Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 4865808, 10 pages
https://doi.org/10.1155/2022/4865808

Research Article

Texture Image Compression Algorithm Based on Self-Organizing Neural Network

Jianmin Han

School of Computer Engineering, Henan Economic and Trade Vocational College, Zhengzhou, Henan 450046, China

Correspondence should be addressed to Jianmin Han; hanjm@peczzu.edu.cn

Received 18 January 2022; Revised 21 February 2022; Accepted 28 February 2022; Published 10 April 2022

Academic Editor: Shahid Mumtaz

With the rapid development of science and technology, human beings have gradually stepped into a brand-new digital era. Virtual reality technology has brought people an immersive experience. In order to enable users to get a better virtual reality experience, the pictures produced by virtual skillfully must be realistic enough and support users' real-time interaction. So interactive real-time photorealistic rendering becomes the focus of research. Texture mapping is a technology proposed to solve the contradiction between real time and reality. It has been widely studied and used since it was proposed. However, due to limited bandwidth and memory storage, it brings challenges to the stain dyeing of many large texture images, so texture compression is introduced. Texture compression can improve the utilization rate of cache but also greatly reduce the pressure on data transmission caused by the system, which largely solves the problem of real-time rendering of realistic graphics. Due to the particularity of texture image compression, it is necessary to consider not only the quality of texture image after compression ratio and decompression but also whether the algorithm is compatible with mainstream graphics cards. On this basis, we put forward the texture image compression method based on self-organizing mapping, the experiment results show that our method has achieved good results, and it is superior to other methods in most performance indexes.

1. Introduction

Due to the limitation of software and hardware of computer system, data compression of digital graphics and images is always the focus of computer graphics research, which has important value in practical application. With the rapid development of science and technology, human beings have gradually stepped into a brand new digital era [1–3]. Digital technology affects every aspect of people's life, among which virtual reality technology has brought people a brand new immersive experience. Its research began in the 1960s. In the early 1990s, it became a relatively complete discipline and aroused great attention. Let users obtain better virtual reality experience; virtual images must have enough sense of reality [3–7]. Therefore, interactive real-time photorealism in virtual reality scenes has become the focus of research. Texture mapping is a technology proposed to solve the contradiction between real time and reality. Texture can obtain realistic details without increasing the complexity of geometric model. Since it was proposed, it has been widely studied and applied [8].

The image has spatial redundancy between adjacent pixels and frequency domain redundancy after DCT. Since the human eye is not sensitive to some signals, the image still has visual redundancy. Removing this redundancy can greatly reduce the storage space required by the image. Image compression technology can reduce the amount of data required for picture preservation and speed up the rate of picture transmission under the premise of keeping the visual quality of pictures basically unchanged. Therefore, image compression has important theoretical and application value and plays an important role in promoting the development of the whole information age [9]. The explosion of mobile Internet users makes many developers transplant applications from PC to mobile terminals. Under the traction of the huge demand for the development of virtual reality, it is imperative for 3D scene visualization to be applied to mobile terminals. However, 3D scene virtual and visualization technology for mobile terminals is still a hot spot and difficulty in current research. Compared with personal desktop computers, mobile terminals of mobile phones have small screen, low resolution, small memory capacity, and limited power [10, 11]. Compared with graphic-based 3D scene visualization methods, the image-based visualization modeling method is suitable for mobile applications due to the low requirements on system hardware capability during rendering and short modeling cycle. For example, texture image data still accounts for 50% to 70% of the massive rendering data. Therefore, massive texture image data still restricts the development of 3D scene visualization in mobile terminals. In order to reduce the amount of texture data transmission, the study of texture image compression technology is extremely urgent [12–14].

