in case of the suggested method. Generally, MAE is 0.07 or less and it brings a good performance without being sensitive to the size of N . On the contrary, the remaining two methods do not bring a good performance for a partially fractal figure while being sensitive to the size of N . In case of Sierpinski carpet in Figure 5.3(b), the suggested method also brings a better performance while not being sensitive to the size of N in comparison with the remaining two methods.

An interesting fact is that the estimated values of two methods change similarly to the pattern of increase and decrease of the average pixel waste ratio (deleted pixel number/total pixel number) of the GS sampling method (Refer to Figure 5.4). In other words, if the pixel waste ratio is lower, the performance becomes better, and if the pixel waste ratio is higher, the performance becomes worse.

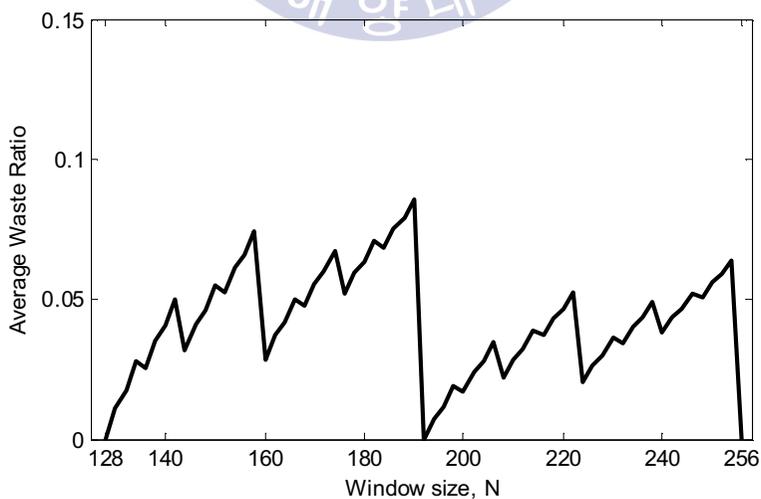
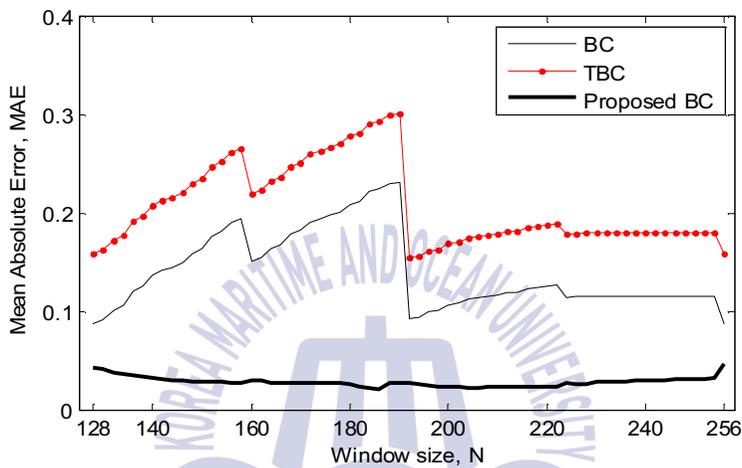
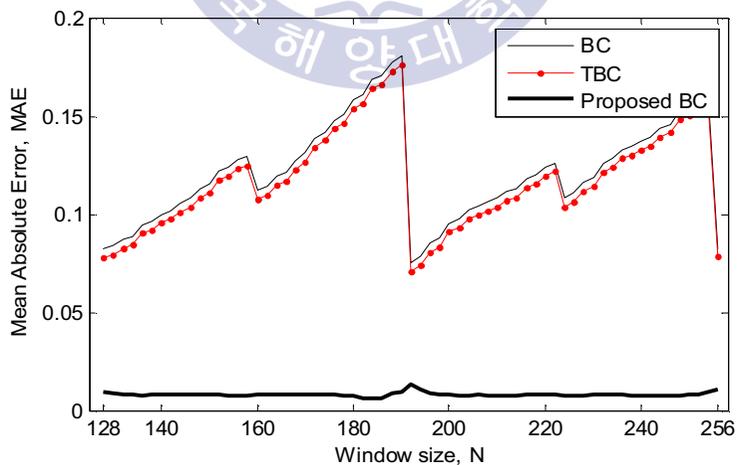


Figure 5.4 Average pixel waste ratio of the GS sampling method

The experiment based on the scenario B was also performed with the same fractal images. Figure 5.5 were drawn by calculating MAE , while fixing M at 191 (high pixel waste rate in GS method), increasing N from 128 to 256 by 2 at each time to obtain $M \times N$ pixels, starting from the top left corner. The Figures show that the performance of the two methods is sensitive to the size of N , whereas the proposed method is less sensitive to the size of N and more precise.



