Modern Management Science & Engineering
ISSN 2052-2576
Vol. 6, No. 3, 2024
www.scholink.org/ojs/index.php/mmse

2024 International Conference on Science and Technology, Modern Education and Management (TMEM 2024)

A Study on Detailed Image Processing of Fractal Characteristics of Irregular Cloud Edges

Lin Xiao[1]

[1] Shandong Zhaoyuan No.1 Middle School, Yantai, Shandong, China

E-mail: 2295212160@qq.com

Abstract

With the recent development of computer graphics and image processing technology, this paper describes the progress of cloud image generation and processing techniques. Clouds in nature usually have complex and changeable shapes, so accurate and detailed processing of their images has always been an important research topic in computer graphics. First, this paper summarizes the application value of fractal theory in cloud image processing. Next, it presents the fractal generation and edge thinning algorithm as crucial technical components in forming the framework of cloud image detail processing. The fractal generation algorithm uses the fractal geometry principle to simulate the basic shape of the cloud. In contrast, the edge thinning algorithm enhances the realism of the cloud edge by adding small-scale features on the boundary.

Keywords

Fractal geometry, Image processing, Boundary refinement, Cloud simulation

1. Introduction

In recent years, due to the rapid development of computer graphics and image processing technology, cloud image generation and processing technology has made significant progress. Clouds are commonly found in nature, showcasing intricate, ever-changing forms and distinct irregular edges. Hence, precise handling of details in cloud imagery has consistently been a key area of study within computer graphics. This paper will examine and outline the current technology for processing cloud images, focusing on the fractal theory method and its use in enhancing cloud image details.

Fractal geometry is a powerful method for explaining intricate forms found in the natural world (Falconer, 2014). It can understand the natural objects' self-similarity and scale-independent features. Researchers use fractal theory in cloud simulation to recreate the intricate texture and layering of cloud edges, enhancing the realism of images. Based on this, we develop a complete algorithmic framework that consists of two main steps: A fractal generation algorithm is used to create the basic shape of the clouds. An edge-thinning algorithm further enhances the edge details of the clouds, making them look more natural.

2. A Survey of Technology for Processing Cloud Images

In the early days, cloud images were primarily created using the artist's hand-painted or basic geometric construction technique. While this technique can produce simple cloud formations, it does not have authenticity and intricate elements. With the improvement of computing power, researchers have begun to use complex algorithms to generate more realistic cloud images. In particular, fractal geometry is widely used in cloud image generation and processing because of its powerful ability to describe complex structures.

Fractal theory involves the creation of intricate structures using basic mathematical principles. Cloud images frequently exhibit this structure through their irregular boundaries Figure 1. The technology for processing cloud images based on fractal theory includes two key parts: The first one is the fractal generation algorithm, which is used to construct the basic form of clouds; the other is the edge thinning algorithm, which adds detail to the cloud edges to make them look more natural (Zhao & Wu, 2016).

