Super-sampling method during decoding for fractal image compression

Jie He [1]*, Hui Guo [1]

[1] *School of Information and Electronic Engineering, Wuzhou University,Wuzhou, Guangxi,543002,China*

Received 1December 2014, www.cmnt.lv

Abstract

A super-sampling algorithm is presented to avoid the need for recoding when magnifying a reconstructed image after fractal image compression, thus making the fractal image compression method more practical. The sizes of and coordinates of range blocks and domain blocks in the decoder are adjusted according to the proportional relationship between the decoded initial image and the encoded image; then, the bilinear interpolation method is applied to complete the re-sampling process; and then, iterations are conducted. Thus, image magnification is directly completed in the decoding process. Image magnification with this algorithm does not lead to change in image texture or loss of image quality, and also image quality through super-sampling mainly depends on the size of sub-blocks during coding, according to experimental results. The method put forward in this paper can broaden the application scope of fractal image compression by effectively avoiding the excess time consumption resulting from re-decoding.

Keywords: Fractal Image Compression, Image Magnification, Super-sampling, Re-sampling, Bilinear Interpolation

1 Introduction

Fractal image compression is an innovative coding idea. The encoding process essentially involves calculating analogous areas on the image to overcome the limitation of the previous entropy coding. Meanwhile, the decoding process relies on an iterated function system (IFS) and can be executed at high speeds. In terms of compression effect, the method has a significantly higher compression ratio than other compression methods. However, fractal image compression has its limitations. Primarily, the matching process requires repeated traversals, which results in an excessively long encoding duration. Images with poor self-similarity or large sub-blocks can lead to a block effect in decoding after fractal image compression. These factors have restricted the wide application of the fractal coding method. Numerous studies on overcoming these limitations have been conducted. K. Jaferzadeh et al.[1] used pixel value space and 1D-DCT(1 Dimension-Discrete Cosine Transform) vector to determine the fuzzy clustering of sub-blocks and, accordingly, enhanced the encoding speed by 40 times without lowering decoding quality. Li [2] successively used the rotary inertia characteristics to locate block D (the neighborhood of block R) and effectively narrowed the search scope. Guo[3] and Sheng[4] improved the quadtree fractal coding method on the basis of the human visual system characteristics to reduce image distortion beyond the sensing ability of human eyes. However, the practicability of an image compression method should be judged from multiple aspects (e.g., compression ratio, encoding and decoding speeds, and quality of decoded image) and if convenient image operation algorithms can be derived. As a compression method in the spatial domain, fractal coding has great potential in supporting other image processing methods. Scholars have used the self-similarity describing thought of fractal coding to provide new solutions to application issues like image retrieval [5,6], image segmentation[7,8], image inpainting [9,10], image denoising [11,12], Pattern recognition [13], and digital watermarking [14,15]. Image magnification is a common and practical image processing operation. Employing super-sampling anti-aliasing (SSAA) technology, this paper applies image re-sampling to fractal image compression to reconstruct images with unreduced quality according to the given size requirements in an attempt to enhance the practicability of fractal coding.

2 Fractal image compression

Barnsley introduced the fractal theory into image compression in 1988 and proposed the IFS [16]. Jacquin, one of his students, further developed IFS into a partitioned IFS (PIFS) [17], which sets the application architecture of the fractal image compression technology. Fractal image compression uses the local self-similarity of the image to explore the relationship between the domain block and the range block, two types of image sub-blocks. For a 256×256 image, the image is primarily segmented into 8×8 non-overlapping range blocks, which constitute a range pool. The image is segmented into 16×16 overlapping domain blocks, with two pixels as the

* Corresponding author’s E-mail: hejie1213@126.com

step value, to constitute a domain pool. Figure 1 shows the segmentation process.