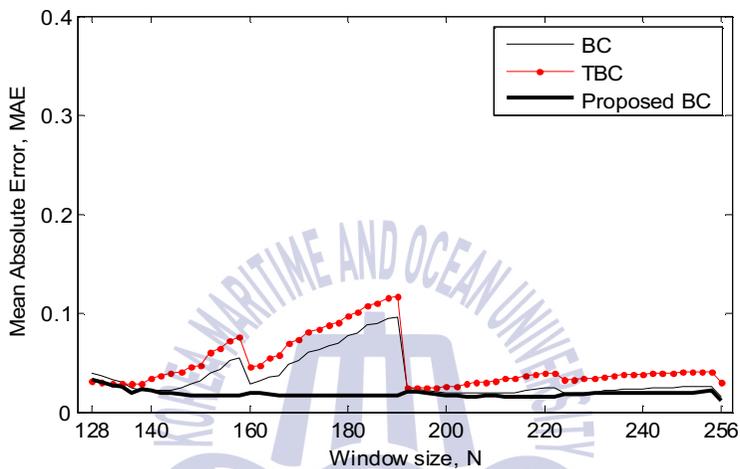
(a) Sierpinski triangle



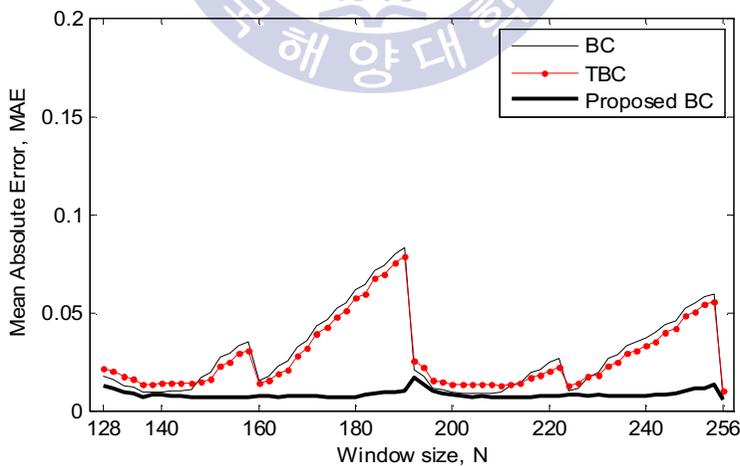
(b) Sierpinski carpet

Figure 5.5 MAE of the three methods for $191 \times N$ images

The experiment based on the scenario C was also performed with the same two fractal images used in the previous experiment. *MAE* was obtained by fixing $M=256$ and changing N to begin with the initial sub-image of $1/2$ size of the original image and Figure 5.6 shows this process schematically. As can be known from the results of the experiment shown in Figure 5.6, the overall performance of the proposed method was good.



(a) Sierpinski triangle



(b) Sierpinski carpet

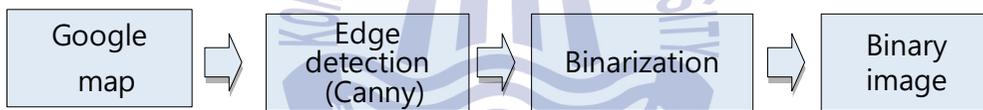
Figure 5.6 MAE of the three methods for $256 \times N$ images

5.2 Experiments on non-deterministic fractal image

Three methods were applied to the experiment to attain a non-deterministic figure where a theoretical fractal dimension cannot be known in advance. Since it is well known that the value of theoretical fractal dimension is unattainable in experiments performed with coastlines having non-deterministic fractal structures, the performance of individual methods may not be directly compared by calculating *MAE* as in the previous experiments. Nonetheless, the relative complexity of the images may be intuitively evaluated. Therefore the results of the fractal dimension estimation were calculated by the three methods.

5.2.1 Converting color images to binary images

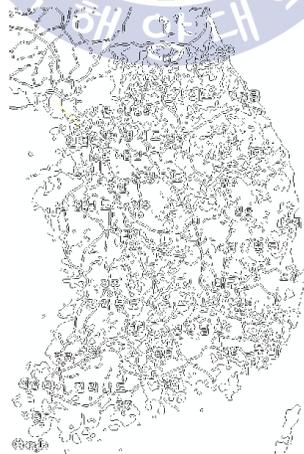
The scaled RGB images for coastlines of the Military Demarcation Line were obtained from the Google Map. Figure 5.7(a) shows such processes and Figure 5.7(b) displays such processes in an image.



(a) Block diagram of binarization of coastline images



(b) Google map



(c) Edge detection (Canny)



(d) Binarization

Figure 5.7 Binarization of coastline images

The coastlines were detected and converted into gray level images using the Canny algorithm that is provided by program MATLAB, and those images were converted again into binary (black and white) images using the drawing board.

5.2.2 Coastline images

In order to obtain the binary images of the Eastern, Southern and Western coastlines, RGB images to a scale of approximately 1:250,000 from the Google map were obtained and these images were converted through the step shown in Figure 5.7. The images to a scale of 1:250,000 could not be obtained from one monitor screen at a time, so the images were obtained separately as shown in Figure 5.8. At this time, the Eastern coast was divided into eight sections, the Southern coast was divided into six sections and the Western coast was divided into nine sections. Figure 5.8 shows the map of south of the Korean Peninsula and the position of acquired coast images, and Appendix A-C show the converted binary images.

The classification of sea area between the Eastern, Southern and Western Seas is a straight line between the Taehwa River Estuary in Ulsan which is the boundary line between the Southern and Eastern Seas and Izumo in Shimane prefecture, Japan according to the announcement of Korean Ministry of Oceans and Fisheries in 1997 and the boundary between the Western and Southern Seas is the overlapping point connected with a straight line between the westernmost tip of Jindo-gun and the westernmost tip of Chagwido near Jeju Island. When the images from Google map were binarized by computer program, tiny islands were eliminated since they were considered irrelevant to fractal estimation. The large islands in the south coast were considered as being geographically connected for convenience's sake, despite being connected by bridges. In particular, the mouths of river were considered as smoothly connected [22].