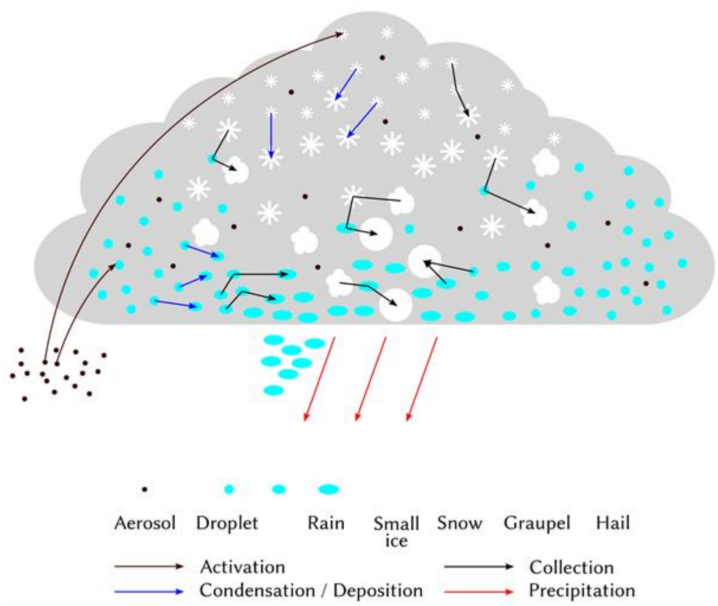


Figure 1. An example of Detailing Cloud Image Processing

3. Description of a Detailed Processing Model for Cloud Images Based on Fractal Theory

3.1 Fractal Generation

In computer graphics, fractal theory has become an effective tool to generate complex natural landscapes because of its unique advantages. Clouds are ideal for fractal modeling due to their irregular borders and intricate formations, representing a common natural scenery. To model clouds, we must initially create their fundamental shape using fractal generation. The core of the fractal generation algorithm is to simulate the natural shape of clouds by using related principles. Fractal geometry is a mathematical branch that examines irregular shapes and intricate structures, particularly useful for analyzing self-similarity in the natural world. For processing cloud images, researchers usually use iterated function system (IFS) as the basic algorithm to generate the basic contours of clouds.

IFS is a fractal generation method based on affine transformation, which can effectively simulate the self-similarity and scale-independent characteristics of clouds. Specifically, IFS consists of a set of affine transformations, each of which can be expressed as $f_i(x)=r_i x+t_i$. In the formula, r_i represents the scaling factor, and t_i is the translation vector. By randomly selecting these transformations and applying them repeatedly to the initial point set, the contour of the cloud can be obtained. In addition, we need to consider the density changes inside the cloud to create a more natural visual effect. This process can be achieved by fractal Brownian motion (fBm). fBm is a continuous Gaussian process with self-similarity and long-range correlation, which can simulate rough surfaces and textures in the natural world.