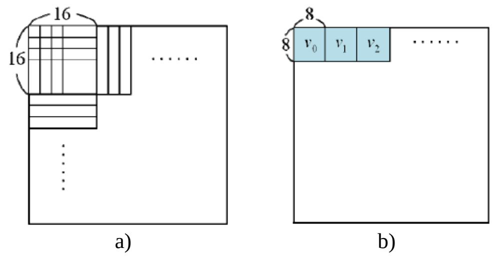


FIGURE 1 PIFS segmentation: a) Domain pool, b) Range pool

In coding, each range block should locate the best domain-block match in the domain pool. Formula (1) shows the PIFS affine transformation from the range block to the domain block.

$$W\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_k & b_k & 0 \\ c_k & d_k & 0 \\ 0 & 0 & p \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} i \\ j \\ q \end{bmatrix} \quad (1)$$

where W is the affine transformation of the domain block; x, y, and z are the abscissa, ordinate, and gray values of the range block, respectively; i and j are the coordinates of the domain block; p is the contrast translation coefficient; q is the brightness translation coefficient; a_i, b_i, c_i, and di are eight kinds of coefficient matrices that constitute isometric transformations as shown in Formula (2).

$$T_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, T_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, T_2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, T_0 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$
$$T_0 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, T_0 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, T_0 = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}, T_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (2)$$

These matrices represent eight kinds of homogeneous rotations in the spatial domain, where T_0 is the original image, T_1 is the 90º clockwise rotation, T_2 is the 180º clockwise rotation, T_3 is the 270º clockwise rotation, T_4 is the horizontal flip, T_5 is the vertical flip, T_6 is the back-diagonal flip, and T_7 is the diagonal flip as shown in Figure 2.

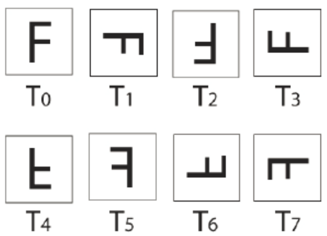


FIGURE 2 Eight kinds of isometric transformations

After coding, (i, j), p, q, and k, the coefficients that represent the rotation, are recorded as fractal codes. The PIFS decoding process involves constant iterations and gradual convergence to obtain the reconstructed image. First, select any image as the initial image, read the fractal code, and use Formula (3) to obtain each range block:

$$v = pu + q \quad (3)$$

Use u, the 16×16 sub-block that the coordinates (i, j) on the initial image correspond to, as the domain block. Downsize it to 8×8 and transform it based on the flip coefficient k, and adjust its contrast and brightness through p and q. The new 8×8 block is the desired domain block v. Repeat the process, and after a certain number of iterations, a high-quality reconstructed image is obtained.

3 Image re-sampling

In image resizing, pixels tend to fall on non-integer coordinates because of scale change. Consequently, the image re-sampling technology is employed, which uses an interpolation method to arbitrarily resize images without causing texture changes. Common image re-sampling methods include nearest-neighbor interpolation, bilinear interpolation, and cubic convolution interpolation.

The nearest-neighbor interpolation method simplifies the calculation by regarding the gray value of the nearest neighbor of the point to be sampled as the gray value of this point among the four adjacent pixels. The method neglects the influence of other adjacent pixels, which causes apparent discontinuity in the gray values of the image after re-sampling, and serration or block effects are often observed on the magnified image.

Bilinear interpolation involves performing linear interpolations in two directions by using the gray values of four adjacent surrounding points to obtain the gray value of the point to be sampled. The corresponding weights are determined according to the distance between the point to be sampled and the adjacent points to calculate the gray value of the point to be sampled. Figure 3 shows this principle.

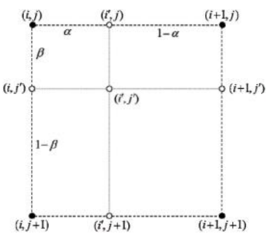


FIGURE 3. Bilinear interpolation

(i', j') is the new interpolated value after image magnification, and the coordinates of the four neighboring pixels are (i, j), ($i+1$, j), (i, $j+1$), and ($i+1$, $j+1$). The horizontal weight α and vertical weight β are evaluated according to the coordinates of the four points. The gray value of point (i', j') is obtained using Formula (4), where f represents the gray value.

$$f(i', j') = (1-\alpha)\times(1-\beta)\times f(i, j) + \alpha\times(1-\beta)\times f(i+1, j) + (1-\alpha)\times\beta\times f(i, j+1) + \alpha\times\beta\times f(i+1, j+1) \quad (4)$$

Figure 4 shows the principle of the cubic convolution interpolation, which involves the use of 16 adjacent pixels surrounding the sample point. This method produces resized images with higher quality than those produced through bilinear interpolation. However, the method is computationally complex.