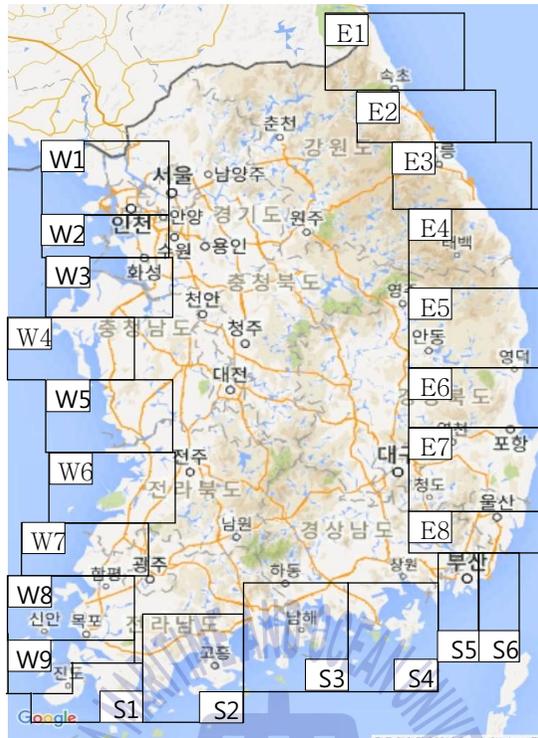


Figure 5.8 Korean coastline maps for experiment

Table 5.4-5.6 show the results of the fractal dimension estimation by the three methods.

Table 5.4 Estimated fractal dimensions of the Eastern coastlines

Division	Method		
	BC	TBC	Proposed
E1	1.078	1.047	1.065
E2	1.050	1.068	1.041
E3	1.029	1.089	0.984
E4	1.012	1.049	1.003
E5	1.092	1.050	1.062
E6	1.100	1.046	1.072
E7	1.193	1.245	1.101
E8	1.261	1.230	1.234
Average (Standard deviation)	1.102 (0.079)	1.103 (0.079)	1.070 (0.071)

Table 5.5 Estimated fractal dimensions of the Southern coastlines

Division	Method		
	BC	TBC	Proposed
S1	1.338	1.302	1.282
S2	1.301	1.299	1.229
S3	1.357	1.325	1.312
S4	1.421	1.387	1.386
S5	1.278	1.274	1.257
S6	1.209	1.202	1.186
Average (Standard deviation)	1.317 (0.066)	1.298 (0.056)	1.275 (0.063)

Table 5.6 Estimated fractal dimensions of the Western coastlines

Division	Method		
	BC	TBC	Proposed
W1	1.246	1.249	1.229
W2	1.203	1.191	1.199
W3	1.287	1.292	1.256
W4	1.238	1.238	1.199
W5	1.225	1.200	1.190
W6	1.184	1.200	1.127
W7	1.099	1.140	1.109
W8	1.372	1.371	1.313
W9	1.389	1.368	1.358
Average (Standard deviation)	1.249 (0.085)	1.250 (0.075)	1.220 (0.076)

Table 5.6 shows that the results of the three methods are similar.

Since it was possible to assume intuitively that the Western and Southern coastlines were more complicated than the Eastern coastline, a relatively larger

fractal dimension was shown through the experiment. A larger fractal dimension means that the structure is more complicated. According to the result of this method, the fractal dimension of coastlines in the Korean Peninsula is considered to be approximately in between 1.03 and 1.28 [23]. This result is also similar to the result that the dimension of Western coastline of Britain is $D=1.25$, and the dimension of borderline between Spain and Portugal is $D=1.14$.



Chapter 6. Conclusion

The BC method which adopts the GS sampling is mainly used for $M \times M$ images, and this method brought an efficient result for the estimation of fractal dimension when M had a series of 2. However, not all images have $M \times M$ size only. If $M \times N$ image is given and the previous BC method is used, more pixels will be deleted, deteriorating the performance of fractal dimension estimation.

In order to solve such problems, a new BC method which expanded the previous integer box counting method and improved the reliability of fractal dimension by counting the boxes with a real number and obtaining a fraction of $N(\delta)$ was suggested. As a result of applying the improved BC method to two deterministic fractal figures where a theoretical fractal dimension was known, comparing and evaluating the fractal estimation performance between the previous BC method and the TBC method suggested by Kaewaramsri, it was confirmed that it worked well also for a partial figure and it brought a better performance without being sensitive to the image size. Also, an excellent result was obtained when an experiment was carried out using Koch snowflake where a fractal dimension was known in order to see its robustness when the image was rotated.

As a result of estimating the fractal dimension in order to improve the complexity of coastlines of the Korean Peninsula that is non-deterministic fractal, the fractal dimension was confirmed to be in between 1.03 and 1.28.



References

- [1] B. Mandelbrot, "How long is the coast of Britain? Statistical self-similarity and fractional dimension", *Science*, Vol. 156, No. 3775, pp. 636-638, 1967.
- [2] A. Pentland, "Fractal-based description of natural scenes", *IEEE transaction on pattern analysis and machine intelligence*, Vol. 6, No. 6, pp. 661-674, 1984.
- [3] J. Li, Q. Du, and C. Sun, "An improved box-counting method for image fractal dimension estimation", *Pattern recognition*, Vol. 42, No. 11, pp. 2460-2469, 2009.
- [4] K. C. Clarke, "Computation of the fractal dimension of topographic surfaces using the triangular prism surface area method", *Computers and geosciences*, Vol. 12, No. 5, pp. 713-722, 1986.
- [5] X. C. Jin, S. H. Ong, and Jayasooriah, "A practical method for estimating fractal dimension", *Pattern recognition letters*, Vol. 16, No. 5, pp. 457-464, 1995.
- [6] L. Yu, D. Zhang, K. Wang, and W. Yang, "Coarse iris classification using box-counting to estimate fractal dimensions", *Pattern recognition*, Vol. 38, No. 11, pp. 1791-1798, 2005.
- [7] K. Shyu, Y. T. Wu, T. R. Chen, H. Y. Chen, H. H. Hu, and W. Y. Guo, "Measuring complexity of fetal cortical surface from MR Images using 3-D modified box-counting method", *IEEE transaction on instrumentation and measurement*, Vol. 60, No. 2, pp. 522-531, 2011.
- [8] M. Lin, L. Chen, and Y. Ma, "Research on stream flow series fractal dimension analysis and its relationship with soil erosion", *IEEE international symposium on geoscience and remote sensing- IGARSS*, Australia, pp. 1821-1823, 2013.