Different from general two-dimensional images, texture images contain not only 2D texture color information but also implicit geometry information corresponding to 3D models. Therefore, if geometric information is not taken into account when selecting salient regions, the visually important regions of the texture map cannot be correctly obtained. However, there is no method that considers geometric information in defining the saliency region of texture image. Texture mapping achieves texture image rendering through a large amount of data transmission and calculation, so it puts forward higher requirements on hardware and computing capability of graphic images [15–17]. Texture image rendering is an indispensable key technology to construct 3D scene visualization. After years of development, 3D scene visualization technology has shown great application in the field of application and research. However, even with the rapid development of graphic image hardware and computing capability today [18, 19], the application of 3D scene visualization technology to mobile terminals also needs to overcome the problems of data transmission and computing power consumption [20–24].

This paper consists of five parts. The first and second parts give the research status, background, and related work. The third part is the texture image compression algorithm based on self-organizing neural network. The fourth part shows the experimental results. The experimental results of this paper are introduced and compared and analyzed with relevant comparison algorithms. Finally, the fifth part concludes the full paper.

2. Related Work

In the past decades, the field of texture image compression has been widely studied by scholars at home and abroad. This part briefly introduces the origin and development process of texture image compression and gives a brief introduction to some classical texture compression algorithms [25]. Prisma oil painting style change software is available, which uses the Deep Web to transform natural images into art images in the style of oil paintings, which has been downloaded nearly 10 million times in just a few weeks, extending deep learning from science and industry to aesthetics. Image compression is divided into lossy compression and lossless compression. Lossless compression algorithm removes the statistical redundancy of images, and the compression ratio is roughly between 5 : 1 and 2 : 1. Farid et al. [26] proposed a simple format for power image compression, called block clipping coding technique, although it is not in order to solve the texture image compression, but a lot of texture image compression algorithms draw lessons from the idea of it, so that each pixel only needs 2 bits. Ernawan et al. [27] published a paper called “Self-Compressed Texture-Based” in which they listed four texture compression features to illustrate the differences between texture compression and normal image compression methods. Pan et al. [28] used a traditional method called Vector Quantization, or VQ, which compresses the texture image to two or one bit per pixel. However, the vector quantization method, like the palette texture method, requires additional memory access to determine which color to use, which is relatively inefficient, making it unsuitable for the side dyeing pipeline of high performance computer graphic. Penney et al. proposed a new idea fundamentally different from the above method. They obtained two low-resolution texture images from the original texture image and stored them. In the process of decompression, the color of the final texture pixel was obtained by magnifying the two low-resolution images through bilinear interpolation. Armi and Fekri-Ershad [29] concluded advantages and disadvantages of well-known texture image descriptors in the result part. But, the above methods are mainly concerned with recognition performance, computational complexity and antinoise, rotation, etc. A brief review is also made on the common classifiers used for texture image classification.

In the texture compression of wavelet decomposition, Qiao et al. [30] introduced wavelet into the field of texture compression. It is a method of fixed compression ratio proposed based on the energy tightening characteristics of transform. The coefficients of wavelet transform are quantified and then encoded and stored by the standard block compression method. The focus of this approach is to ensure that simple and fast decoding is achieved [31]. The idea is a new one, and it works well. On the same year, this method was formally adopted as an extended standard. Its function is very powerful, supports a range of compression ratios, is basically the most cutting-edge result in the field of texture image compression, but the new method is still to take into account its hardware implementation cost which is relatively high, and is not the most mainstream method, but the future development prospects are very good. With the development of the Internet market, Baidu Map and AutoNavi map have also opened up the market for 3D scene visualization, thus 3D scene visualization has become more and more common into the public’s vision, and scholars’ research has never been interrupted. Liu et al. [32] proposed to construct procedural syntax, transmit unit texture, and compress redundant texture image data by extracting repeatability and symmetry of 2D texture image. Because the drawing speed is slow three-dimensional scene rendering large amount of calculation, the visualization method based on graphics than most is limited to use desktop for personal desktop, mobile terminal has a small screen resolution, and low memory capacity small data calculation and graphics processing ability is low. Therefore, in the mobile network bandwidth calculation under the premise of limited memory resources, the graphic-based 3D scene visualization method is not suitable for mobile terminal development. In recent years, image-based scene visualization methods have been studied and applied more and more. Some scholars focused on the mosaic and presentation of panoramic images, while some authors study the stretching of panoramic texture images. Deformation and other operations increase the depth of information, but in practical development and application, little attention is paid to the compression of texture image data.