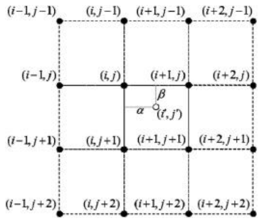


FIGURE 4 Cubic convolution interpolation

Among the three methods, bilinear interpolation has the most balanced performance. Bilinear interpolation is generally employed to guarantee both image quality and execution speed. Therefore, this paper uses bilinear interpolation to address the interpolation issue during image amplification in fractal decoding.

4 Super-sampling for fractal decoding

The decoding process after fractal image compression is independent from the resolution. A similar reconstructed image can still be acquired after multiple iterations even if an image with the same size as the original image but with different texture and resolution is used as the initial image. If the size of the decoded image needs to be changed without losing image quality, the original image with the new size should be recoded. When the image is magnified, the encoding time increases accordingly.

The SSAA technology is a common image-edge softening technology. First, the image is mapped into the cache and amplified, and the pixels of the amplified image are re-sampled. Generally, two or four adjacent pixels are mixed to generate the final pixel so that each pixel bears the characteristics of neighboring pixels, which ensures a smooth color transition on the image edge. The image size is restored, and the original image is replaced. This series of operations not only smoothen and amplify the original image, but also produce a new image with enhanced details but the same size. The super-sampling process for fractal image compression is realized using SSAA. Only original images with certain sizes are decoded. At the decoder, the sub-blocks of the decoded original image are adjusted according to the proportional relationship to magnify the image. The image is further smoothened through re-sampling. The image is further downsized to the size of the original image for a detailed reconstructed image.

In the decoding process, an image with the same size as the original image is usually used after classic fractal image compression to achieve a resemblance between the reconstructed image and the original image in texture (Figure 5).

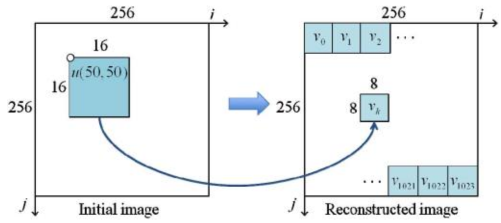


FIGURE 5 Decoding after fractal image compression

An initial image with a larger size than the original image is used in decoding to make the reconstructed image bigger than the original image and avoid recoding. The original fractal code is still used for decoding, and the size disparity between the initial image and the original image causes a difference in the total number of range blocks and domain blocks between the two images. The proportional relationship between the original image and the initial image should first be acquired as shown in Formula (5).

$$P_w = I_w / O_w$$
$$P_h = I_h / O_h \tag{5}$$

Where O represents the original image, I represents the initial image in decoding, w represents image width, and h represents image length.

The sizes of the range block and the domain block of the decoded image are calculated with P_w and P_h, as shown in Formula (6).

$$D_f \times D_f = (P_w \times D_o) \times (P_h \times D_o)$$
$$R_f \times R_f = (P_w \times R_o) \times (P_h \times R_o) \tag{6}$$

The decoding after fractal image compression requires the use of the location of point (i, j) recorded in the fractal code. Based on (i, j), the most analogous domain block is obtained. The domain block is downsized, flipped, and adjusted (in contrast and brightness) to form a new block, which is placed in the location of the corresponding range block. Given that the decoded initial image is larger than the original one, (i, j) should also be adjusted as shown in Formula (7).

$$(i_f, j_f) = ((P_w \times i), (P_h \times j)) \tag{7}$$

Moreover, pixels usually fall on non-integer coordinates because of image magnification. Linear interpolation is also applied for interpolation, and new pixels and their gray values are obtained using Formula (4). Figure 6 shows the super-sampling process for fractal decoding.

Assuming that the original image size is $O_w \times O_h$, the algorithm of super-sampling for fractal decoding is as follows:

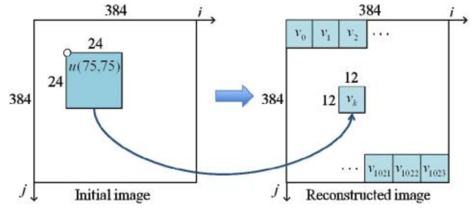


FIGURE 6 Super-sampling for fractal decoding

Input: $I_w \times I_h$ decoded initial image I, and fractal code [(i, j), p, q, k], where (i, j) is the location of the domain block, p is the contrast coefficient, q is the brightness coefficient, and k is the flip coefficient.