- [9] F. Barakou, D. Koukoula, N. Hatzargyriou, and A. Dimeas, "Fractal geometry for distribution grid topologies", *IEEE power tech 2015 conference*, Netherlands, pp. 1-6, 2015.
- [10] K. Foroutan-pour, P. Dutilleul, and D. L. Smith, "Advances in the implementation of the box-counting method of fractal dimension estimation", *Applied mathematics and computation*, Vol. 105, No. 2-3, pp. 195-210, 1999.
- [11] A. K. Bisoi and J. Mishra, "On calculation of fractal dimension of images", *Pattern recognition letters*, Vol. 22, No. 6-7, pp. 631-637, 2001.
- [12] N. T. Miloevic and G. N. Elston, "Box-count analysis of two dimensional images: methodology, analysis and classification," *19th international conference on control systems and computer science*, Bucharest, pp. 306-312, 2013.
- [13] Y. Kaewaramsri and K. Woraratpanya, "Improved triangle box-counting method for fractal dimension estimation", *Recent advances in information and communication technology 2015*, H. Unger, P. Meesad and S. Boonkrong (Eds), pp. 53-61, 2015.
- [14] X. Zhu and J. Wang, "On fractal mechanism of coastline-A case study of China", *Chinese geographical science*, Vol. 12, No. 2, pp. 142-145, 2002.
- [15] K. W. Kwon and J. D. Lee, "A study on the terrain interpolation using fractal method", *Korean society of civil engineers*, Vol. 26, No. 5, pp. 895-907, 2006.
- [16] H. J. Park, "How mathematics tries to deal with problems in an imperfect world?", bookhouse publishers, 2015.
- [17] D. Saupe, "Algorithms for random fractals", *The Science of Fractal Images*, In H. -O. Peitgen and D. Saupe, Editors, Springer-Verlag, 1988.

- [18] E. Fuchs, “*Arithmetic properties of Apollonian circle packings*”, Doctoral dissertation, Princeton University, 2010.
- [19] Z. Liu and L. Guo, “Connectivity and synchronization of Vicsek model”, *Science in China series F: Information sciences*, Vol. 51, No. 7, pp. 848-858, 2008.
- [20] K. J. Vinoy, K. A. Jose, V. K. Varadan, and V. V. Varadan, “Hilbert curve fractal antenna: A small resonant antenna for VHF/UHF applications”, *Microwave and optical technology letter*, Vol. 29, No. 4, pp. 215-219, 2001.
- [21] J. J. Gagnepain and C. Roques-Carmes, “Fractal approach to two-dimensional and three-dimensional surface roughness”, *Wear*, Vol. 109, pp. 119-126, 1986.
- [22] I. P. Hong and J. W. Ko, “The fractal characteristics of river basin”, *Korean society of civil engineers proceedings*, Vol. 3, No. 3, pp. 281-284, 1998.
- [23] H. S. Woo, K. S. Kwon, and B. U. Kim, “A study on the length characteristic of Korea coastlines with fractal dimensions”, *Korean society of civil engineers proceedings*, pp. 623-624, 2014.

