We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

10

The Application of Fractal Concept to Content-Based Image Retrieval

An-Zen SHIH
Jin-Wen University of Science and Technology
Taiwan

1. Introduction

The saying 'a picture is worth a thousand words' conveys the idea that images transmit information more efficiently than words can. Furthermore, it is increasingly true that images are an important feature of our daily lives. Because of progress in computer technology, people tend to rely on computers to handle this image data. However, significant problem is that computers are far less adept than people are interpreting this data.

The problem lies in the difficulty that computers have in understanding the image data. This vision process is something most humans take for granted – computers however need to be programmed both to progress the image data and to extract an understanding. Vision implies not only seeing but also understanding the contents of an image. For a computer, an image is just an array of numbers. When people look at an image, they see beyond the array of numbers to the semantics of the data, the visual content of the image. This is why people can easily distinguish different pictures but computers can't.

Nevertheless, a human's eyes can be deceived. In addition, it is tedious for people to look at lots of images, or to concentrate on performing tasks based on image content. Therefore, programming computers to distinguish different images or to select interesting image from a welter of them become an important task.

To date, many image processing methods have been proposed and tested to solve the problem of reading and selecting image data. Fractal, which is a new concept found in last century, suggest a way around the problem. Fractals are patterns that exhibit self similarity. Self similarity means the fractal patterns can be subdivided recursively into smaller non-overlapping parts and each part is a small replica of the whole. Many natural structures exhibit fractal characteristics. By extension fractal patterns will also appear in images. These fractal patterns interested people because that they could be used as a powerful tool in image retrieval.

The application of fractal concept to content-based image retrieval is classified into two groups: fractal dimension approach and fractal compression approach. In this chapter we will describe the idea how to use fractal compression technique in image retrieval.

This chapter is arranged as following. In the next section we will introduce the concept of fractal and fractal compression. And then a introduction section of content-based image retrieval is followed. After that a whole section will be used to describe the idea how to use fractal compression technique in image retrieval. Finally, a brief summary can be found in the end of this chapter.

2. Fractal Concept

2.1 Self Similarity and Self Affinity

Mandelbrot invented the word 'fractal' and described it as[1]

'Mathematical and natural fractals are shapes whose roughness and fragmentation neither tend to vanish, nor fluctuate up and down, but remain essentially unchanged as one zooms in continually and examination is refined.'

From the above words, we notice that fractal presents a strong similarity in its shape. This similarity is defined as self-similarity.[2] In other words, the meaning of self-similarity is that each part is a linear geometric reduction of the whole, with the same reduction ratios in all directions.

Self-similarity can be deterministic self-similarity or statistical self-similarity. Deterministic self-similarity means the self-similarity object can be derived into non-overlapping parts and each non-overlapping part is exactly similar to the whole. A good example is the Sierpinski triangle.(see figure 1) Statistical self-similarity means that the self-similarity object can be derived into non-over-lapping parts and each non-overlapping part only looks similar to the whole but is not exactly similar. The difference is that there is a small difference between the divided part and the entire patter. However, the difference is so small we can usually ignore it. A good example is a cauliflower head. Each part of a cauliflower head is "statistically" look like the whole cauliflower head with a little distortion.

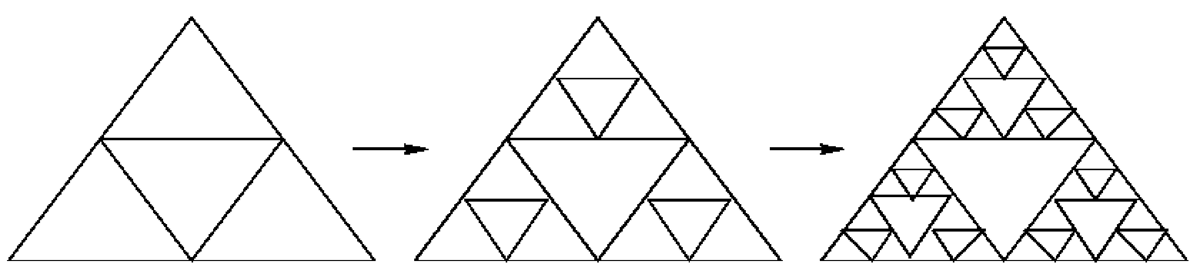


Fig. 1. Sierpinski Triangle – an example of self-similarity

There is also another form of similarity called self-affinity. A self-affinity object is that different ratio will be applied to different direction.. That is, different ratio is applied in different direction continuously to generate a pattern. In Figure 1, if we reduce the Sierpinski triangle one third in horizontal direction but by one half in the vertical direction, then we have a Sierpinski triangle of self affinity.