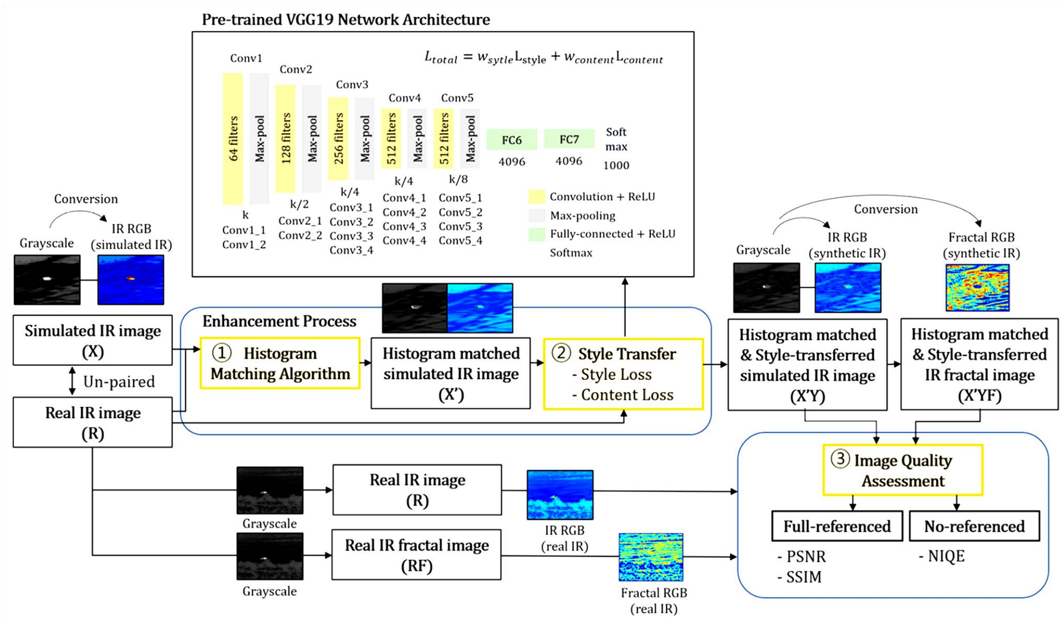


Figure 2. Fractal Generation Algorithm Architecture

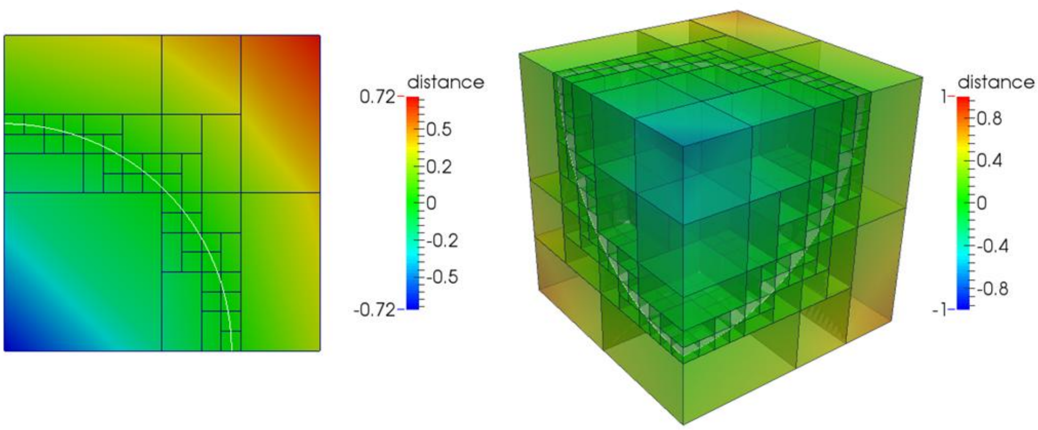
Figure 2 shows an image processing flow based on the pre-trained VGG19 network structure. It combines a fractal generation algorithm to enhance the realism of images (Xian, Wang, & Teng, 2021). First, enter two pictures: a simulated IR image (X) and a real IR image (R). Second, the analog infrared image is converted into a gray image through the conversion process. In the enhancement process, in addition to comparing the histograms between the simulated infrared image and the aligned historical real infrared image, a fractal generation algorithm is also introduced. The fractal generation algorithm utilizes fractal geometry principles to create self-similar intricate structures, enhancing image details and textures for a more realistic appearance. It improves the realism and details of the simulated infrared image.

In this process, we use two different loss functions: Style Loss and Content Loss (Fang, Li, Jia et al., 2023). The loss function captures the content information of the original image and guides online learning on how to imitate the target style. In application, the fractal generation can improve the effect of style transfer so that the output image has the characteristics of the target style and more details. We utilize style transfer technology to convert the improved analog infrared image into an image that mimics the style of a true infrared image from the past. During this phase, the fractal generation algorithm remains active. Adding subtle self-similar patterns to the newly generated image makes the image more textured and realistic. We evaluate the quality of the generated images, including full reference and no reference, and calculate PSNR and SSIM. By combining it with a fractal generation algorithm, it is expected that the visual impact and objective indicators of the generated images will be improved (Setiadi, 2021).

3.2 Edge Thinning Algorithm

Once we have created the basic outline of the cloud, we need to refine the edge details of the cloud to make it look more natural. Edge thinning algorithms are one of the most essential steps in improving the details of a cloud image [6]. Adding small features to the borders makes the edges softer and more layered. The algorithms usually involve local deformation and refinement of existing contours to better simulate the appearance of clouds in nature.

The edge thinning algorithm can be divided into two main parts: local deformation and detail addition. Local deformations mimic the unpredictable nature of the environment through slight random adjustments at the edges of the clouds. For detail addition, we need to add small features like ridges and valleys along the edges to increase complexity.



(**a**) Immersed circle ($R = 0.72$) (**b**) Immersed spherical surface ($R = 0.72$)

Figure 3. Application Examples of Edge Thinning Algorithm

This image shows two mathematical models: an immersed circle (a) and an immersed spherical surface (b). In model (a), there is a circular object with a radius of 0.72, which is immersed in a two-dimensional space. The model's color coding indicates distance, with red for close and green for far distances. The transition from red to green in Figure 3 becomes more apparent as the distance becomes longer.