From the above analysis, we know that the above methods have studied the texture image compression widely. However, some problems still exist. For example, no scholar has applied the self-organizing neural network to this field till now, so the research here is still a blank, which has great theoretical research and practical application value for logistics enterprises.

3. Texture Image Compression by Self-Organizing Neural Network

3.1. Texture Image Compression Process. For example, we tend to draw high level details of characters in movies, such as pore less attention and more attention to the character's facial features. The whole system of the method is given in Figure 1, and the details are as follows: (1) finding the offset threshold; (2) selecting feature points; (3) feature points are iteratively obtained until a satisfactory quantity set is obtained; (4) obtaining texture image feature points 5.

For the progressive grid, not only faces but also edges have tree structure, as shown in Figure 1, because the actual and predicted offset positions of vertices in the grid correspond to each vertex, and each vertex corresponds to an edge in the upper-resolution grid with four edges of one resolution and four edges of the lower-resolution grid Therefore, quadtree of offset can be constructed by constructing quadtree of edge.

3.2. Self-Organizing Neural Network. Self-organizing map (SOM) is a neural network that can perform unsupervised learning and clustering of data. Its idea is simple, and it is essentially a neural network with only an input layer and a hidden layer. Armando et al. [33] believed that neurons in different regions of the space had different labor divisions. After the original data are input into the neural network, the network is divided into different response regions according to the response characteristics of each input mode. In the SOM neural network, the mapping relationship between the input layer and the competition layer is shown in the following formula:

$$f\colon M \Rightarrow N, \tag{1}$$

where M is the set of input vectors, whose dimension is equal to the dimension of the input vector; N varies with the output grid. For example, in the SOM neural network of a two-dimensional grid, N is a two-dimensional plane.

SOM neural network learning algorithm includes three processes: competition, cooperation, and renewal. Competition: the neuron in the network calculates its discriminant function value for each input mode according to the discriminant function, and the neuron with the largest discriminant function value is selected as the winning neuron. Cooperation: the winning neuron influences the spatial location of the excitatory neuron. Update: the excitatory neuron uses its weight adjustment to enhance its response to future similar mode inputs. Calculation of Euclidean distance between weight vector and input vector at mapping layer: at mapping layer, the formula is used to calculate the distance between input sample and each output neuron *J*, as shown in the following formula:

$$d_j = \left\| X - W_j \right\| = \sqrt{\sum_{i=1}^{m} \left(x_i(t) - w_{ij}(t) \right)^2}. \tag{2}$$

In the formula, w_{ij} represents the weight between i neuron at the input layer and j neuron at the output layer, and the information of all output vectors is stored in each weight vector, that is to say, the weight coefficient represents the relationship between failure symptom and failure mode. The size of input vector and weight vector should keep the same to make the network works.

$$d_k = \min_j \left(d_j \right). \tag{3}$$

The modified formula is shown as follows:

$$w_{ij}(t+1) = \begin{cases} w_{ij}(t) + \eta(t)\left[x_i - w_{ij}(t)\right], & \left(i \in s_k(t)\right), \\ w_{ij}, & \left(i \notin s_k(t)\right), \end{cases} \tag{4}$$

and usually $\eta(t)$ to be variational, the details can refer to [33]

$$\eta(t) = 0.2\left(1 - \frac{t}{1000}\right), \tag{5}$$

or

$$\eta(t) = \frac{1}{t}. \tag{6}$$

We have

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \alpha_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2, \tag{7}$$

and

$$a_t^2 - \eta_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i a_{t-i}^2 + \sum_{j=1}^{q} \beta_j \left(a_{t-j}^2 - \eta_{t-j} \right). \tag{8}$$