2.2 Iterated Function System

Iterated Function System (IFS) were first described by Michael Barnsley in 1985 as a tool to create deterministic fractals. The advantage of IFSs is that IFS itself is very straightforward in form but it is capable of generating complex functions.

The IFS code can be represented in matrix form as follows. Assume that a coordinate system in the place is given and that all the maps ω_i are affine. Where a transformation ω is affine if it may be represented by a matrix M and translation t as ω (x) =Mx+t, or

$$\omega \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \tag{1}$$

For application to an image, the above matrix needs to change to

$$W \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e \\ f \\ o \end{bmatrix} \tag{2}$$

Here z is the grey level of the image. The parameter a, b, c, d e and f control the spatial transformation, s controls contrast and o controls brightness.

Not only fractal pattern can be constructed by an IFS system, but also that this procedure can be inverted. That is, for a specific fractal images, we can determine the IFS and store that instead of the whole image.

Fractal image compression is based on IFS theory. It was based on affine transformations acting locally rather than globally. An image to be encoded is partitioned into non-overlapping range blocks R. The task of a fractal encode is to find a large block of the same image (a domain block D) for every range block such that a transformation of the block W(D) is a good approximation of the range block. This transformation can be written as Equation 2 which is a combination of geometrical transformation and luminance transformation.

In the decoding period, the transformation of each range block needs to be transmitted to the decoder. This code, when applied to the initial image, will generate a simulation of the original image.

There are two different kinds of IFS compression. The first is IFS compression and the other is PIFS compression. The PIFS compression is also called local iterated function system. Because the image is usually not perfectly fractal, it is hard to find a global self-similarity or self-affinity spreading the whole image. Instead, we should try to find parts of the image which are self-similar.

2.3 The PIFS Compression Method.

We will begin our description of PIFS with a simple illustration. An image of size 32 * 32pixels is divided into 16 non-overlapping sub-images called ***ranges*** with size 8 *8 pixels. Next a window of size 16 *16 pixels is moved line by line horizontally then vertically in the same image to choose a sub-image called the ***domain***. The total numbers of domains in the image will be 16 *16.

For each range we will find a domain which contains the same pattern inside the range. We use a window to select the domain and to execute the IFS transformation shown in equation

(2) on the domain. Then we can compare the transformed domain and the range. After searching for all the possible domains in the image, we can possibly find the domain that is considered to have same pattern on the range. If we cannot find the domain we want, then we divide the image into smaller sub-image 8 *8 and search the image again. This procedure continues until every range on the image has found a domain with the same pattern.
As a real image is composed of many fractal patterns, it is natural way of thinking to divide the image according to the different fractal patterns and then try to find different IFS code for each of then. However, it is very difficult to implement this in an automatic algorithm.
There are many ways to partition the image. The most popular method is quadtree partitioning method. The essential idea of quadtree partitioning is that the image is broken up into four equal-sized sub-squares. Each sub-image can be broken up into a further four equal-sized sub-images if necessary. Finally the image is composed of different sized squares.
It should be pointed out that the procedure of dividing the image can't repeat beyond a specific bound. Before we execute the quadtree partitioning, we should set up the smallest size of the square. The division will stop if the square size is smaller than the above pre-defined size.
The implementation of PIFS is very similar to the above simple illustrated sample. Firstly, the image is divided into 16 ranges by using the quadtree portioning method. Then for each range we will find a domain which will look like the range after transformation. If a range can't be mapped onto a suitable domain, then this range is divided into four parts using quadtree partitioning. For each sub-image we again try to find a domain containing the same pattern as the range. If a sub-image still can't find a good domain then this sub-image has to be divided into four parts yet again. The process continues until every part of the image finds a domain somewhere in the same image.
To sum up, we can describe PIFS algorithm as below. An image can be viewed as a three dimension function. That is, the grey level represented as $z = f(x, y)$ where x and y are the locations of the pixels in the image. The total image then is represented as $F = (x, y, f(x, y))$.
The image can tiled into subsets, $D_1, D_2, \dots, D_n$ and $R_1, R_2, \dots, R_n$ where D_n and R_n are called domains and ranges respectively. On each range and domain, it is possible to find a set of mapping, $\omega_1, \omega_2, \dots, \omega_n$, which will transform ranges to domains. The transformations, $\omega_1, \omega_2, \dots, \omega_n$, compose The PIFS system of the image. In other words, we have transformed the image into several matrix$\omega_1, \omega_2, \dots, \omega_n$.