Model (b) is a three-dimensional sphere whose surface is immersed in a three-dimensional space. The color coding of this model also represents the distance, in which red represents the distance is close, and blue represents the distance is far. From the figure 4, we can see that the color gradually changes from red to blue as the distance increases, and the interior of the sphere is green, signifying the existence of different properties or conditions inside and outside the sphere. In this research, the edge thinning algorithm can highlight the surface boundary of the sphere and ensure the smoothness and consistency of the surface, thereby improving the model's overall quality.

4. Algorithm Evaluation Criteria and Methods

A set of rigorous evaluation criteria and methods are required to ensure the effectiveness and practicality of detail processing algorithms based on cloud images. These evaluation criteria can help quantify the algorithm's performance and guide its optimization and improvement.

Accuracy is one of the important indexes to evaluate the edge thinning algorithm. To quantify accuracy, metrics such as IoU (Intersection over Union) and Dice coefficient can be used (Guindon & Zhang, 2017). These measurements aid in comprehending the intersection level between the polished cloud perimeter and the initial or anticipated perimeter. For example, IoU measures the proportion of overlap between the predicted boundary and the actual boundary by dividing their intersection area by their union area, which is expressed mathematically as:

$$\mathrm{IoU} = \frac{\mathrm{Area}_{\mathrm{intersection}}}{\mathrm{Area}_{\mathrm{union}}} \tag{1}$$

The Dice coefficient is twice the intersection of two sets divided by the sum of their respective areas, and its mathematical expression is:

$$\mathrm{Dice} = \frac{2 \times \mathrm{Area}_{\mathrm{intersection}}}{\mathrm{Area}_1 + \mathrm{Area}_2} \tag{2}$$

$\mathrm{Area}_{\mathrm{intersection}}$ represents the intersection area of the predicted boundary and the actual boundary, Area_1 represents the area of the predicted boundary, and Area_2 represents the area of the actual boundary.

Efficiency is another important evaluation dimension, reflecting the algorithm processing speed and resource consumption. We can evaluate the efficiency by measuring the time required for algorithm execution and memory usage. Usually, researchers will pay attention to the time complexity of the algorithm $O(n)$, where n is the size of the input data. In addition, the space complexity, that is, the storage space required by the algorithm, can also be considered.

Robustness refers to the ability of an algorithm to maintain stable performance despite real-world perturbations such as noise, occlusion, lighting changes, etc. One way to evaluate robustness is to test the algorithm under different conditions and compare the consistency and stability of the output. For example, we can observe the algorithm's performance by adding various degrees of Gaussian noise or changing the brightness and contrast of the image to see if the algorithm can still maintain good performance.

Visual quality refers to whether the resulting image looks natural and realistic. Visual quality is difficult to quantify but can be subjectively assessed through expert review, user surveys, etc. In addition, we can use objective metrics such as the Structural Similarity Index (SSIM) to evaluate the similarity between the refined cloud image and the original image [8]. SSIM measures the structural similarity between images with the formula:

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3}$$

In the formula, x and y represent the original image and the refined image respectively; μ_x and μ_y are the means of the two images; σ_x^2 and σ_y^2 are the variances; σ_{xy} is the covariance of the two images; c_1 and c_2 are small constants used to stabilize the denominator.

In addition to the quantitative metrics mentioned above, we can also intuitively evaluate the effectiveness of algorithms by comparing cloud images produced by different algorithms in terms of the richness of edge details and the overall naturalness and realism of the clouds. Through a comprehensive evaluation of its accuracy, efficiency, robustness, and visual quality, the effectiveness and applicability of the detail processing algorithm for cloud images using fractal theory can be determined. These evaluation criteria help to continuously improve the algorithm and make it more suitable for various application scenarios.