So, calculating the output,

$$\mathrm{O}_k = f\left(min \left\| X_j - W_j \right\| \right). \tag{9}$$

To ensure that variance is positive, that is, variance exists and is finite [34],

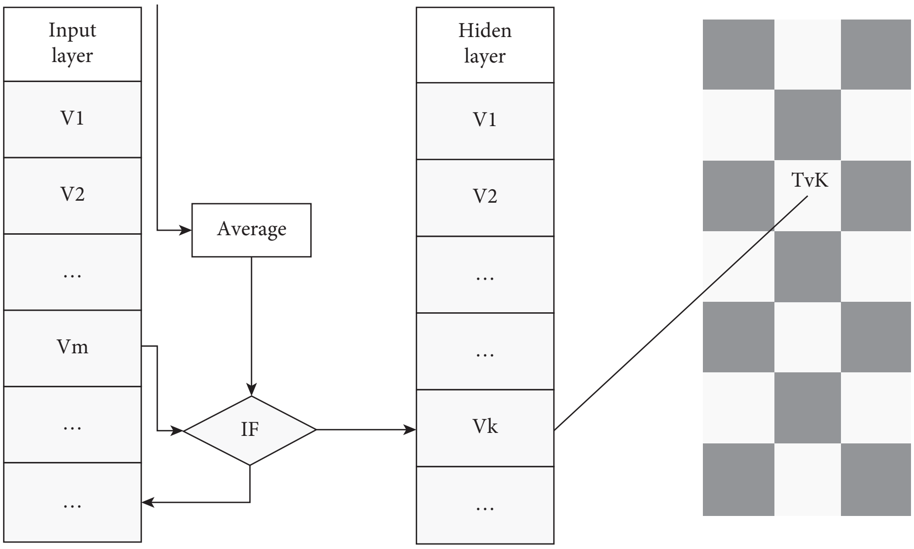


Figure 1: The image compression diagram of SOM networks.

$$\sum_{i=1}^{\max(p,q)} \left(\alpha_i + \beta_i\right) < 1. \tag{10}$$

Once the active node is found, we also want to update neighboring nodes, assigning them an update weight

$$T_{j,I(x)} = \exp\left(-\frac{S_{j,I(x)}^2}{2\sigma^2}\right), \tag{11}$$

and then it is time to update the parameters of the node by gradient descent:

$$\Delta w_{ji} = \eta(t) \cdot T_{j,I(x)}(t) \cdot \left(x_i - w_{ji}\right). \tag{12}$$

(11) updates neighboring nodes and (12) updates the parameters of the node.

A two-dimensional SOM neural network structure is given as Figure 2, which contains the following four parts: (1) array of processing units: receive input signals and form their discriminant functions. (2) Comparison and selection mechanism: compare the discriminant functions and mark the neuron node with the maximum function output value as the winning unit. (3) Local interconnection: when the selected neuron is excited, it will also affect its neighboring neurons. (4) Adaptive process: in this process, the parameters of the excited neuron are constantly modified so that the output value of its discriminant function increases.

4. Experimental Results and Analysis

4.1. Introduction to Experimental Environment and Data Set. In terms of the selection of data sets, this topic selects Raise and ImageNet data sets as the network training set Raise data set, consisting of 4000 high-resolution and basically lossless images. The Raise data set is composed of 3 different cameras, from Europe 80 multiple locations, including outdoor, indoor, and natural; the size of a picture in a Raise data set is about 4000. 3000 ImageNet data set includes thousands of object categories, and each category contains thousands of pictures.

