

## Research Article

# Underwater Acoustic Image Encoding Based on Interest Region and Correlation Coefficient

Liu Lixin <sup>1</sup>, Guo Feng<sup>1,2</sup> and Wu Jinqiu <sup>3</sup>

<sup>1</sup>*Institute of Deep-Sea Science and Engineering, Chinese Academy of Sciences, Hainan 572000, China*

<sup>2</sup>*University of Chinese Academy of Sciences, Beijing 100049, China*

<sup>3</sup>*Beijing Institute of Control And Electronic Technology, Beijing 100038, China*

Correspondence should be addressed to Liu Lixin; [liulx@idsse.ac.cn](mailto:liulx@idsse.ac.cn)

Received 19 July 2018; Revised 17 September 2018; Accepted 8 October 2018; Published 21 October 2018

Academic Editor: Danilo Comminiello

Copyright © 2018 Liu Lixin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It is difficult for the conventional image compression method to achieve good compression effect in the underwater acoustic image (UWAI), because the UWAI has large amount of noise and low correlation between pixel points. In this paper, fractal coding is introduced into UWAI compression, and a fractal coding algorithm based on interest region is proposed according to the importance of different regions in the image. The application problems of traditional quadtree segmentation in UWAI were solved by the range block segmentation method in the coding process which segmented the interest region into small size and the noninterest region into large size and balanced the compression ratio and the decoded image quality. This paper applies the classification, reduction codebook, and correlation coefficient matching strategy to narrow the search range of the range block in order to solve the problem of the long encoding time and the calculation amount of encoding process is greatly reduced. The experimental results show that the proposed algorithm improves the compression ratio and encoding speed while ensuring the image quality of important regions in the UWAI.

## 1. Introduction

The UWAI refers to an image generated by imaging sonar according to the characteristics of the underwater target echo signal by means of acoustic wave detection. Due to the limitation of the underwater acoustic (UWA) channel and the storage capacity of the device [1, 2], the UWAI must be compressed to meet the actual application requirements. Compared to ordinary optical images, the UWAI is affected by uncertain factors such as underwater environment, various types of noise, ocean reverberation, ocean internal heterogeneity, and seabed terrain irregularity, which is seriously polluted by kinds of noises. Generally, the signal to noise ratio is low, and there are many outliers that differ greatly from the gray values of surrounding pixels. The image target area has less gray level; however the background area has rich gray level. The whole image is rough, and the complex and fine edges are missing [3, 4].

Due to the particularity of UWAI, many methods used for optical image compression cannot always achieve good

results. The traditional image compression method is based on removing the correlation redundancy in the image. Due to the existence of noise, the correlation between the pixels of the UWAI is very poor. At the same time, limited by the entropy, the compression ratio of traditional image compression method is generally not high in the UWAI compression. Therefore, this paper applies the fractal coding based on the fractal theory and iterative function system to UWAI compression to improve the image compression performance. Fractal coding algorithm breaks the limitations of the information theory which is based on the widely existing self-similarity in images. The basic idea is to achieve high compression ratio by removing the structural redundancy through a set of compression transformations. Since Barnsley first applied fractal theory to image compression in 1988 [5], fractal coding has attracted wide attention by its large compression potential, fast decoding speed, and resolution of decoded image resolution.

In 1992, inspired by Barnsley's research, Jcaquin proposed fractal coding based on the local iterative function

system [6], which made the fractal coding into practice and promoted the rapid development of fractal coding research. However, there are still some problems in the fractal coding. (1) The coding process is complicated and the amount of calculation is large, which costs long time for the coding procedure. (2) The compression ratio and the decoded image quality cannot be balanced. In order to solve the long fractal coding time, scholars have proposed several methods such as classification matching [7–10], optimized search range, and nonsearch coding [11–15]; scholars propose a variety of different range block partitioning schemes, such as quadtree partitioning [16–18], HV segmentation [19], triangulation, and irregular segmentation [20, 21] to balance the problem of compression ratio and decoded image quality. The latter three methods require more information to describe the position and shape of each subblock compared with quadtree partitioning, and the search matching processes are more complicated. In image processing, the K-Means algorithm which belongs to the machine learning algorithm is a clustering algorithm based on distance similarity [22, 23]. By comparing the similarity between samples, the samples of the form are divided into the same category. However, the usage of AI algorithms requires a large number of data sets such as different signal-to-noise ratios, different textures, different types, different sources of UWAI, and then the algorithm can find the most suitable codebook based on various feature information of the images, which means a large number of underwater acoustic experiments is required and which is unrealizable currently because the acquisition of UWAI is difficult. Therefore, this paper utilizes the widely used quadtree partitioning algorithm in the segmentation of UWAI. However, because of the widespread noise in UWAI, most of the segmented images have little or no large size image blocks which causes difficult in the application of the quadtree partitioning algorithm. Focusing on this, we propose the improved quadtree partitioning algorithm which is based on the region of interest. In addition, by classifying the range blocks, the search range of the range blocks is optimized which can significantly improve the encoding speed. At the same time, the codebook capacity is reduced and the correlation coefficient is matched which can not only balance the compression ratio and the decoded image quality but also gain a higher encoding speed.

The structure of this paper is as follows: the second section briefly introduces the principle of basic fractal coding; the third section focuses on the fractal coding algorithm based on region of interest and the correlation coefficient fractal coding algorithm is proposed; the simulation and experiment results are given in the fourth section; the fifth section comes the conclusion.

## 2. Basic Fractal Coding

The basis fractal coding (BFC) is to find a set of compression transformations whose fixed points are similar with the original image. These sets of compression transformation are called the Iterated Function System (IFS). Jacquin proposed the Partial Iterative Function System (PIFS) and the basic

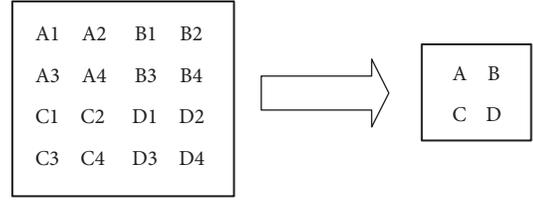


FIGURE 1: Four neighborhood pixel average.

fractal coding, inspired by IFS. The implementation of BFC algorithm is as follows.

When encoding, the image is first divided into a Range Block (R block) and a Domain Block (D Block). The R blocks do not overlap with each other, and the entire R blocks can form the original image. The D blocks can be generated by sliding a certain size window in the original image according to a preset step size which are allowed to have overlapping areas. To ensure the convergence of PIFS, the block size of D should be larger than that of R (usually the D block size is twice larger than the R block).

Supposing the size of the original image is  $M \times M$ , the block size of R should be  $B \times B$  and the block size of D block will be  $2B \times 2B$ . The sliding step is represents by  $\delta$ , and then the whole image is divided into  $M^2/B^2$  R blocks and  $((M - 2B)/\delta + 1)^2$  D blocks.

In order to match the D and R blocks, the D block needs to do the spatial contraction, and its size is scaled from  $2B \times 2B$  to  $B \times B$ . Usually, the four-neighbor pixel averaging method is used for spatial contraction (as shown in Figure 1), and S represents the spatial contraction operation.  $D'$  is the contracted D block, and then the value of each pixel in  $D'$  can be written as

$$d'_{k,l} = \frac{d_{2k-1,2l-1} + d_{2k-1,2l} + d_{2k,2l-1} + d_{2k,2l}}{4} \quad (1)$$

where  $d_{k,l}$  and  $d'_{k,l}$  are the gray values of  $D_j$  and  $D'_j$  at the point  $(k, l)$ , respectively.