5. Algorithm Overview and Comparative Analysis

5.1 Algorithm Overview

5.1.1 Fractal Brownian Motion (fBm) and Fractal Terrain (FT)

Fractal Brownian Motion (fBm) is a technique widely used in cloud image processing. fBm simulates the irregular texture by superimposing sine waves of different frequencies. It can be expressed as:

$$\mathrm{fBm}(t) = \sum_{k=-\infty}^{+\infty} a_k e^{ikt} \tag{4}$$

In the formula, (a_k) represents the amplitude, (k) represents the frequency, and (t) represents the time or space position. By adjusting the distribution of amplitude (a_k) and frequency (k), we can change the self-similarity and roughness of the generated cloud texture. One advantage of the fBm is that it can generate complex textures with a natural appearance. Nevertheless, it is based on superimposed sine waves, so it may not be as flexible as others in simulating boundary details.

Fractal terrain (FT) is another commonly used technology for processing cloud images based on fractal theory. It simulates the edge of clouds by recursively refining the terrain. The first step of the algorithm is to define an initial simple shape and then increase the details by continuous segmentation and random shift. The recursive refinement process can be expressed as the following formula:

$$\text{New Point} = \text{Old Point} + \text{Random Offset} \times \text{Scale Factor} \tag{5}$$

Random shift come from a standard normal distribution, and the scaling factor diminishes as the recursion levels increase. This method offers the benefit of creating precise boundaries, but it could lead to decreased computational efficiency.

5.1.2 Iterated Function System (IFS)

An iterative function system (IFS) is an effective fractal generation method to simulate clouds' self-similarity and scale-independent characteristics of clouds. IFS consists of a set of affine transformations, each of which can be expressed as:

$$f_i(x) = r_i x + t_i \tag{6}$$

In the formula, (r_i) represents the scaling factor and (t_i) is the translation vector. By randomly selecting these transformations and repeatedly applying them to the initial set of points, the outline of the cloud can be obtained. The IFS can efficiently create cloud images with complex structures, but it needs a meticulously crafted set of affine transformations to replicate the desired type.

5.2 Comparative Analysis

5.2.1 Fractal Brownian Motion (fBm) and Fractal Terrain (FT)

Fractal Brownian Motion (fBm) and fractal terrain (FT) have advantages in generating natural textures, such as cloud images. FBm is known for accurately simulating natural textures. The central idea is constructing complex texture patterns by superimposing a series of sine waves with different frequencies and amplitudes. With this superposition technique, fBm can effectively capture self-similar features that are widely present in the natural world. Nevertheless, fBm predominantly depends on combining sine waves, which might not be sufficient for addressing boundary intricacies, particularly in images that require precise boundaries.

In contrast, fractal terrain (FT) focuses more on simulating the boundary of clouds through recursive thinning. Starting with a simple geometric shape, it adds detail through successive subdivisions and random offsets, creating highly intricate yet natural-looking borders. FT is particularly well suited for image processing tasks that require a high level of detail. However, this process is often time-consuming, especially when higher resolution and more specific details are needed, and the calculation cost is significantly increased.

5.2.2 Iterated Function System (IFS)

Iterative Function System (IFS) provides an efficient way to generate complex structures. IFS uses a series of affine transformations to simulate the self-similarity of natural phenomena. By repeatedly applying these transformations to an initial set of points, very complex texture patterns can be created. One of the advantages of IFS is its computing power and the ability to generate high-quality images without too much computing effort. However, IFS needs to carefully design affine transform sets for simulating certain types of clouds, which may be a challenging task.

5.2.3 Summary

To sum up, fBm, FT, and IFS have their advantages. FBm is good at generating complex textures with a natural look but falls short on edge detail, while FT is good at generating highly detailed edges but has lower computational power. IFS has high computational efficiency and can generate complex structures at the same time. However, it needs to design the transform set carefully. The decision of which method to select relies on the particular application context and the balance between image quality, computational efficiency, and detail needs. For example, IFS may be the best choice when cloud images with self-similar characteristics need to be generated quickly. In the case of fine boundaries and less sensitivity to calculation time, FT may be a suitable way.