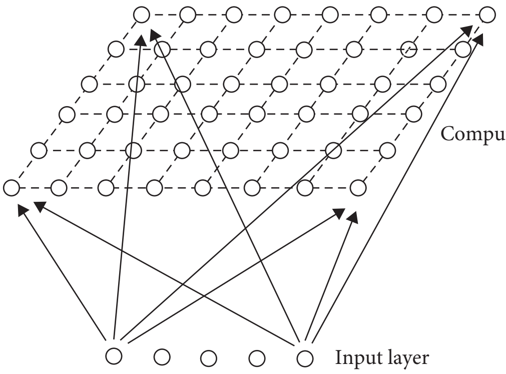


Figure 2: Two-dimensional SOM neural network structure.

When the Raise data set is used in this project, all 4000 training images are used to randomly select 50 categories for the ImageNet data set, and then 200 images are randomly selected from these categories to obtain a total of 10000 images from the ImageNet data set. In the process of random selection, images with overexposure and dark size and too small size should be filtered out to avoid a large difference between the image scale exposure of the training set and common natural images.

The hardware environment of the platform was E3-1230 V3 CPU, 32 GB memory, GTX Titan 12 GB video memory, and the PyTorch deep learning framework was used in the Ubuntu 14.04 system experiment. The network uses Adam algorithm to update the gradient and iterates 100 epochs, and the learning rate decayed from 41E in the form of log function.

4.2. Experimental Results Analysis. Taken early parameter adjustment: in the process of photo collection, due to the particularity of wide-angle lens, its perspective effect is stronger, edge images are prone to distortion. Therefore, the short focal length lens is not very appropriate in the process of use. As for the choice of focal length, different focal lengths can be selected in different scenes, but one principle must be guaranteed: the pictures are in the same scene. In addition to the shooting angle, all parameters such as focal length, ISO resolution, and aperture speed need to be fixed in order to fix the deformation of the scene and postprocessing, which is shown in Figure 3.

Taken early parameter adjustment: in the process of photo collection, due to the particularity of wide-angle lens, its perspective effect is stronger, edge images are prone to distortion. In addition to the shooting angle, all parameters such as focal length, ISO resolution, and aperture speed need to be fixed in order to fix the deformation of the scene and postprocessing. Postshooting processing: use the PTGui panorama making tool to put all the photos of the same scene into PTGui for processing and output a rough spherical panorama according to the steps of synthesis and adjustment. Then, put the spherical panorama into Krpano and use the batch processing file of Covert SPHERE to CUBE Droplet to form the storage structure of the CUBE panorama model required in this paper.

In this paper, four storage surfaces of cubic panorama are used. According to the principle of scene recognition by the human visual nervous system, the effective area of interest, namely, the area with high complexity, can be quickly recognized according to the scene recognition by the human eye. According to the division of the three levels of areas, each region is divided. The corresponding image resolution is also divided into three levels. The two regions with the largest information entropy, namely, the first level region, are rendered with the highest resolution texture image. The texture image resolution of the two regions in the middle of the order of information entropy size, namely, the second-grade region, is reduced to 0.5 times of the original length and width of the texture image. The length and width of the fingerprint image in Figure 4 were reduced to 0.25 times of the original image in the two regions with the minimum information entropy.

The specific compression effect is shown in Figure 4. Specifically, the image information entropy is used to classify and compress three texture images to achieve multilayer texture image rendering on the same level. This method not only reduces the compression amount of texture data through image scaling but also solves the problem of large computation amount of texture image rendering.

Before using a certain experimental environment and related parameter settings, we give the experimental results of the corresponding database for each encoder, including root mean square error and running time. Besides running time, all values are the average values of all images in the whole database. Red means the method with the shortest running time, and blue means the method with the best quality of decompressed images. For these measures, a smaller RMSE means a smaller error. The higher the PSNR value is, the higher the signal intensity is and the lower the noise is, that is, the better the image quality is. The shorter the run time, the more efficient it is, which is given in Figure 5.