3. Content-Based Image Retrieval Concept

Content-based image retrieval (CBIR) is any technology which can help us to organize digital picture archives by their visual content. The term "content-based" implies that computers have to analyze the content of an image during the search and the term "content" refers to colors, shapes, textures, or any other information which can be extracted from the image. By this definition, anything ranging from an image similarity function to a robust image annotation engine falls under the purview of CBIR. People from different fields, such as, computer vision, machine learning, information retrieval, human-computer interaction, database systems, Web and data mining, information theory, statistics, and psychology contributing and becoming part of the CBIR community.[3]
Two problems exist in CBIR. The first is how to mathematically describe an image. The

second is how to access the similarity between images based on description. The first problem lies in the difficulty of .computers have in understanding the image data. When an image is presented, people can usually see beyond the shapes and colors on the image to the real content of that image. However, computers can't understand the content of the image if we don't program the computers. This is because an image data is just an array of numbers for the computers. So we hope to find a mathematic description of the image, which is sometimes called signature, in order that computers can understand the semantic meaning of an image. After we find the mathematic description of an image, computers can possibly use the signature to compare different images and select interesting ones.

The general algorithm of CIBR begins with extraction of features such as color, texture, shapes, etc. Then these features are analysis to acquire mathematical descriptions of the image. Because it is often impossible that an image is composed with only one pattern, people begin to shift from finding a global feature representation of image, such as color histogram or a global shape, to local features. This is helped with image segmentation, which is critical for characterizing shapes within images. In other words, good image segmentation is toward better image understanding. To date, precise segmentation of an image is, however, still remained an open problem.

All methods of mathematical description extraction have their advantage and limitations.

4. Application of Fractal Concept to CIBR

4.1 Background

In 1848, Shannon suggested that digital data set is made up of information and redundancy. Redundancy means duplicated or unnecessary information in the data set. His theory implies a way of data compression.

Data compression is preformed to decrease redundancy for data storage and data communication. This can be done by transforming the raw data into a new representation while the data, or most of the data, is the same and the length of the new representation will be as small as possible. Because of the way in which data compression tries to use a new representation to shorten the data, sometimes the data compression is called coding.

Compression efficiency can be measured in two ways: algorithm complexity and amount of compression. Algorithm complexity means the time required to compress the data. The amount of compression can be calculated by redundancy, average message length, and compression ratio.

Instead of what is suggested by 'compression', it is important to get another point of view of data compression. By preserving the essential data and removing the duplicate and unnecessary data, data compression gives not only an efficient storage method but also a new representation of the data. This new representation, while still having the essential properties of the raw data, can become an index into the original data. This 'index' can be very useful for selecting the data with similar characteristics in large collections of data. Thus we argue that the compression not only solves the problem of storage but also can become a classification method.

An image is a kind of digital data. It is precisely on such grounds that Shannon's compression theory can be applied onto images. So it is possible to compress an image and remove redundancy while preserving the important feature in the image. That is, image compression suggests itself to be a way around the problem of image retrieval. These

important features are clearly captured in the compressed code.
An important property of the compressed code therefore is that it can be used in image recognition because (somehow) represent the 'essence' of the image. This property is often neglected. Since the compression code contains the 'essential' information of an image, it is reasonable to differentiate images by comparing their compressed codes. Thus if two compressed codes are identical or almost identical, then we can say these two images are identical or similar. Alternatively, if two compression codes are different, then the images are different.
With the help of compressed codes, it is possible that computers can be given a limited 'understanding' of the essential difference between images.
However, there are a number of compression methods on the world. Among these methods, fractal compression method suggests itself to be a good algorithm to retrieve images.
As mentioned in Section 2, fractal image compression is based on IFS theory. It can be classified into two main groups: IFS compression and PIFS compression. IFS compression compress the whole image with one IFS function while PIFS divides the image into several parts then compress different parts with different IFS functions. Nevertheless, most real images are not fractal patterns. We may consider them a composition of several fractal patterns. It is obvious that the IFS method is not applicable to real images. Only with PIFS can we hope to compress ordinary images.
The advantage of PIFS compression is that it can use a very simple form to represent the image. This simple form, in turn, can be a classification criterion in selecting images. In addition, because it will occupy less memory space than the original images, it can solve the memory problem and same on CPU time.