For each R blocks, it is necessary to find the closest block in the contracted D blocks to obtain the fractal code. According to PIFS, between the R block and the D block there is an affine similarity, which means that the R block is similar to the D block after the equidistant transformation (rotation, flipping, etc.). Jacquin simplified and summarized the isometric transformations, which is shown in Table 1.

In the table,  $D_{i,j}$  represents the pixel value of point  $(i, j)$  and  $t_k$ ,  $k = 1, 2, \dots, 8$  represents the isometric transformation.

After the entire D block is contracted and isometric transformed, the codebook  $\Omega$  (also called the definition domain pool) required for block matching is formed. For convenience, the block in the codebook  $\Omega$  is represented by  $D$ .

After the above process, for any  $R_i$ , the D block in the codebook  $\Omega$  is selected for match searching, that is, to find a best matching block  $D_{m(i)}$ , so that  $D_{m(i)}$  is the closest one to  $R_i$  after proper affine transformation, which is

$$R_i \approx s \cdot t_k (D_{m(i)}) + o \quad (2)$$

TABLE 1: 8 kinds of isometric transformations.

Index	Transformation name	Transformed pixel
1	Identity transformation	$(t_1 D)_{i,j} = D_{i,j}$
2	Axisymmetric reflection on the vertical axis	$(t_2 D)_{i,j} = D_{i,B-1-j}$
3	Axisymmetric reflections on the horizontal	$(t_3 D)_{i,j} = D_{B-1-i,j}$
4	Symmetric reflection on the main diagonal	$(t_4 D)_{i,j} = D_{j,i}$
5	Symmetric reflection on the sub-diagonal	$(t_5 D)_{i,j} = D_{B-1-j,B-1-i}$
6	90° counterclockwise about the center	$(t_6 D)_{i,j} = D_{j,B-1-i}$
7	180° counterclockwise about the center	$(t_7 D)_{i,j} = D_{B-1-i,B-1-j}$
8	270° counterclockwise about the center	$(t_8 D)_{i,j} = D_{B-1-j,i}$

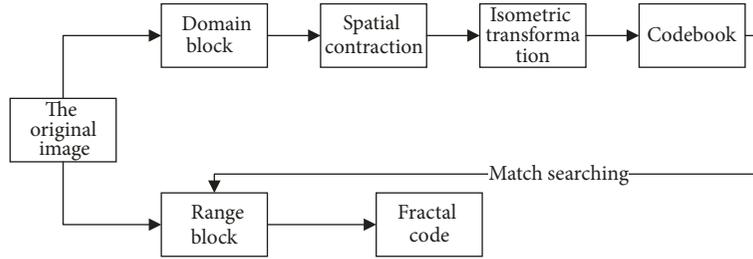


FIGURE 2: The procedure of the BFC.

The matching error between  $D$  block and  $R$  block is  $E(R, D)$ , which is calculated by the following [24, 25]:

$$\begin{aligned}
 E(R, D) &= \|R - \bar{R}\|^2 - s^2 \|D - \bar{D}\|^2 \\
 s &= \frac{\langle R - \bar{R}, D - \bar{D} \rangle}{\|D - \bar{D}\|^2} \\
 o &= \bar{R} - s \cdot \bar{D}
 \end{aligned} \tag{3}$$

When searching in different  $D$  blocks, the one with the smallest matching error  $E(R_i, D)$  is selected as the best matching block. The procedure can be expressed as

$$\begin{aligned}
 E(R_i, D_{m(i)}) &= \min_{D \in \Omega} \{E(R_i, D)\} \\
 &= \min_{D \in \Omega} \{\|R_i - (s \cdot D + o)\|^2\}
 \end{aligned} \tag{4}$$

in which  $\langle \cdot, \cdot \rangle$  represents the inner product;  $R$  and  $D$  represent  $R_i$  and  $D_j$ , respectively;  $\bar{R}$  and  $\bar{D}$  represent the constant block formed by the mean value of  $R$  and  $D$ ;  $\|\cdot\|$  represents the vector 2 norm. Such tetrad  $\{m(i), k, s, o\}$  is the fractal code of block  $R$ .

The procedure of the BFC is shown in Figure 2.

The fractal decoding process is relatively simple. By compression transformation, the decoded image can be applied to any image of the same size as the original image.  $w$  corresponds to the fractal code after 5~8 iterations (8

iterations are selected in this paper). The  $R$  block of the image  $I^{(k)} = wI^{(k-1)}$  generated by the  $k$ th iteration is determined by

$$\begin{aligned}
 R_i^{(k)} &= s \cdot t_k \left( S \left( D_{m(i)}^{(k-1)} \right) \right) + o \\
 D_{m(i)}^{(0)} &= D_{m(i)}
 \end{aligned} \tag{5}$$

### 3. The Proposed Algorithm

**3.1. Quadtree Partitioning Based on Region of Interest.** As can be seen from Section 2, the fractal coding method only encodes each  $R$  block after image segmentation, not the entire image. Therefore, the block size of  $R$  directly affects the compression ratio and the image quality. The larger the block size of  $R$  is, the less the block number of  $R$  will be. Accordingly, the less ‘‘entry’’ in fractal code, the higher compression ratio which causes the increasing of the matching error. On the contrary, the smaller the block size of  $R$  is, the better the decoded image quality will be, but the compression ratio becomes lower. In order to balance the compression ratio and the decoded image quality, this paper improved the fractal coding based on quadtree partitioning.

Quadtree partitioning is similar to the tree structure in data structure. The original image can be regarded as the root of the tree, and the four fork points (corresponding to the four quadrant regions of the image respectively) are separated from the root node. Each of the fork points can select whether to further divide the four new subnodes according to certain rules (corresponding to four sub-regions of the image area corresponding to the intersection, respectively). The new subnode continues to split according to this rule, and when the segmented node reaches the maximum segmentation

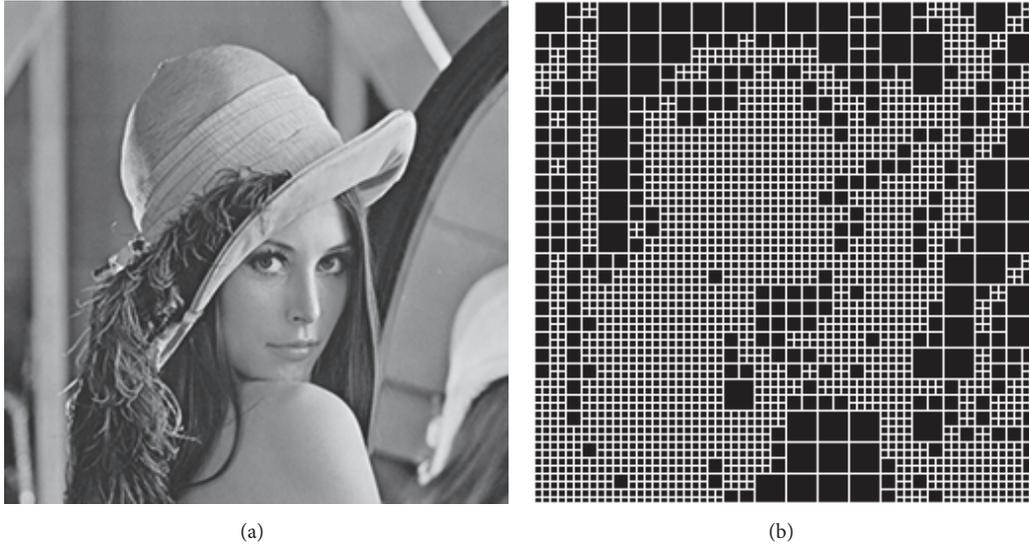


FIGURE 3: Ordinary gray image quadtree segmentation: (a) original image Lena and (b) effect of quadtree segmentation.