In order to further validate this article, based on the SOM network, a panoramic view of the universal applicability of texture image compression method, the images of acquisition can be divided into two groups, the first set of data is indoor scene and a second group of data is the natural scenery as the main scene (given in Figure 5), and respectively for the two scenarios to experiment, calculate each scene data before and after compression Figure 6.

In scene 1, the total data of texture image in each area before compression is 3511 KB, and after compression, the total data of texture image in each area is 2457 KB, and the compression rate can reach 30.01%. In scene 2, the total amount of texture image data in each area before compression is 3481 KB, and after compression, the total amount of texture image data in each area is 2335.3 KB, and the compression rate can reach 32.91%. The compression rate will be slightly adjusted with different scenes, but the overall compression rate of the scene is above 30%.

The SOM-based panoramic texture image compression method in this paper was used for two different scenes above. From the compression rate, it can be seen that this method is suitable for all different scene compressions, and the compression rate is at least 30%. The 8–12% interest rate range is back in the ascendant. In this experiment, the texture image data is compressed by SOM-based texture image data compression algorithm and compared with the panoramic texture data without this compression method.

It can be seen from Figure 7 that the ICA model can achieve the fastest encoding speed and is better than the fast encoder of some existing DXT1 encoders in terms of quality. The iterative method based on K-means has achieved good results and is faster than the compared algorithms under the condition of similar encoding quality. In addition, increasing the number of iterations can improve the coding quality to a certain extent, but the coding time will increase a lot at the same time. In order to better compare the relationship between measurement indexes of different methods, we gave the line graph corresponding to the operation results of different methods on H databases. Since there are too many methods, it is difficult to draw all indexes on the same graph. We selected the PSNR values that are most used in texture compression quality evaluation for display.

As shown in Figure 8, the X-axis represents the original image in the database, and the Y-axis represents the ICONS in the PSNR value graph, representing different encoder algorithms. We can see from the figure that our fast coding method Lsq3d achieves good results, which is higher in quality than the other fast modes, and the iterative method based on K-means is in the middle to the upper level of quality. Finally, our combination method achieves the highest PSN values in every image.

As can be seen from the comparative analysis of the above experimental effects. Our method has achieved good

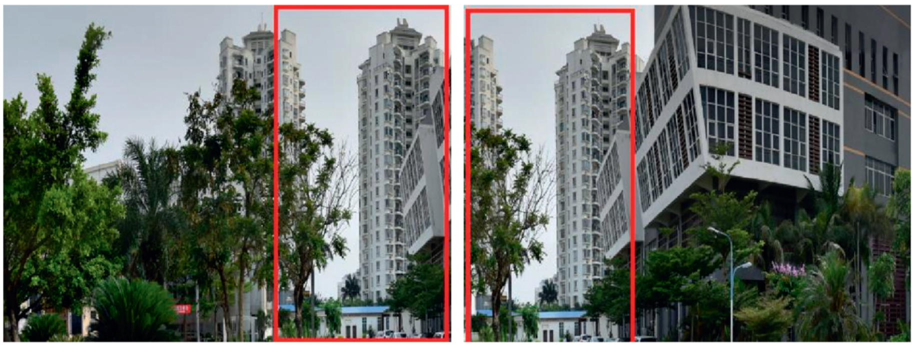
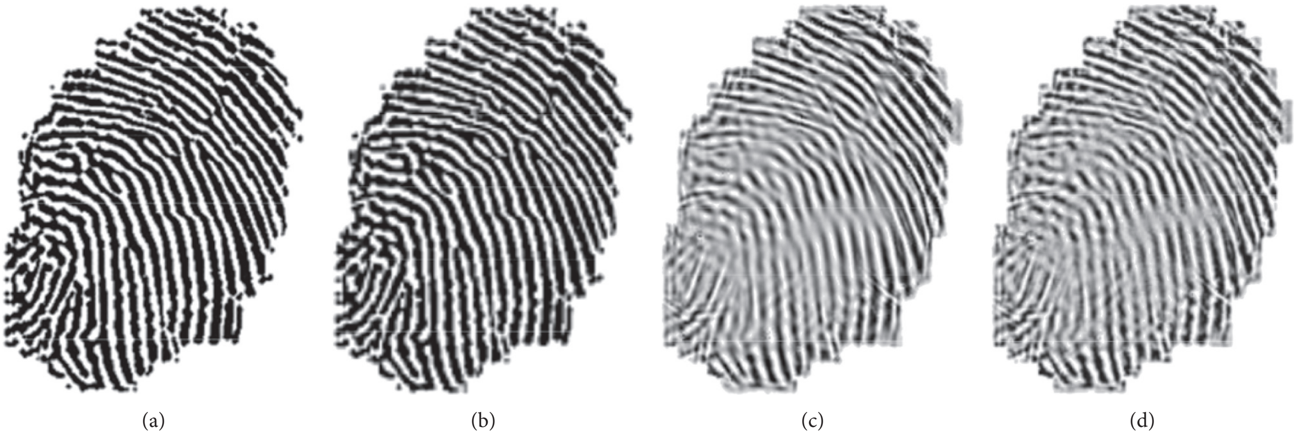