4.2 Literature review

Describing and extracting image's feature is always a key question in content-based image retrieval system. An image can be characterized by its fractal codes, and fractal codes can be used as the image's feature to retrieve the images effectively.
Many people have found that fractal coding is a very promising way in image retrieval. Here we just select some of their works to give a global view of application of fractal compression in image retrieval.
In 1995, Cheng et al. used fractal coding to index image content for a digital library. They conducted experiment on some natural images as well as biomedical images. Their method is that they convert all images to fractal codes before adding to the database. After that they compute the measure on the fractal codes and organize then into indices.[4]
Two years later, Mari-Julie et al. proposed a new method, based on fractal transformation, which can quickly pick up a image pattern from 100 images. They convert texture and edge of pattern into fractal code and then use it for searching process.[5]
In 1998, in his PhD thesis, Shih used fractal compression technique to search specific gamma images from nucleus dataset. In addition, he found that a-priori knowledge of the data can help the computer to recognize images more efficient.[6]
Tien, in his master dissertation, used fractal code to select images. He used Fisher's discrimination function to judge whether the selected image is similar to the one wanted or not.[7]
Then in 2008, Fan et al. suggested to use fractal code to select images. In order to shorten the time of selection, they organized the fractal code into several index and used those index to

search the data. They found that this could accelerate the period of image retrieval.[8]
The same year in 2008, Zhang et al. used IFS code for image retrieval on the compression domain. Their experiment result showed that compared with the direct image pixels similar matching strategy, the algorithm using fractal compression code shortens the retrieval times of compression domain greatly and guarantees the retrieval accuracy.
The above works make it clear that the essence of an image can be captured in the compression code and the code is very useful is querying the specific image from images.
4.3 Algorithm of Fractal Coding of Image Retrieval
If we took a closer look at how people use fractal code to select images, we will find there are two ways of selection. The first is that, we have a specific image and we want to pick up same images, or similar images, from the data set. The second is that, with a-priori knowledge, we know there are some images are different with others and we want to pick them up. Nevertheless, we can preprocess the second with part of the image data set and find the specific image as the target image.
The algorithm that using fractal compression codes in image retrieval is as below.

Algorithm
begin
read target image
get PIFS transformation ω_i
set tolerance ε
while not end-of-file
{
  read tested images
  get PIFS transformation ω_i
  compare with target image
  if (comparison value $\leq \varepsilon$)
    image found
  else
    discard
}
end

4.3 Discussion of the problems

In this section, we want to talk some problems might occurred when using fractal compression code in image retrieval.
In section 4.2 we have found that many people have tested the fractal compression method on different images. In section 2 we have mentioned that there are two kinds of fractal compression methods: the IFS compression and the PIFS (partitioned IFS) compression. The IFS method is good for describing classic fractal patterns but not real images. That is why people use PIFS of image retrieval.
Unlike the IFS compression method which explores the self similarity of the entire image, the PIFS compression methods try to explore the self similarity on sub-image. During the compression process PIFS method recursively divides the image and tries to find sub-images containing the same pattern. In this way, the PIFS compression method records self similarity patterns into the set of coefficients of the compression code.

The partitioning method chosen for implementation was important. It can determine whether the compression result is good or bad. When we partition an image, we are looking for two sub-images containing the same pattern. Thus it is important that the sub-image will contain a pattern. A good partitioning method can do this.
The documents listed in the literature review section confirmed that the PIFS method does code the essential information in an image. Different images have been used as tested data and their results indicated that the method can compress and decompress the image without losing key feature of the image.
Another advantage of PIFS method is that, since we are looking for the same pattern in two sub-images, it could help us to find the pattern if we can confine the search area to where the pattern might appear. We don't need detailed a-priori knowledge of where this pattern is, all we need is a possible target area where the pattern might be found. This can be defined arbitrarily by the user.
We can speed the image recognition process by the use of parallel processing. Usually, the problem of PIFS method is that it is time consuming. However, if we use distributed work station for each of the different parts of the same image, we could save a lot of time.
Nevertheless, several weakness of PIFS method is discovered. Firstly, usually there is more or less background noise on real images. This might affect the compression results. That is, a pattern could mix with noise so that it becomes difficult to be detected. De-noising method is a need before we execute PIFS method on the image.
Another weakness is the comparison method. We should choose a proper comparison method to detect the intensity difference between two sub-images. If the difference is small, then they might contain smaller features. However, the difficulty is to set a threshold value on to the computed comparison value to decide how small the intensity can be tolerated. This threshold value has empirically determined. However, the danger is that this empirical value might not be suitable for every kinds of image.
The full PIFS compression code might be very long. Thus to analyze this data and to extract meaningful information is not easy. However, the code set could be modified to classify images.