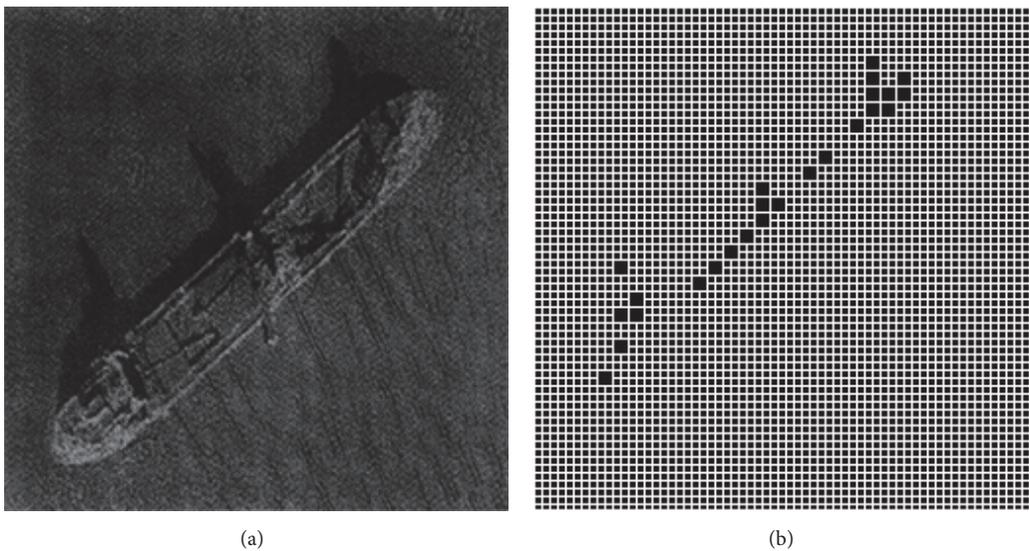


FIGURE 4: UWAI quadtree segmentation: (a) original image ship and (b) effect of quadtree segmentation.

depth, the segmentation will stop. Then the entire image is divided into several subblocks with different sizes.

In practice, the gray-scale uniform criterion is generally used for the quadtree partitioning. When the block size of one image is larger than the minimum segment size, which means the difference between the maximum gray value and the minimum gray value is greater than a preset threshold, the block is further divided. Otherwise the block does not need to be split.

For different size image blocks generated after segmentation, the fractal code can be obtained by match searching in the corresponding codebooks using the basic fractal coding method. Therefore, the quality of the decoded image can be increased by dividing the portion of the image with a smaller difference in the pixel value by a larger size, thereby increasing

the compression ratio and dividing the portion with a larger difference in the pixel value by a smaller size, thereby ensuring the decoding image quality.

Figures 3 and 4 show the effect after the quadtree segmentation of the ordinary grayscale image Lena and the UWAI Ship (the maximum segmentation size is  $16 \times 16$ , and the minimum segmentation size is  $4 \times 4$ ) respectively. As can be seen from Figure 3, the Lena image has a good segmentation effect, all blocks of  $16 \times 16$ ,  $8 \times 8$ , and  $4 \times 4$  account for a certain percentage. Due to the noise in UWAI, the effect of quadtree partitioning is not obvious. All  $16 \times 16$  blocks in the image are not smaller than the gray threshold and, therefore, further subdivision is needed. For the  $8 \times 8$  blocks subdivided, only a few blocks satisfy the requirement with no need for further segmentation, most of blocks are divided into  $4 \times 4$  blocks. It is

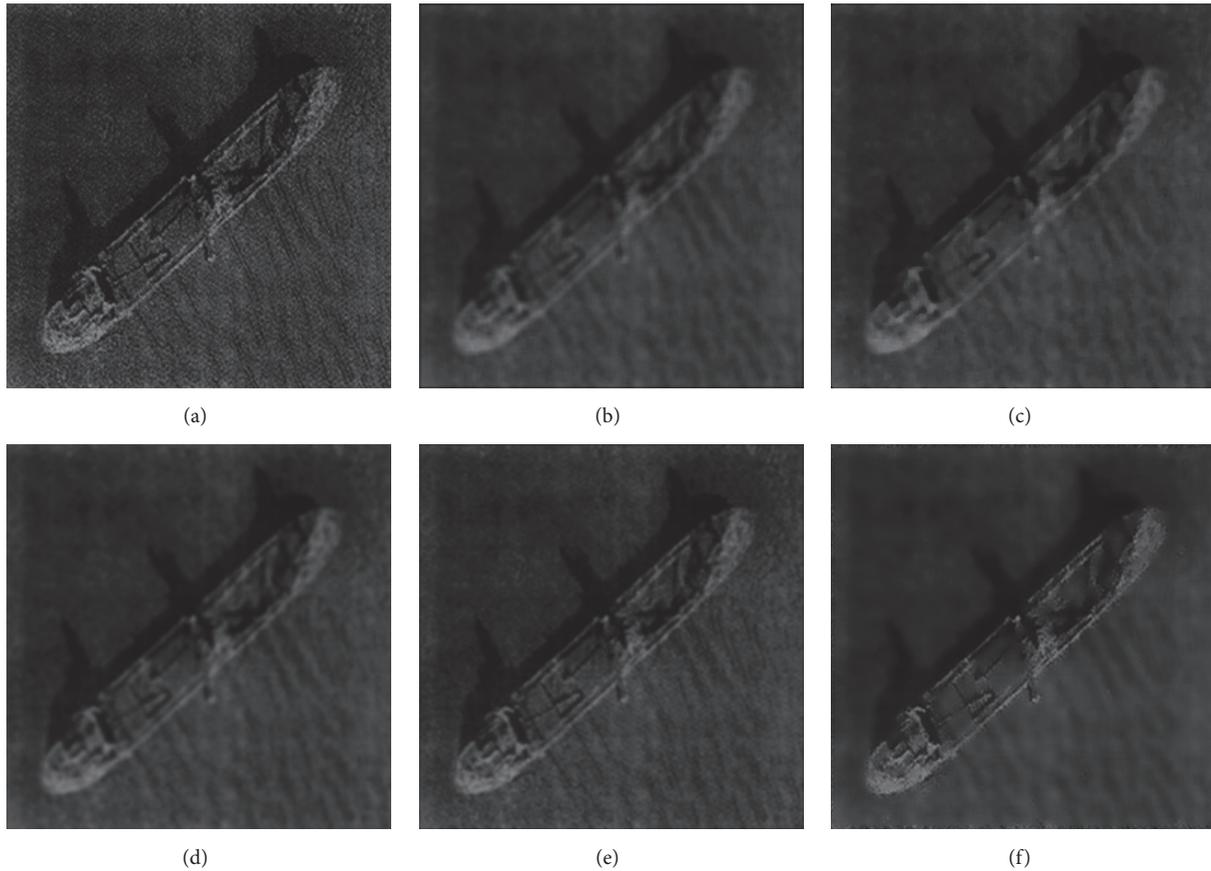


FIGURE 5: Denoising effect of different methods: (a) original image; (b) average filtering; (c) median filtering; (d) Butterworth low-pass filtering; (e) Gaussian low-pass filtering; (f) Wiener filtering.

conceivable that when image noise pollution is more serious, the effect of quadtree segmentation will be exactly the same as  $4 * 4$  segmentation. Therefore, new quadtree segmentation methods that suitable for the UWAI are needed.

Considering that the main information of the UWAI is concentrated in the area where the target is located, the noisy background area does not need much attention. For this reason, different size segmentation is used for different important areas in the UWAI in this paper. The important area in the image (hereinafter referred to as the region of interest) is divided by a small size, and a large size segmentation is adopted for the noninterest region. The specific segmentation process is as follows.