Figure 3: Take samples of adjacent photos.



(a) (b) (c) (d)

Figure 4: Compression result of fingerprint image.

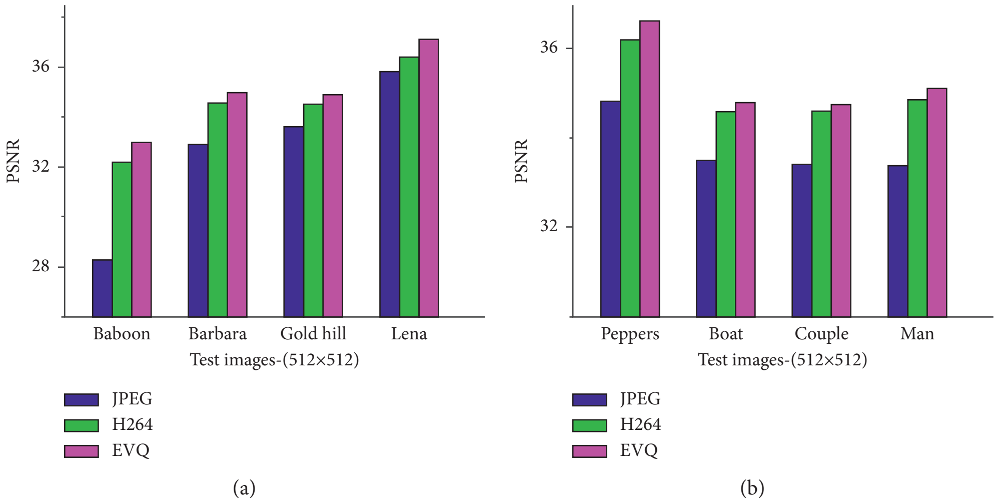


Figure 5: Data compression ratio results of different data sets.

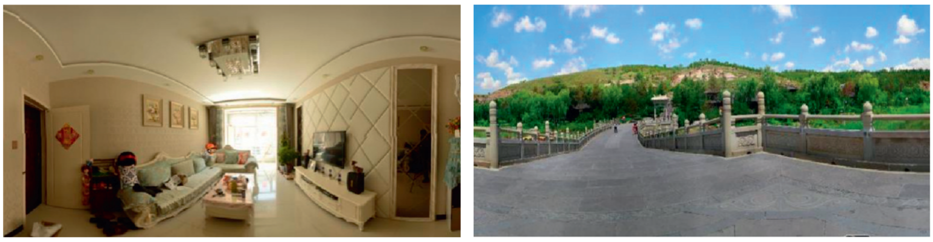


Figure 6: A display of the two scenarios.