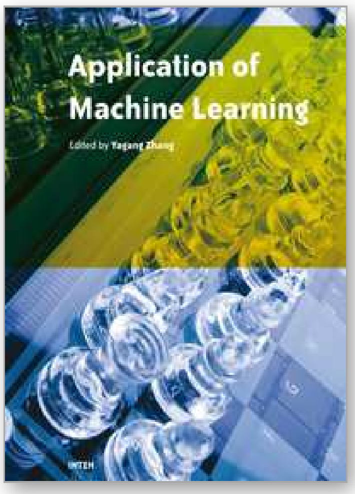
5. Summary

In the beginning of this chapter we noted that images are a key information resource and while processing them in computers is straightforward, understanding them is not so simple.
Image recognition suggests a solution to this problem. To compress an image is to remove the redundancy, i.e. unnecessary information, and transform the essential information into compressed codes. Because the compressed codes contain the essential information of the image, it should be possible to use the compression code to compare and recognize the images themselves.
Among many image recognition methods, fractal compression method proves itself to be a promising method. It provides good compression and arguably the most concise codes for image recognition. This method is based on fractal theory. In section 2 we noted that fractals are patterns which exhibit self similarity, which means fractal patterns can be sub-divided recursively into smaller non-overlapping parts and each part is a smaller replica of the entire pattern. Thus, no matter how complicated a fractal pattern looks, it is actually composed of

the same pattern at a different size.
The principle of fractal compression is to compress an image by exploring the possible fractal patterns in the image. The principle of the IFS approach is that we can use a set of coefficients to describe the self similarity of a fractal pattern. This set of coefficients then can be used to re-constructed fractal patterns. Alternatively, if there is fractal pattern in an image, it should be possible to use IFS coefficients to represent it. Thus an image can be compressed by simply storing the IFS coefficients rather than the original image. The advantage of the IFS approach is it is simple in form but it can represent complicate 3d structure.
These PIFS codes are a very promising tool in image retrieval. We have illustrated some people’s works It shows that fractal is a good tool to be develop in image retrieval. In addition, constrains of using fractal in image retrieval are mentioned and discussed.

6. Reference

[1] D. J. Tildesley M Fleischmann and R.C.Ball. editors. Fractal in the Natural Science
[2] B. Mandelbrot. Fractals-forms, chance and dimension. W. H. Freeman and Company. 1997
[3] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang, Image Retreival :Ideas, Influence, and Trends of New Age, ACM Computing Survey, Vol. 40 No. 2, Article 5, April 2008
[4] Biao Cheng, Aidong Zhang, Raj Acharya, and Claudio Sibata Using Fractal Coding to Index Image Content for a Digital Library, Tachnical Report, University of Buffalo, 1995.
[5] Jean Michel Marie-Julie, Hassane Essafi Image Database Indexing and Retieval Using the Fractal Transform, Multimedia Applications, Services and Techniques –ECMAST ’97, p169-p182, Springer Berlin / Heidelberg, 1997
[6] Tien Fu-Ming, Fractal-based Image Database Retrieval, Master dissertation, Zhong Shan University, Taiwan, 2001
[7] Ce Fan, Dongfa Gao, Xiaorong Wu and Ying Li, Retrieval based on Indexing for Compressed Domain, 2008 International Conference on Computer Science and Software Engineering, p1291-p1294, 2008
[8] Liangbin Zhang, Yi Wang, Lifeng Xi, Kun Gao and Tianyun Hu, New Method of Image Retrieval using Fractal Code on the Compression Domain, WSEAS Transaction on System, Issue 12, Volume 7, Dec. 2008



Application of Machine Learning

Edited by Yagang Zhang

ISBN 978-953-307-035-3
Hard cover, 280 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

The goal of this book is to present the latest applications of machine learning, which mainly include: speech recognition, traffic and fault classification, surface quality prediction in laser machining, network security and bioinformatics, enterprise credit risk evaluation, and so on. This book will be of interest to industrial engineers and scientists as well as academics who wish to pursue machine learning. The book is intended for both graduate and postgraduate students in fields such as computer science, cybernetics, system sciences, engineering, statistics, and social sciences, and as a reference for software professionals and practitioners. The wide scope of the book provides them with a good introduction to many application researches of machine learning, and it is also the source of useful bibliographical information.