Output: $I_w \times I_h$ reconstructed image F.

Step 1: Calculate the proportion coefficients P_w and P_h of the initial image and the original image, respectively, using Formula (5). Afterwards calculate $D_f \times D_f$ (size of the domain block) and $R_f \times R_f$ (size of the range block) in decoding using Formula (6).

Step 2: Define areas R and D, which have the same sizes as the initial image I, using R for preserving the image generated during iterations and D for generating codes.

Step 3: Place I in area D and initialize it.

Step 4: Segment R into sub-blocks of size $R_f \times R_f$ in a non-overlapping way to obtain the decoding domain pool. For each R block, locate the D block of size $D_f \times D_f$ at point [($P_w \times i$), ($P_h \times j$)] in area D according to the fractal code [(i, j), p, q, k]. Decrease its size to R_f x R_f, flip it, and adjust its contrast and brightness according to p, q, and k. After decoding all R blocks, collage them and complete the re-sampling according to Formula (4). Finally, obtain a one-iteration image.

Step 5: Copy the image in area D and place it in area D.

Step 6: Stop if the preset iterations are complete; otherwise proceed to Step 4.

Step 7: Output the constructed image.

5 Experimental results and analysis

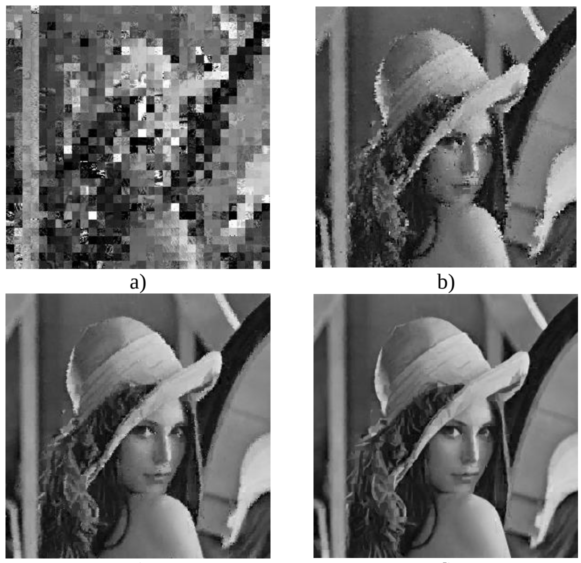
The experimental environment consists of Intel Core i3-3110 processor (2.4 GHz), 2G memory, Microsoft Windows 7 Professional operating system, and MATLAB 7.0 integrated development environment. 256×256 Lena is used as the original image for encoding, and the sizes of the domain block and the range block are 16×16 and 8×8.In the decoding process, 288×288 Baboon is used as the initial image. The range block and the domain block at the encoder should be magnified by 9/8 times. Figure 8 shows the test result after 15 iterations.

When reconstructing the image through super-sampling in the decoding process, Figure 8 demonstrates that the image quality gradually becomes analogous to the original image as the number of iterations increases. However, the texture becomes consistent with that of the original image after image magnification.

Some block effects are observed in the facial details, which are not caused by super-sampling. For coding methods that use image sub-blocks as the basic representation unit, large differences in the pixel values of two adjacent blocks easily cause a block effect in decoding (i.e., the greater the block size, the higher the possibility of a block effect). If the sizes of the range block and the domain block are reduced, the image effect is also significantly improved in super-sampling. 256x256 Lena is used as the initial image for encoding, with range blocks of 8×8, and 4×4 to generate three groups of fractal codes. 384×384 Baboon is used as the initial image for decoding, and the image is reconstructed through 15 iterations using the three groups of fractal codes (Figure 9).

a) b)

FIGURE 7 Original image and initial image: a) 256×256 Lena, b) 288×288 Baboon



a) b) c) d)

FIGURE 8. Super-sampling process of fractal decoding: a) 1st iteration, b) 3rd iteration, c) 4th iteration, d) 15th iteration

a) b)

FIGURE 9. Super-sampling based on range blocks of different sizes: a) 8×8 range block, b) 4×4 range block

The size adjustment of the range block significantly improves the overall quality of the reconstructed image after decoding and super-sampling. The square effect is significantly weakened with the downsizing of the range block, and the size transformation from 8×8 to 4×4 shows the most prominent visual effect changes, particularly in the hair and facial features.