Appendix A. Eastern Coastline Images



Figure. A.1 E1 image of 580×767

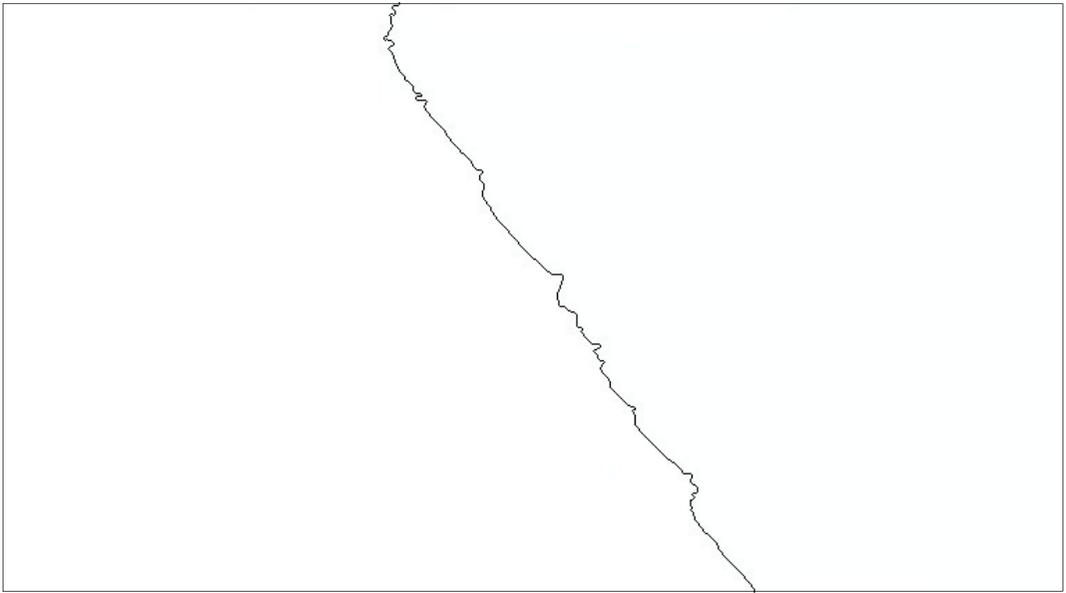


Figure A.2 E2 image of 424×767



Figure A.3 E3 image of 507×767



Figure A.4 E4 image of 557×767





Figure A.5 E5 image of 596×767

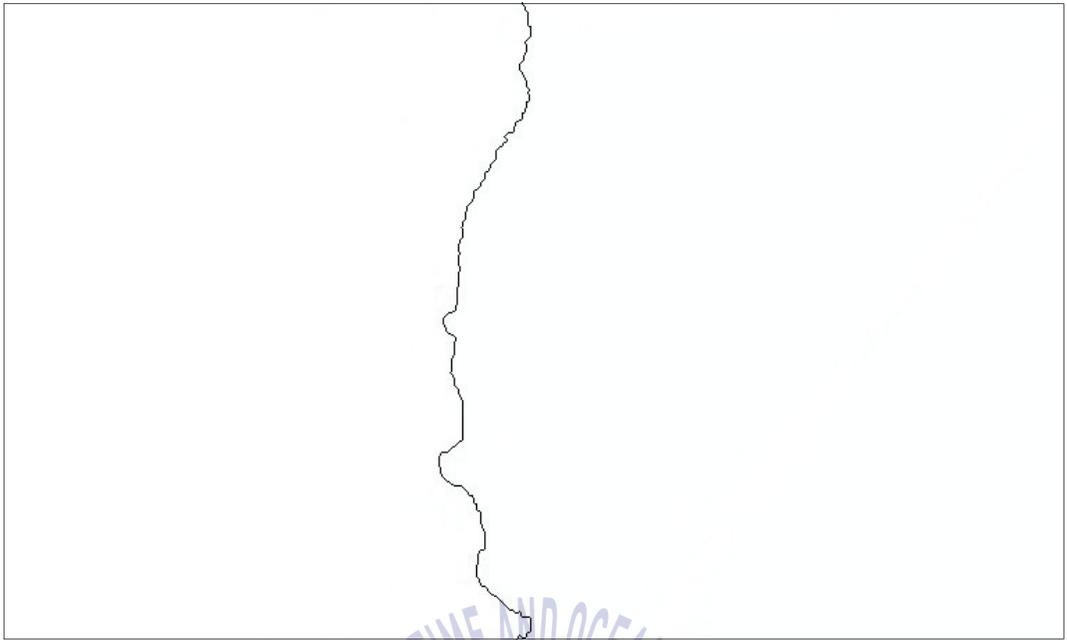


Figure A.6 E6 image of 458×767

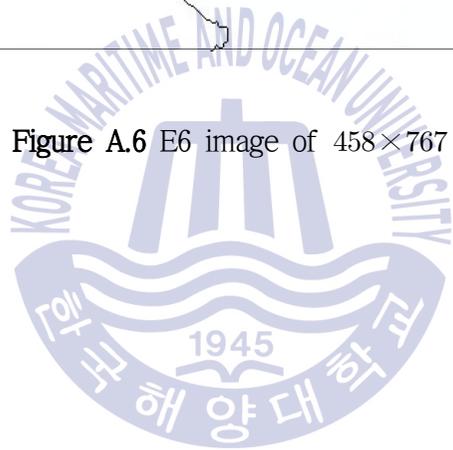




Figure A.7 E7 image of 581×767

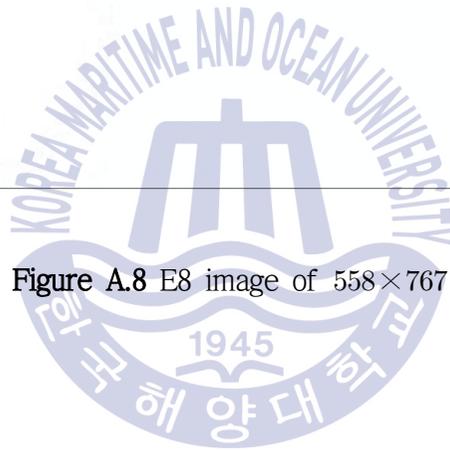
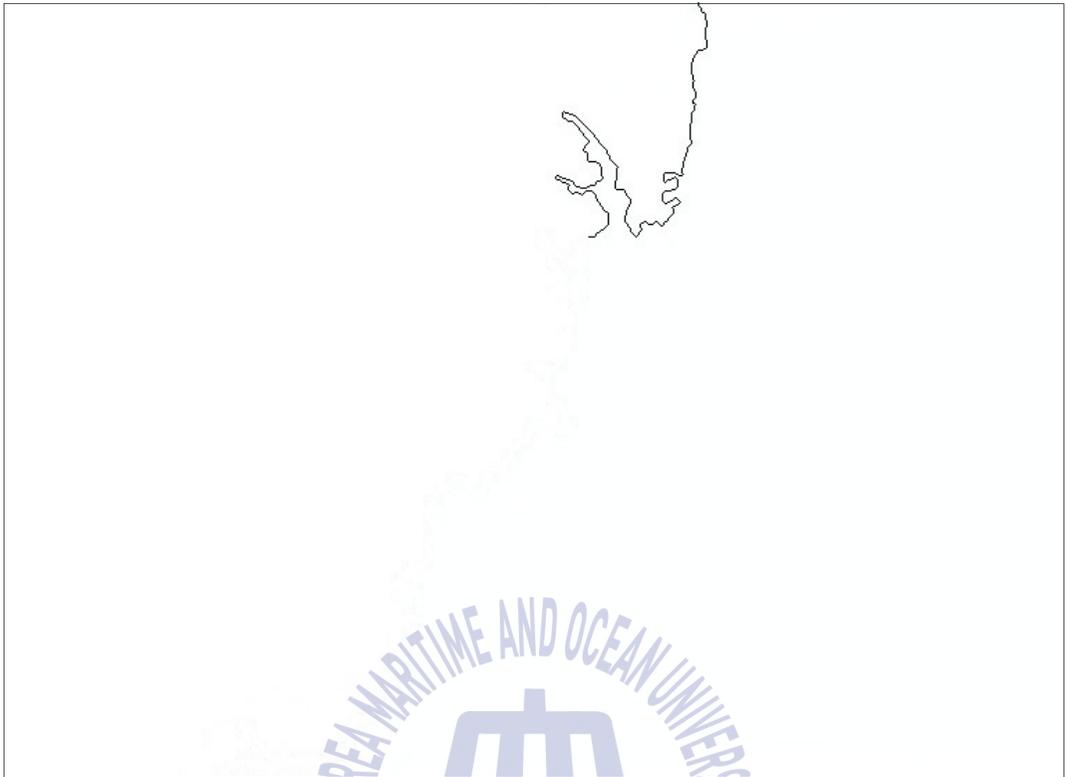