(1) *Image Denoising*. UWAI usually has noise interference, so the image needs to be denoised first to reduce the impact of these interferences on the region of interest in the image. Since this step is to prepare for the extraction of the region of interest in the next step, in order to reduce the nonedge line interference, the image is smoothed while maintaining the contour of the region of interest while denoising. In this section, the average filtering, median filtering, Butterworth low-pass filtering, Gaussian low-pass filtering, Wiener filtering, and other methods are used for denoising. Figure 5 shows the denoising effect of the UWAI Ship by different methods. It

can be seen that the five methods produce different degrees of smoothing effects on the image. The average filtering, median filtering and Butterworth low-pass filtering blur the boundary between the target and the background while denoising, which is not conducive to the extraction of the region of interest. The Gaussian low-pass filtering does not have the problem of boundary blurring, but the smoothing effect is not good, while wiener filtering has good denoising effect and maintains the contour of the target area which is the reason for selecting it in this paper.

(2) *Extraction of Interest Region*. In this section, the interest region in the UWAI is extracted by means of edge detection and morphological operations. Both the target area and the shadow area formed by UWAI contain important information. The principle of extracting the area of interest is to include some background areas, which will result in the final area of interest being larger than the range extracted by traditional target segmentation. The specific steps for extracting the interest region are as follows (for the sake of convenience, the target and the shadow are collectively referred to as the target):

① Use the canny operator for edge extraction on the denoised image.

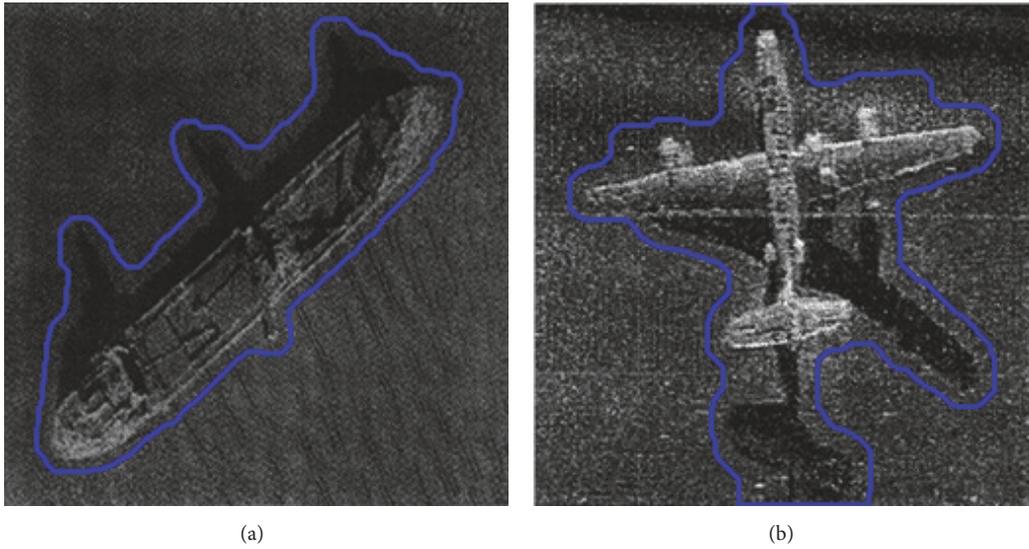


FIGURE 6: Extract results of interest region: (a) ship and (b) plane.

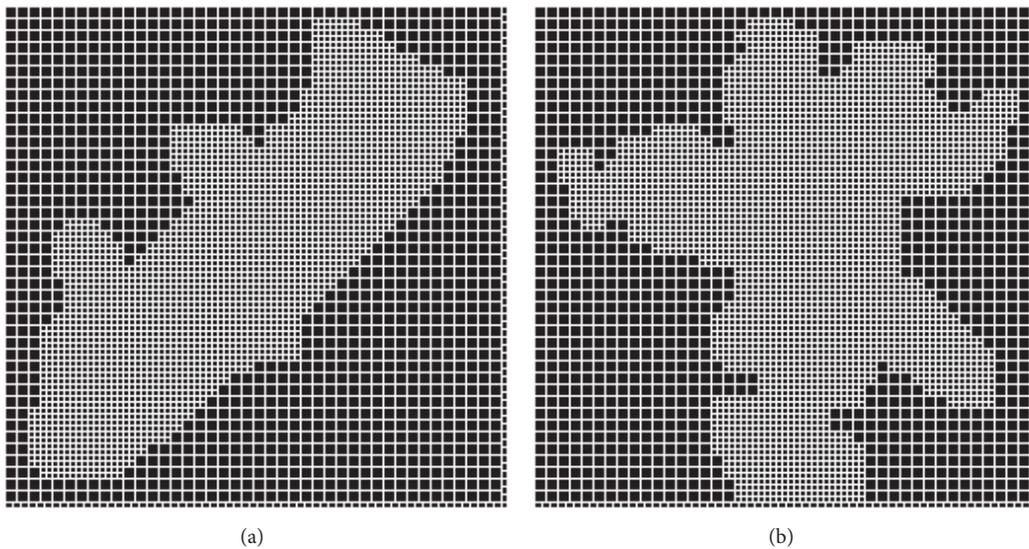


FIGURE 7: Segmentation effect of UWAI: (a) ship and (b) plane.

② Morphological expansion is used to connect discontinuous target edge lines into closed lines.

③ The nontarget contour lines caused by interference are removed by morphological operation.

④ Fill the target contour area.

⑤ Morphological expansion is applied to expand the target area, and the expanded area is the required interest region for coding.

Figure 6 shows the results of the extraction of the regions of interest for the two UWAI

(3) *Image Segmentation.* On the basis of Section 2, small-size segmentation is performed for the region of interest, and large-scale segmentation is adopted for the non-interest

region. Figure 7 shows the segmentation effect of UWAI by the small size  $4 \times 4$  and the large size  $8 \times 8$ .

3.2. *Reducing the Search Area of R Blocks.* After the UWAI is segmented, R blocks of different sizes are generated. For each R block, it is necessary to search for the best matching block in its corresponding codebook to obtain a fractal code. According to the basic fractal coding proposed by Jacquin, the R block searches in all codebooks, which will cost a long time on encoding. Therefore, the search area of R block is reasonably reduced, and the matching process between R block and D block is optimized to reduce the calculation amount of encoding process and improve the encoding speed.

From formula (3):

$$\begin{aligned} E(R, D) &= B^2 \left( \text{var}(R) - s^2 \cdot \text{var}(D) \right) \\ s &= \frac{\langle R - \bar{R}, D - \bar{D} \rangle}{B^2 \cdot \text{var}(D)} \\ o &= \bar{R} - s \cdot \bar{D} \end{aligned} \quad (6)$$

in which,  $B^2$  is the total number of R block pixels,  $\bar{X}$  is the mean value of  $X$ ,  $\langle \cdot, \cdot \rangle$  represents the vector inner product, and  $\text{var}(X)$  is the variance of  $X$ .

As can be seen from (6), the following holds for any  $D \in \Omega$ :

$$E(R, D) \leq B^2 \cdot \text{var}(R) \quad (7)$$

When the variance of the R block is small, the matching error is accordingly small. Therefore, the R block is divided into two categories according to the variance: the R block whose variance is less than the preset threshold  $\eta^2$  is regarded as a smooth block; otherwise it is regarded as a nonsmooth block. For a smooth block, the pixel value of each point is replaced by its pixel mean value, so there is no need to search again. For nonsmooth blocks, in the next section the following methods are adopted.

Equation (5) also shows that the D block with smaller variance is unlikely to be the best matching block of the R block with larger variance (if these two are the best matching pairs, it is likely that  $s$  is greater than 1 and then the PIFS will no longer converges). Therefore, D blocks with smaller variance can be eliminated from the codebook in advance. The reduction codebook is defined as

$$\Omega_\delta = \{D \in \Omega : \text{var}(D) \geq \delta^2\} \quad (8)$$

in which,  $\delta$  is the preset threshold. For nonsmooth R blocks whose variance is greater than the threshold  $\eta^2$ , it is only necessary to find the best matching D block in the reduced codebook.