The effect of super-sampling for fractal decoding is best demonstrated by comparing the resulting image with the original image through re-sampling, magnified by the same ratio. First, the 256×256 initial image Lena is magnified to 384×384 using bilinear interpolation (Figure 10).

FIGURE 10 Original image, linear interpolation, 384×384 Lena

Figure 10 shows that although the whole image has soft and smooth edge transitions, the image is obviously blurred compared with the original 256x256 Lena and Figure 9(b). According to edge smoothness and texture complexity, the sub-blocks of three different parts extracted from Figures 9(b) and 10 are compared. The three sub-blocks, whose positions are marked with a white border in Figure 11 and are all 36×36 in size, are magnified to 600% for comparison (Figure 12).

FIGURE 11 Positions of the extracted sub-blocks

The comparison of the sub-block details at the hat edge in Figures 12(a) and 12(b) shows that fractal super-sampling and bilinear interpolation has a similar effect in the smooth part. Figures 12(c), 12(d), 12(e), and 12(f) reflect that, for cases with continuous dramatic pixel-value changes within a small scope or complex local texture, the image from fractal super-sampling has an accurate reflection of changes in gray scale and texture, although the block effect is slightly heavier than the smooth part, causing a stiff edge transition. Therefore, Figure 9(b) may have a clear outline in terms of overall visual effect but is not as smooth as Figure 10.

Only a minor visual difference is observed between Figures 9(b) and 10 when downsized to 256×256 (Figure 13). PSNR is the objective standard of evaluating image quality, and it requires that image sizes involved in the operation should be consistent. Therefore, using the 256x256 original image Lena as a benchmark, Figures 9(a) are also downsized to 256×256 to calculate the PSNRs of the four downsized images (Table 1).

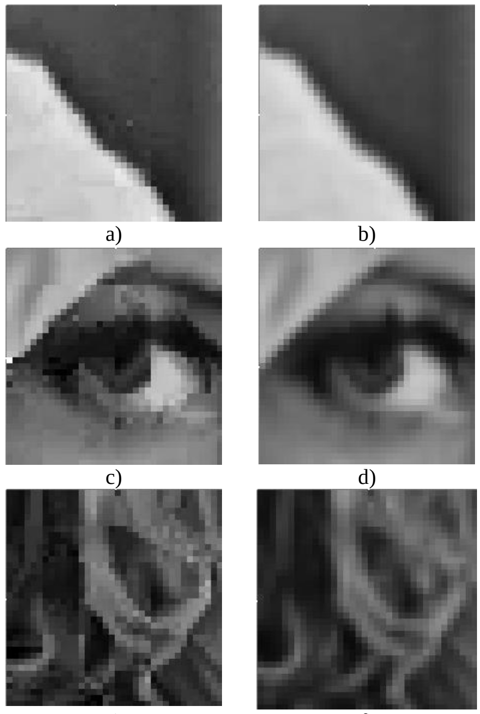


FIGURE 12 Detailed comparison of sub-blocks magnified to 600%:a) FIC super-sampling, 4×4 range block, hat, b) Original image, linear interpolation, hat, c) FIC super-sampling, 4×4 range block, eyes, d) Original image, linear interpolation, eyes, e) FIC super-sampling, 4×4 range block, tassel, f) Original image, linear interpolation, tassel

(a) Image 9(b) shrunk to 256×256

(b) Image 10 shrunk to 256×256

FIGURE 13. Shrunken images of Figure 9(b) and 10

TABLE 1 PSNR of Figures 9 and 10 shrunk to 256×256

Image name	PSNR(dB)
The Figure 9(a) shrink to 256×256	33.5925
The Figure 9(b) shrink to 256×256	38.6126
The Figure 10 shrink to 256×256	40.8204

6 Conclusion

In fractal image compression, the decoded initial image, which has the same size as the original image, is usually chosen for image reconstruction. The size of the reconstructed image cannot be changed without recoding a previous image with a different size, which results in a long encoding duration and limits the wide application of the method. This paper devises a mechanism of adjusting the sizes of sub-blocks at the decoder according to the initial image. Based on bilinear interpolation, a super-sampling algorithm is proposed for decoding after fractal image compression. According to the experimental results, the algorithm enables image magnification without changing image texture or damaging image quality and broadening the application scope of fractal image compression. In our future work, this method will be combined with the sub-block segmentation method, which has strong self-adaptability to gray value and texture, to further alleviate the block effect on the reconstructed image after re-sampling.