Figure A.8 E8 image of 558×767

Appendix B. Southern Coastline Images



Figure B.1 S1 image of 620×729



Figure B.2 S2 image of 620×711



Figure B.3 S3 image of 620×694

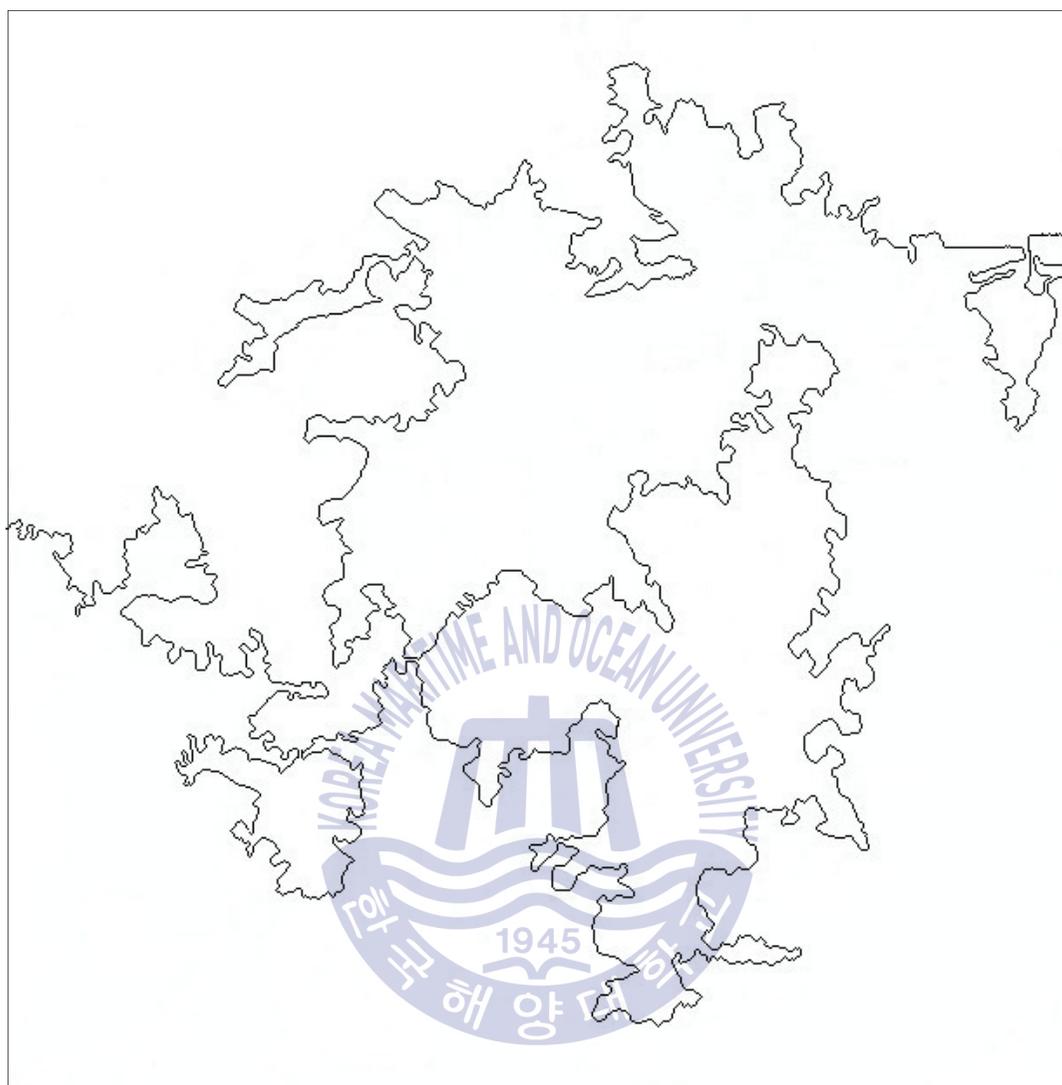


Figure B.4 S4 image of 620×612

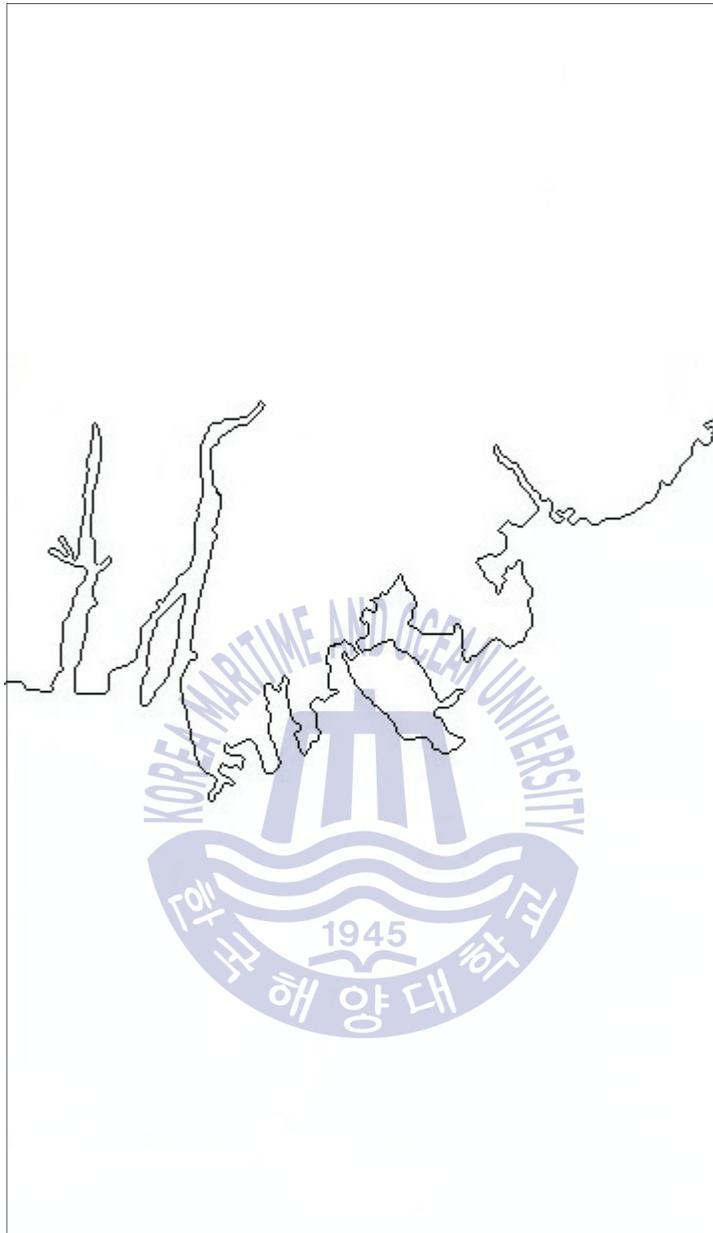