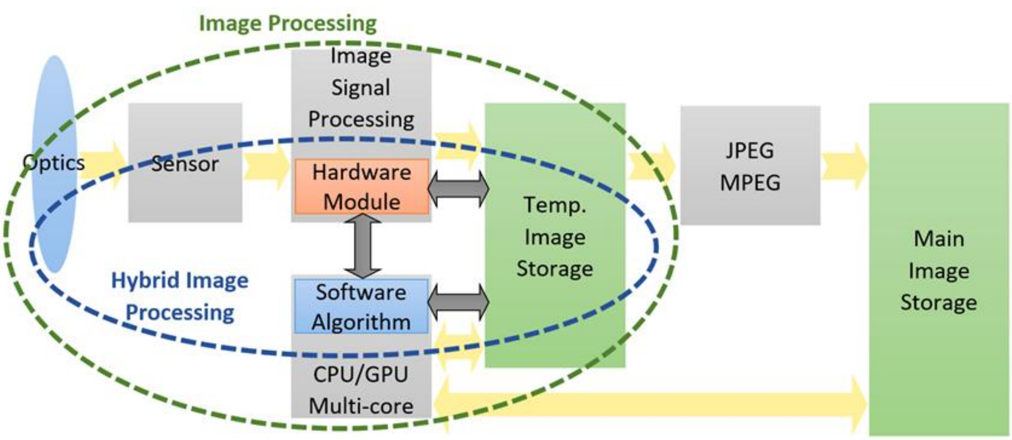


Figure 4. Effects of Different Parameter Settings on Cloud Image Details

6. Conclusion

In recent years, with the rapid development of computer graphics and image processing technology, the generation and processing technology of cloud images has made significant progress. The form of clouds is intricate and variable by nature, with distinct boundary features that are notably exceptional. Hence, precise handling of details in cloud images is now a significant research focus in the realm of computer graphics. Utilizing fractal theory in technology to process cloud images provides notable benefits in enhancing the authenticity and intricacy of cloud formations. By using rational design and optimization algorithms, it is possible to create cloud images with intricate designs that look natural while maintaining a balance between computational speed and level of detail. Additional research can delve into enhancing the performance optimization tactics of these algorithms in real-world scenarios, along with integrating them with other cutting-edge technologies, such as deep learning, to make significant advancements in cloud image processing.

References

Dehshiri, M., Ghavami, S. S., & Khorsandi, A. (2021). Structural similarity assessment of an optical coherence tomographic image enhanced using the wavelet transform technique. *Journal of the Optical Society of America A*, *38*(1), 1-9.

Falcner, K. (2014). Fractal geometry: mathematical foundations and applications. *John Wiley & Sons*.

Fang, J., Li, H., Jia, Y. et al. (2023). Thangka mural style transfer based on progressive style-attentional network and multi-level loss function. *Journal of Electronic Imaging*, *32*(4), 043007-043007.

Guindon, B., & Zhang, Y. (2017). Application of the dice coefficient to accuracy assessment of object-based image classification. *Canadian Journal of Remote Sensing*, *43*(1), 48-61.

Liu, L., Chambers, E. W., Letscher, D. et al. (2010). A simple and robust thinning algorithm on cell complexes[C]//Computer Graphics Forum. *Oxford, UK: Blackwell Publishing Ltd*, *29*(7), 2253-2260.

Setiadi, D. R. I. M. (2021). PSNR vs SSIM: Imperceptibility quality assessment for image steganography. *Multimedia Tools and Applications*, *80*(6), 8423-8444.

Xian, Y., Wang, X., & Teng, L. (2021). Double parameters fractal sorting matrix and its application in image encryption. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(6), 4028-4037.

Zhao, H., & Wu, Q. (2016). Application study of fractal theory in mechanical transmission. *Chinese Journal of Mechanical Engineering*, *29*, 871-879.