### 3.3. Match Searching Based on the Correlation Coefficient.

When searching for the best matching block of the R block, it is necessary to calculate the parameters according to (5) and then calculate the matching error  $E(R, D)$  between the R block and each D blocks. Although Section 3.2 has narrowed down the searching area, the computation of this process is still large. If the D blocks that may not be the best matching block of the R block can be removed in advance, a large number of complicated calculations can be avoided and the encoding speed can be improved. In this paper, according to the correlation coefficient between R block and D block, D blocks with little correlation is excluded; thus the encoding process can be speed up.

For R block and D block, the correlation coefficient is represented by

$$r(R, D) = \frac{\langle R - \bar{R}, D - \bar{D} \rangle}{\|R - \bar{R}\| \cdot \|D - \bar{D}\|} \quad (9)$$

in which  $\bar{X}$  represents the mean of  $X$ ,  $\langle \cdot, \cdot \rangle$  represents the product of the vector, and  $\| \cdot \|$  represents the norm of vector 2.

This is available in formula (3)

$$\begin{aligned} E(R, D) &= \|R - \bar{R}\|^2 - s^2 \|D - \bar{D}\|^2 \\ &= \|R - \bar{R}\|^2 \\ &\quad - \left( \frac{\langle R - \bar{R}, D - \bar{D} \rangle}{\|D - \bar{D}\|^2} \right)^2 \|D - \bar{D}\|^2 \\ &= \|R - \bar{R}\|^2 \left( 1 - \left| \frac{\langle R - \bar{R}, D - \bar{D} \rangle}{\|R - \bar{R}\| \cdot \|D - \bar{D}\|} \right|^2 \right) \\ &= B^2 \text{var}(R) (1 - |r(R, D)|^2) \end{aligned} \quad (10)$$

Equation (10) shows that, with the increase of absolute value of the correlation coefficient between the R block and the D block, the possibility of the two blocks becoming matching pairs increases. For the R block of the variance  $\text{var}(R) > \eta^2$  (in which  $\eta$  is a preset threshold), it is known from (10)

$$E(R, D) > B^2 \eta^2 (1 - |r(R, D)|^2) \quad (11)$$

This means that if the R block matches the D block ( $E(R, D)$  is small),  $|r(R, D)|$  is larger which means that enough large  $|r(R, D)|$  is the necessary and sufficient condition for the R block matching for the D block.

According to the above analysis, in the encoding procedure, for each R blocks, the correlation coefficient and the matching error of the first D block in the reduced codebook are calculated, respectively as  $C_0$  and  $E_0$ . And then the correlation coefficient  $C$  between the other D blocks in the reduced codebook and this R block are calculated. If  $C > C_0$ , then the matching error  $E(R, D)$  between block D and block R is further calculated; otherwise the D block is ignored and turned to the next D block. This can prevent the calculation of the parameter  $s$  and the matching error for all D blocks, which greatly reduces the computation complexity of the entire encoding process and saves the encoding time.

### 3.4. The Encoding Procedure of the Proposed Algorithm.

For a UWAI to be compressed, firstly extract the interest region. Then divides the interest region and the non-interest region into different sizes, and generates the codebooks with different size simultaneously. A reduced codebook is generated according to the threshold  $\delta^2$ . According to the method proposed in Section 3.3 of the reduction codebook, the fractal code is obtained by searching and matching the divided R blocks of different sizes. The detailed encoding process is as follows:

(1) Preset the threshold  $\eta$  of R block, codebook threshold  $\delta$ , etc.

(2) According to the method proposed in Section 3.1, the UWAI is segmented by quadtree partitioning based on region of interest.

(3) According to the R block partition size, two codebooks of different sizes are respectively produced; and the reduced codebook is constructed according to the codebook threshold  $\delta$ .

(4) For smooth blocks (R blocks with variance less than), take the constant block as approximated, and the fractal code is directly output without searching.

(5) For nonsmooth blocks (R blocks with variance greater than or equal to  $\eta^2$ ), perform match searching in the reduced codebook. The search strategy is as follows: select the first D block in the reduced codebook and calculate its correlation coefficient  $E_0$  and matching error  $C_0$  with R block. Then, the correlation coefficient  $C$  of other D blocks in the reduced codebook with the selected R block is sequentially calculated. If  $C > C_0$ , the matching error  $E(R, D)$  of the two blocks is further calculated. If  $E(R, D) < E_0$ , let  $C_0 = C$ ; otherwise ignore this D block and move to the next. After traversing all the D blocks in the reduced codebook, select the D block corresponding to  $C_0$  as the best matching block and the fractal code is output.

(6) Complete the encoding of all R blocks to generate an encoded file.

The specific steps of the encoding are shown in Figure 8.

**3.5. The Decoding Process of the Proposed Algorithm.** (1) Read information such as the information of the quadtree partition, the position of the best matching block of the R block in different size codebooks, the equidistant transformation mode, the contrast factor, and the grayscale offset.

(2) Define three images  $Y_1$ ,  $Y_2$ , and  $X$  having the same size with the original image, in which  $X$  is used to store the image generated by the iterative process and  $Y_1$  and  $Y_2$  are used to generate two codebooks with different sizes.

(3) Initialize  $Y_1$  and  $Y_2$ .

(4) For each R block in  $X$ , according to its fractal code  $\{m(i), k, s_i, o_i\}$ , locate its corresponding best matching block  $D_{m(i)}$  in the corresponding codebook  $Y$  and restore the R block according to the following formula:

$$R_i = s_i \cdot (t_k(S(D_{m(i)}))) + o_i \quad (12)$$

in which  $S$  is a four-neighbor averaging operation. After all the R blocks are obtained by the above affine transformation, the iterative image is formed by tiling the above R blocks.

(5) If the number of iterations reaches 8, go to step (6); otherwise, copy  $X$  to  $Y_1$ ,  $Y_2$  and go to step (4).

(6) Output image  $X$ .

## 4. Experimental Results and Analysis

In order to verify the effectiveness of the algorithm, two UWAs: ship and plane were used for the simulation experiment. The test image is shown in Figure 9. The PC configuration used in the experiment is as follows: CPU: Intel Core i5-3470, Memory: 8G, Operating System: Windows10, Simulation Software: MATLAB2015B. The parameters that test the performance of the algorithm are as follows: coding time, PSNR, and compression ratio. PSNR is the objective

evaluation standard for quality of the decoded image, which is obtained by

$$PSNR = 10 \log \frac{255^2 \cdot M^2}{\sum_{x=1}^M \sum_{y=1}^M [f(x, y) - f'(x, y)]^2} \quad (13)$$

In which  $f(x, y)$  is the pixel value of the original image with size  $M \times M$  at the point  $(x, y)$  and  $f'(x, y)$  is the pixel value of the decoded image with size  $M \times N$  at the point  $(x, y)$ .

(1) *The Influence of Parameters on Encoding Performance.* The two parameters  $\eta$  and  $\delta$  in the algorithm directly affect the encoding time and the quality of the decoded image. The optimal values for the two parameters are difficult to determine theoretically and therefore they can only be selected experimentally. In order to observe the influence of parameters on the decoded image clearly, the Lena image is selected as test image. When analyzing the influence of the parameter  $\eta$  on the coding performance, it is set as  $\delta = 0$  to eliminate the interference of other factors. Similarly, when analyzing the parameter  $\delta$  accordingly set  $\eta = 0$ .