Figure B.5 S5 image of 620×360



Figure B.6 S6 image of 620×360

Appendix C. Western Coastline Images



Figure C.1 W1 image of 583×951



Figure C.2 W2 image of 271×951

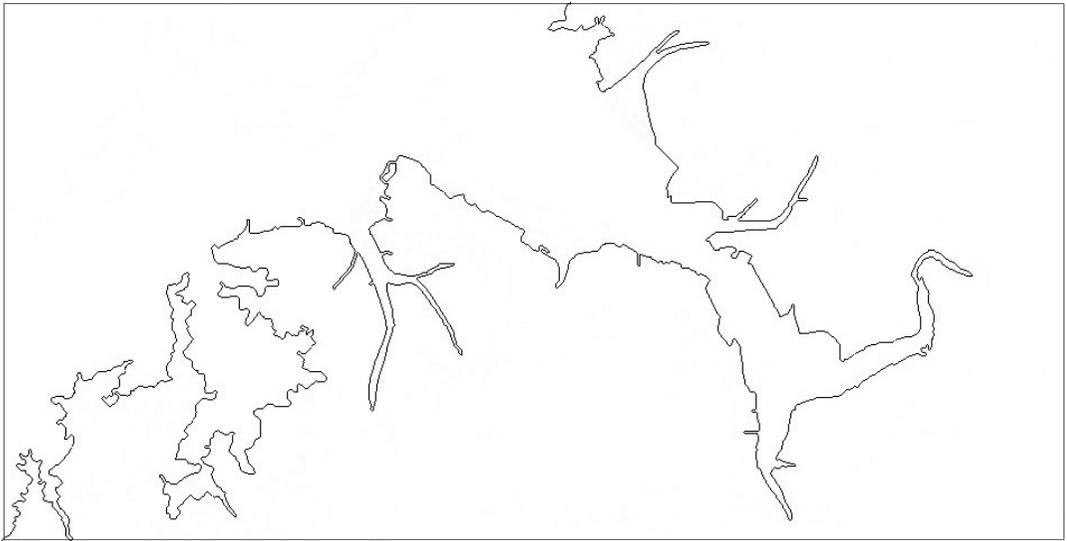


Figure C.3 W3 image of 480×951



Figure C.4 W4 image of 481×951