Acknowledgments

This paper is financially supported by Guangxi Natural Science Foundation Program (2013GXNSFBA019275, 2013GXNSFBA019276) and Guangxi University of Science and Technology Research Program(2013YB227, 2013YB228).

References

[1] K. Jaferzadeh, K. Kiani, S. Mozaffari 2012 Acceleration of fractal image compression using fuzzy clustering and discrete-cosine-transform-based metric *IET Image Processing* **6** (7) 1024-30

[2] Li G P, Yang J , Chen Y H 2013 Fast fractal image encoding algorithm based on improved moment of inertia feature *Computer Engineering and Applications* **49**(24) 144-8 (in Chinese)

[3] Guo H , Zheng Y P, He J 2012 A new HVS-Based fractal image compression algorithm *Lecture Notes in Electrical Engineering* **138** 753-9

[4] Sheng Z 2009 Fractal color image coding algorithm based on human visual system *Journal of Computer Applications* **29**(12), 244-6 (in Chinese)

[5] Zhang L B, Xi L F 2008 New image retrieval method based on entropy and fractal coding *Computer Engineering and Applications* **44**(9), 203-5

[6] Wang X Y 2009 A fast fractal coding in application of image retrieval *Fractal* **17**(4), 441-50 (in Chinese)

[7] Ida T, Sambonsugi Y 1995 Image segmentation using fractal coding *IEEE Transactions on circuits and systems for video technology* **5**(6), 567-70

[8] Ida T, Sambonsugi Y 1998 Image segmentation and contour detection using fractal coding *IEEE Transactions on circuits and systems for video technology* **8**(8), 968-75

[9] Li J J, Zhang C M, Fan H, Yuan D 2010 Image inpainting algorithm based on fractal theory *Acta Electronica Sinica* **38**(10), 2430-5

[10] Yao Y, Tang J, Luo B 2012 Image inpainting algorithm based on iterative fractal decoding and edge expansion *Computer Engineering* **38**(2) 218-20 (in Chinese)

[11] Mohsen G, George H F, Edward R V 2006 Fractal-Wavelet image denoising revisited *IEEE Transactions on Image Processing* **15**(9) 2669-75

[12] Wang Z W, Li S Z 2014 Adaptive fractal wavelet image denoising based on multivariate statistical model *Chinese Journal of Computers* **37**(6) 1380-8 (in Chinese)

[13] Han J S 2013 Automatic recognition of M-Star spectral based on fractal coding *Spectroscopy and spectral analysis* **33**(8) 2259-62 (in Chinese)

[14] Huang X Q, Yu S L 2012 Watermarking method on fractal coding for image authentication *Journal of Image and Graphics* **17**(2) 183-90 (in Chinese)

[15] Nie D C, Zheng H Y 2014 A digital watermarking algorithm for color Images based on orthogonal fractal coding method *Computer & Digital Engineering* **42**(1) 129-33 (in Chinese)

[16] Michael F B, Alan D S 1988 A better way to compress images *Byte* **13**(1) 215-33

[17] Arnaud E J 1992 Image coding based on a fractal theory of iterated contractive image transformations *IEEE Transactions on Image Processing* **1**(1)

Authors

Jie He , 1982.2,Yiyang, Hubei Province, P. R. China

Current position, grades: the associate professor of School of Information and Electronic Engineering, Wuzhou University, China.
University studies: He graduated from Guangxi Normal University in 2011, received a master's degree in engineering.
Scientific interest: His research interest fields include Image Processing, Virtual Reality.
Publications: more than 20 papers published in various journals.
Experience: He has teaching experience of 11 years, has presided over 9 scientific research projects.

Hui Guo , 1981.9, Wuzhou, Guangxi Province, P. R. China

Current position, grades: the associate professor of School of Information and Electronic Engineering, Wuzhou University, China.
University studies: She graduated from Guangxi Normal University in 2011, received a master's degree in engineering.
Scientific interest: Her research interest fields include Image Processing, Embedded System.
Publications: more than 20 papers published in various journals.
Experience: She has teaching experience of 11 years, has presided over 7 scientific research projects.