① Experimental study of the parameter  $\eta$ . The larger  $\eta$  is, the more number of R blocks are considered as smooth blocks. Since the smoothing block does not need to perform match searching, the encoding speed is faster. However, since more R blocks are smoothed, the quality of decoded images is also degraded (such as Figure 10). It can be seen from Figure 10 that when  $\eta > 4$ , blocking effect occurs in some parts of the decoded image, and the larger  $\eta$  is, the more obvious the blocking effect will be. When  $\eta \leq 4$ , the blocking effect is almost negligible. Figure 11 is the comparison of decode image with different parameters. Therefore, the best value of this parameter is 4.

② Experimental study of the parameter  $\delta$ . The value of  $\delta$  determines the size of the reduced codebook. As can be seen from Figure 12, the larger  $\delta$  is, the shorter the encoding time will be. When  $\delta$  increases from 0 to 20, the PSNR decreases slowly, and when  $\delta$  increases from 20 to 45, the PSNR drops rapidly. This means when the threshold  $\delta$  is small, the encoding time can be significantly reduced at the cost of the decoded image quality's slightly lower. While the threshold  $\delta$  is large, the best matching block is likely to be excluded from the codebook, resulting in degrade in the quality of the decoded image quality. Considering the encoding time and the decoded image quality, this paper chooses 20 as the best value for  $\delta$ .

(2) *Comparison of the Proposed Algorithm with Other Algorithms.* The maximum segment size of the algorithm in this paper is  $8 \times 8$ , and the minimum split size is  $4 \times 4$ . The R block size of other algorithms used for comparison is also chosen as the above two sizes. Table 2 lists the experimental results of the proposed algorithm and the basic fractal algorithm (BFC) [6], the variance based fast search algorithm (VBFC) [26] and the particle swarm optimization algorithm (PSO) [27].

It can be seen from Table 2 that as the partition size increases from  $4 \times 4$  to  $8 \times 8$ , the coding time and the PSNR of the BFC algorithm, the VBFC algorithm, and the PSO algorithm are gradually reduced and the compression ratio

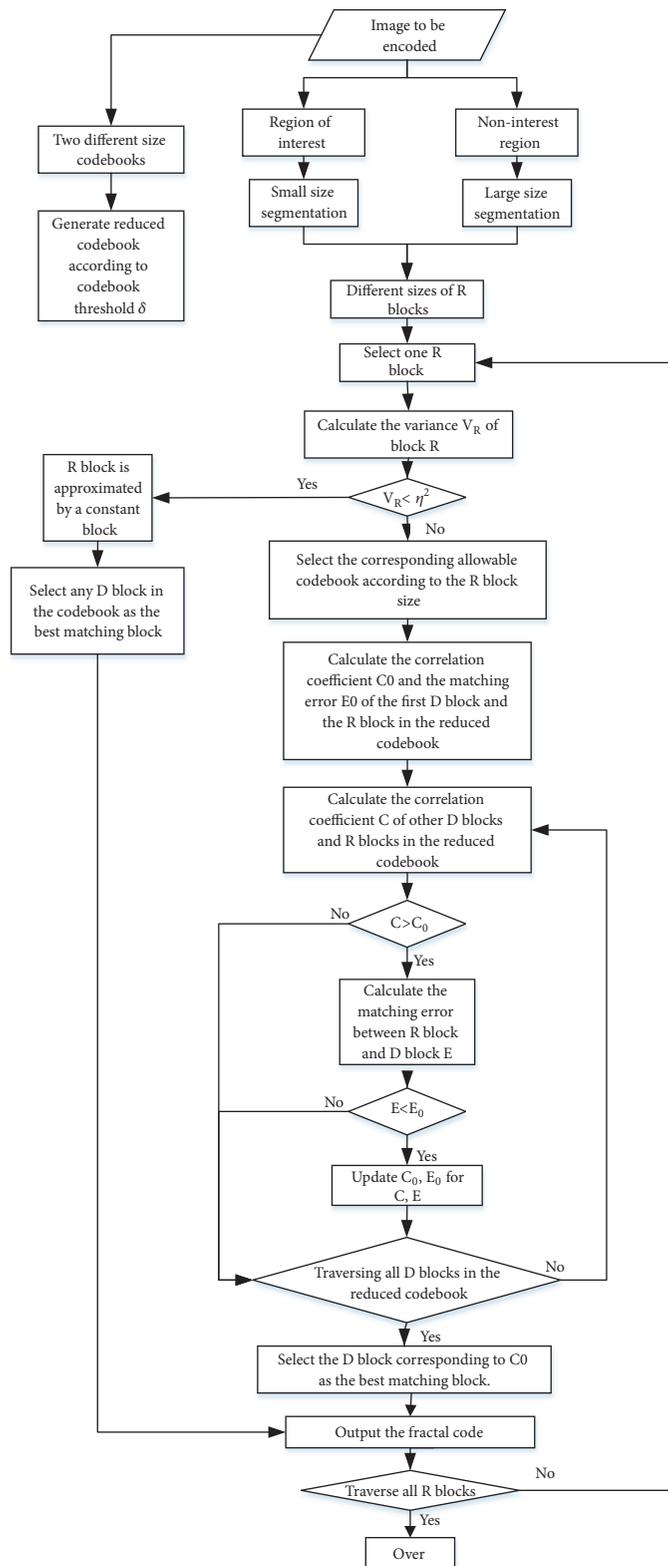


FIGURE 8: Specific steps of encoding based on the interest region algorithm.

TABLE 2: Experimental results comparison of different algorithms.

Algorithm	Image					
	Encoding Time (s)	Ship PSNR (dB)	Compression ratio	Encoding Time (s)	Plane PSNR (dB)	Compression ratio
BFC ( $4 \times 4$ )	30.40	159.70	5.12	29.19	160.12	5.12
BFC ( $8 \times 8$ )	26.58	40.41	20.48	25.02	40.54	20.48
VBFC ( $4 \times 4$ )	29.80	17.71	5.12	28.41	17.31	5.12
VBFC ( $8 \times 8$ )	26.38	5.05	20.48	24.75	5.12	20.48
PSO ( $4 \times 4$ )	29.23	11.78	5.12	28.04	11.38	5.12
PSO ( $8 \times 8$ )	25.69	3.46	20.48	24.25	3.57	20.48
The proposed algorithm	28.77	3.02	13.87	27.49	3.15	13.25

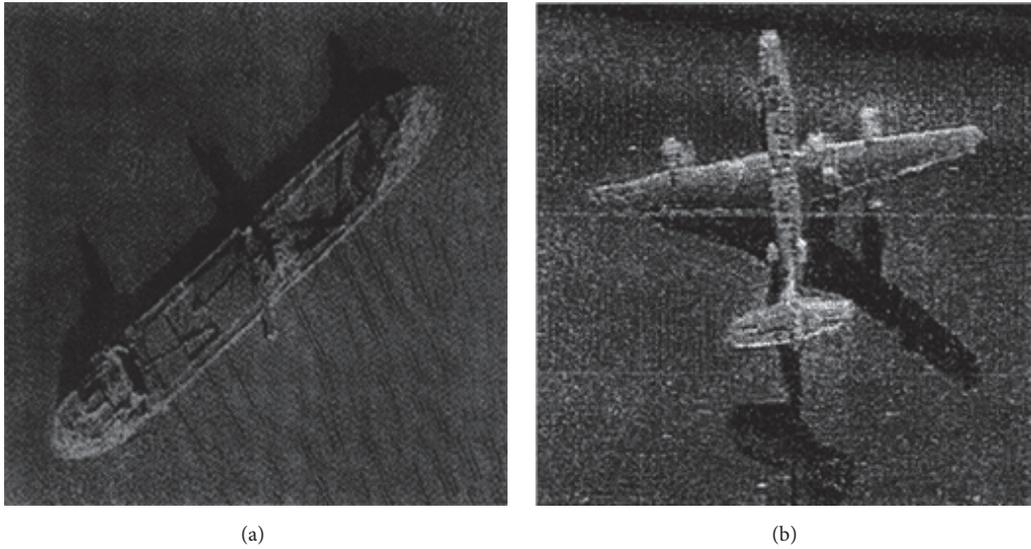
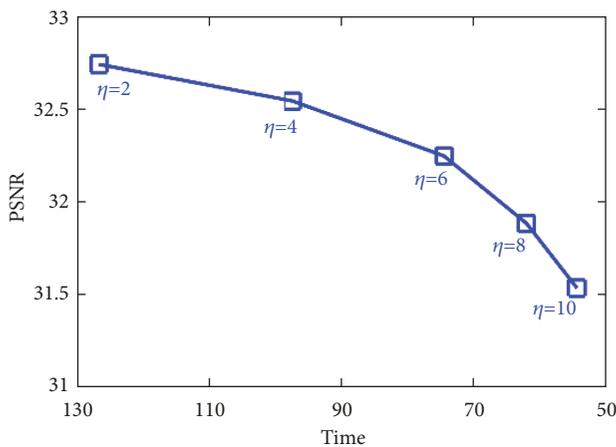