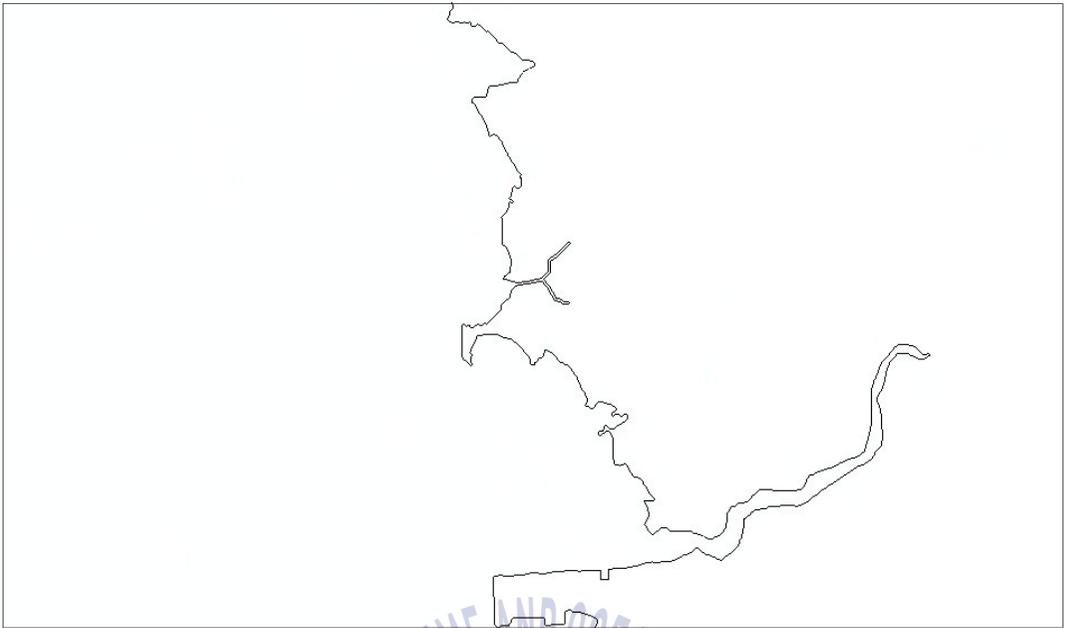


Figure C.5 W5 image of 558×951



Figure C.6 W6 image of 508×951

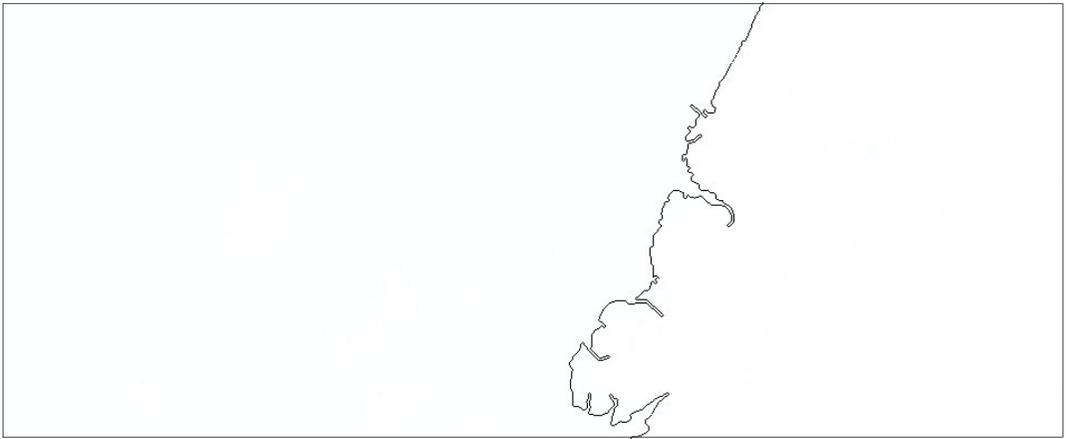


Figure C.7 W7 image of 389×951

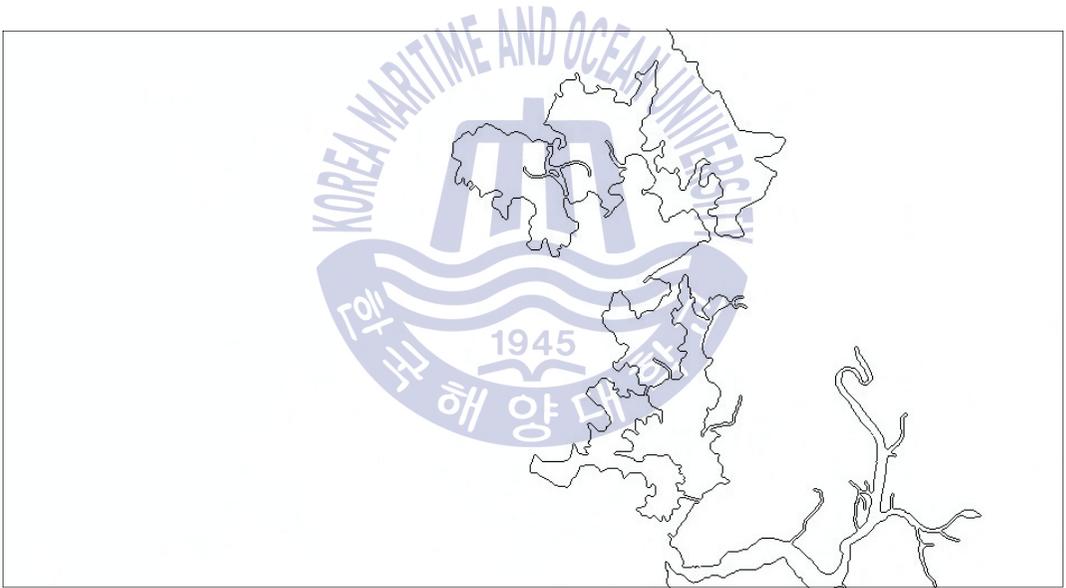


Figure C.8 W8 image of 498×951



Figure C.9 W9 image of 555×951