FIGURE 9: UWAI: (a) ship and (b) plane.

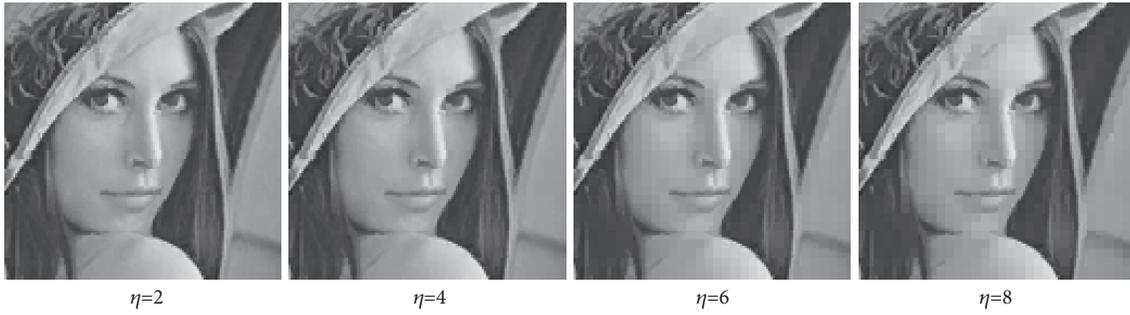
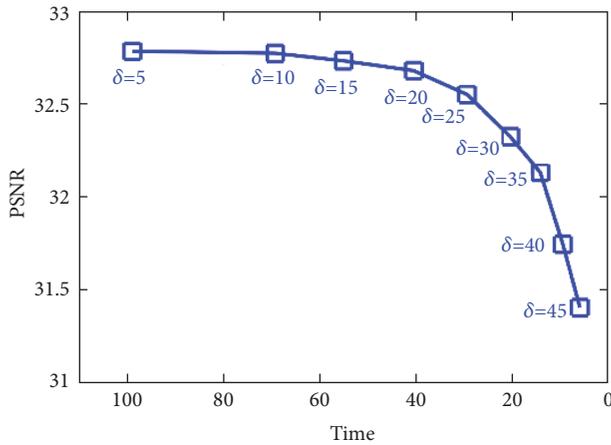
FIGURE 10: Conversion curve of encoding performance with  $\eta$ .

is gradually increased. In the comparison of the encoding time, the proposed algorithm of is only 1/53 of the  $4 \times 4$  BFC algorithm and 1/13 of the  $8 \times 8$  BFC algorithm, which is mainly own to the optimization of the encoding process.

According to the strategy of Section 3.2, by reducing the codebook capacity, searching for matching based on correlation coefficient, setting the ending condition of matching, etc. The calculation amount of the encoding process is effectively reduced, and the encoding speed is greatly improved.

Since most of the images are noninterest regions, this paper adopts a large size  $8 \times 8$  segmentation for these parts. And for interest region, the small size  $4 \times 4$  segmentation is adopted. Therefore, the compression of the algorithm in this paper is significantly improved compared with the basic  $4 \times 4$  algorithm. In the quality of the decoded image, it can be seen from Figure 13 that the decoding quality of the interest region in the image is close to the BFC  $4 \times 4$  algorithm, and the recovery of the noninterest region approaches the BFC  $8 \times 8$  algorithm. Although the PSNR is not as good as the BFC  $4 \times 4$  algorithm, the important information in the image is well recovered. The proposed algorithm balances the compression ratio and the decoded image quality well.

The VBFC algorithm and the PSO algorithm use the method of variance approximation matching and particle swarm optimization to improve the encoding speed. It can be seen from the experimental results that the coding time of

FIGURE 11: Decode image comparison with different  $\eta$ .FIGURE 12: Conversion curve of encoding performance with different  $\delta$ .

the VBFC and PSO algorithms decreases with the increase of the R block partition size, but even the PSO ( $8 \times 8$ ) algorithm with the smallest coding time is also longer than the encoding time of the proposed algorithm. This is mainly because the algorithm in this paper comprehensively adopts multiple strategies such as reduced codebook and fast search based on correlation coefficient, while VBFC algorithm and PSO algorithm only improve the match searching mode, and the calculation efficiency is not good than the proposed algorithm.

It can be seen from Figure 13 that the quality of the decoding image in the proposed algorithm is higher than the VBFC and PSO algorithms when the R block partition size is  $8 \times 8$ . For the interest region in the image, in order to reduce the matching error and improve the image quality, this paper adopts a small size  $4 \times 4$  segmentation to preserve the important details in the image. Results show that the recovery quality is better than the other two algorithms with the same segmentation size. In addition, the compression ratio of the algorithm proposed in this paper is also significantly higher than that of the VBFC and PSO algorithms using  $4 \times 4$  segmentation (as shown in Table 2).

The proposed algorithm is also compared with the quadtree fractal algorithm. The quadtree fractal algorithm was first proposed by Fisher, but Fisher's method needs to calculate the matching error of four subblocks in each block,

and the matching process adopts the full search strategy which is the same as the BFC algorithm, resulting in a large amount of computation and long encoding time. Therefore, this paper chooses an improved quadtree fractal coding algorithm for comparison. The improved algorithm firstly performs quadtree decomposition on the image according to the gray uniformity criterion and then performs matching search on the decomposed R blocks of different sizes in the reduced codebook. The encoding efficiency is greatly improved compared with Fisher's method.

The maximum segmentation size of the improved quadtree algorithm and the proposed algorithm are  $8 \times 8$ , and the minimum segmentation size is  $4 \times 4$ . As can be seen from Table 3, when the quadtree algorithm is used to decompose the image, the small size image block occupies a large proportion, which leads a significantly lower compression ratio of the quadtree improvement algorithm compared to the proposed algorithm. At the same time, due to the large number of small sized blocks, the quadratic tree improved algorithm has a larger computational complexity and because of adopting the full search strategy, which results in a much longer coding time than the proposed algorithm.

## 5. Conclusion

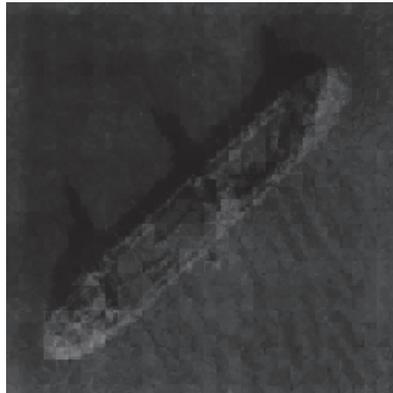
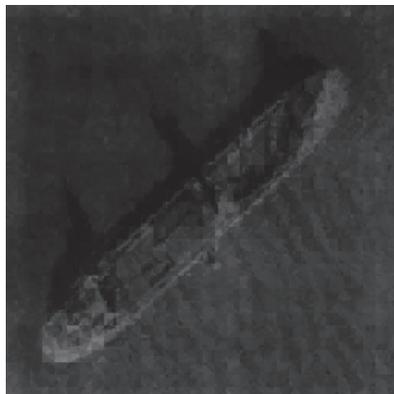
In view of the particularity of UWAI, this paper uses fractal coding based on partial similarity to compress the UWAI. In order to improve the encoding speed and compression ratio, this paper proposes a fractal coding algorithm based on interest region and correlation coefficient. The algorithm divides the interest region in the image into small size and divides the noninterest region into large size, which effectively increases the compression ratio and recovers the information of important regions well. At the encoding stage, the R block searching range is reduced by reducing the codebook and classification, and the inappropriate D block is pre-excluded according to the correlation coefficient between the R block and the D block, which greatly reduces the calculation amount of the encoding process. The simulation results verify that the proposed algorithm not only improves the UWAI compression ratio but also significantly reduces the encoding time, while at the same time ensuring the restoration quality of the interest region in the image.

TABLE 3: Comparison of improved quadtree algorithm with the proposed algorithm.

	The number of $4 \times 4$ blocks	The number of $8 \times 8$ blocks	Encoding time	PSNR	Compression ratio
Improved quadtree algorithm	3996	25	186.35	30.18	5.53
Proposed algorithm	1128	742	3.02	28.77	13.87



(a) The original image

(b) BFC ( $4 \times 4$ )(c) BFC ( $8 \times 8$ )(d) VBFC ( $4 \times 4$ )(e) VBFC ( $8 \times 8$ )(f) PSO ( $4 \times 4$ )(g) PSO ( $8 \times 8$ )

(h) The proposed algorithm

FIGURE 13: Decoding image of BFC, VBFC, PSO, and the proposed algorithm.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors acknowledge the project of the National Natural Science Foundation of China (Grant no. 61701487), the Innovation Foundation of Chinese Academy of Sciences (Grant no. CXJJ-17-M126), the Natural Science Foundation of Hainan Province (Grant no. 417211), the Young Talents' Science and Technology Innovation Project of Hainan Association for Science and Technology (Grant no. QCXM201812), the National Key Research and Development Program of China (Grant no. 2016YFC1400100), the Strategic Priority Research Program of Chinese Academy of Sciences (Grant no. XDA13030000), and the Fundamental Research Funds in Heilongjiang Provincial Department of Education (no. 135209239). The authors also thank the Technical Bureau of Qiqihar GYGG-201622.

## References

- [1] L. Liu, H. Bian, S.-I. Yagi, and X. Yang, "A prior-knowledge-based threshold segmentation method of forward-looking sonar images for underwater linear object detection," *Japanese Journal of Applied Physics*, vol. 55, no. 7, 2016.
- [2] W. Jinqui et al., "Influence of pulse shaping filters on PAPR performance of underwater 5G communication system technique," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 4361589, 4 pages, 2017.
- [3] S. Hong, *Research on Key Technologies of Sonar Image Processing*, Harbin Engineering University, 2011.
- [4] S. Zhengyan, *Research on Sonar Image Denoising and Segmentation Technology*, Harbin Engineering University, 2010.
- [5] M. Barnsley and A. Sloan, "A better way to compress images," *Byte*, vol. 1, pp. 215–222, 1988.
- [6] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Transactions on Image Processing*, vol. 1, no. 1, pp. 18–30, 1992.
- [7] Y. Deng and Y. Ke, "Fast fractal image coding scheme," in *Proceedings of the 1996 3rd International Conference on Signal Processing, ICSP'96. Part 1 (of 2)*, pp. 1047–1050, October 1996.
- [8] E. W. Jacobs, Y. Fisher, and R. D. Boss, "Image compression: a study of the iterated transform method," *Signal Processing*, vol. 29, no. 3, pp. 251–263, 1992.
- [9] X.-Y. Wang and D.-D. Zhang, "Discrete wavelet transform-based simple range classification strategies for fractal image coding," *Nonlinear Dynamics*, vol. 75, no. 3, pp. 439–448, 2014.
- [10] L. Weisheng and L. Gaoping, "Improved fractal image coding algorithm for fractional box dimension," *Journal of Southwest University for Nationalities*, vol. 1, pp. 141–145, 2012.
- [11] H. Chuanjiang and H. Xiwei, "Fast fractal image coding algorithm based on image block crossing," *Chinese Journal of Computers*, vol. 10, pp. 1753–1761, 2005.
- [12] C. K. Lee and W. K. Lee, "Fast fractal image block coding based on local variances," *IEEE Transactions on Image Processing*, vol. 7, no. 6, pp. 888–891, 1998.
- [13] Z. Aihua, S. Fei, Y. Pei et al., "Fast fractal coding algorithm based on similarity ratio," *Computer Technology and Development*, vol. 11, pp. 176–178, 2012.
- [14] W. Lina, L. Xiaodong, H. Xinghua et al., "Fast fractal image coding algorithm based on subdomain diagonal sum," *Microelectronics and Computer*, vol. 05, pp. 82–86, 2011.
- [15] S. Furao and O. Hasegawa, "A fast no search fractal image coding method," *Signal Processing: Image Communication*, vol. 19, no. 5, pp. 393–404, 2004.
- [16] Q. Chunqiang and W. Jicheng, "Application of quadtree theory in fractal image coding," *Computer Engineering and Applications*, vol. 23, pp. 61–63, 2007.
- [17] Z. Yunping and C. Chuanbo, "A fast fractal image compression algorithm based on new quadtree," *Small Computer Systems*, vol. 8, pp. 1465–1469, 2007.
- [18] B. D. Choi and S. J. Ko, "Split-and-merge based block partitioning for high efficiency image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 99, p. 1, 2018.
- [19] A. Ait-Kheddache and S. A. Rajala, "Texture classification based on higher-order fractals," in *Proceedings of the International conference on acoustics speech and signal processing*, pp. 1112–1115, 1988.
- [20] Z. Zhiliang, Z. Yuli, and Y. Hai, "Fast fractal image compression algorithm based on pixel distribution and triangle segmentation," *Journal of Computer Applications*, vol. 2, pp. 337–340, 2010.
- [21] Q.-M. Zheng, M. Zhao, F.-H. Wang, and J.-Z. Zhao, "Fractal compression algorithm based on irregular region segmentation and gray sorting classification," *Journal of China University of Petroleum*, vol. 3, pp. 169–173, 2014.
- [22] J. G. Conejeros, "A distributed K-means Segmentation Algorithm applied to Lobesia botrana Recognition," *Complexity*, vol. 2017, Article ID 5137317, 14 pages, 2017.
- [23] Y. C. Hum, K. W. Lai, and M. I. Mohamad Salim, "Multiobjectives bihistogram equalization for image contrast enhancement," *Complexity*, vol. 20, no. 2, pp. 22–36, 2014.
- [24] H. Chuanjiang, *Algorithm Research of Fractal Image Coding Technology*, Chongqing University, 2004.
- [25] G. Li, *Fractal Image Compression Coding*, vol. 347, Southwest Jiaotong University Press, 2010.
- [26] A. N. Backiam and R. Kousalyadevi, "Fast fractal image compression based on Fisher's classification scheme," in *Proceedings of the 2014 International Conference on Electronics and Communication Systems, ICECS 2014*, India, February 2014.
- [27] A. M. A. Banu, "Adaptive fractal image compression using PSO," *Procedia Computer Science*, pp. 338–344, 2010.