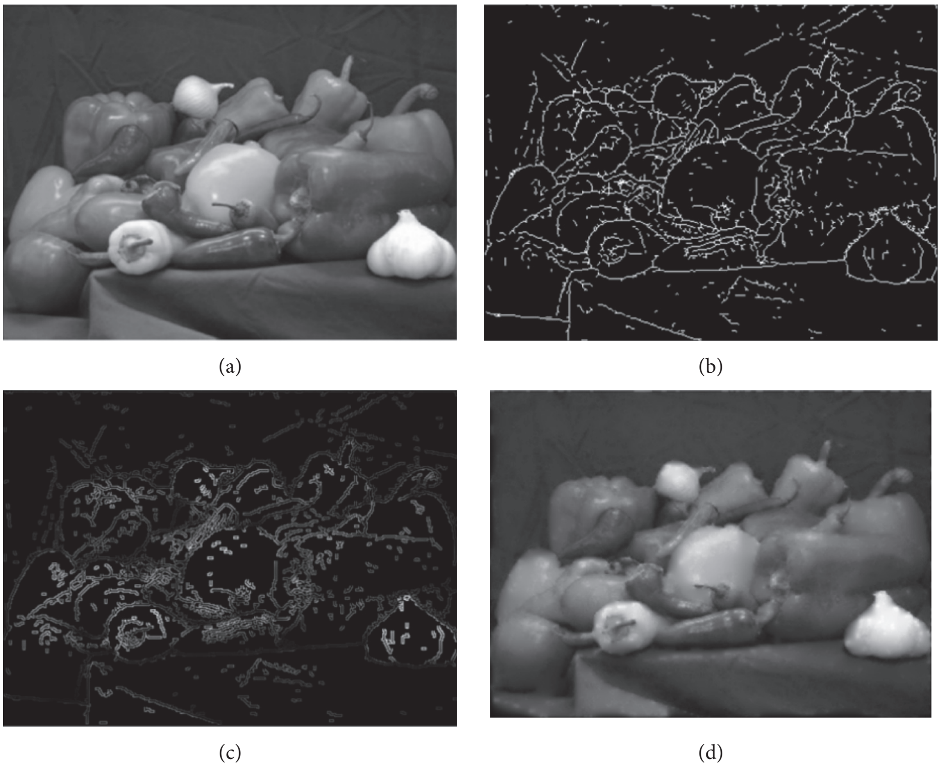


Figure 7: Image compression results display based on SOM.

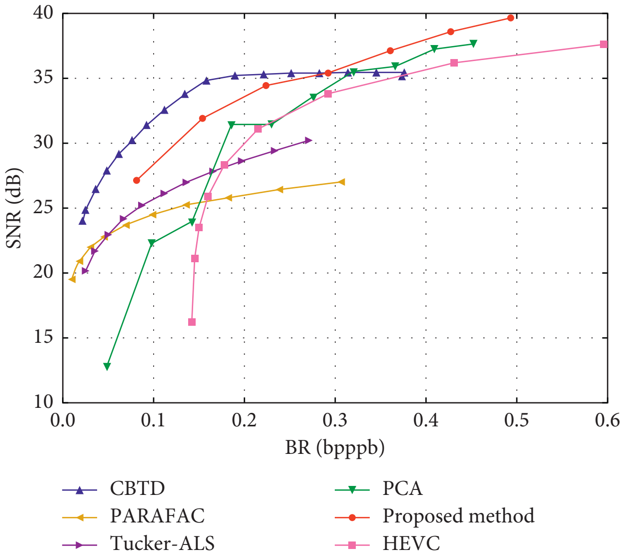


Figure 8: The results of compressed SNR by different methods.

results, but there are still shortcomings, especially the quality-oriented method, the image quality obtained by this method is as good as the existing best DXT1. There are still some gaps in the encoder, so we use the combination method, which beats all the encoders in quality, but the encoding time is much slower, and our combination method has no significant improvement in quality compared with other methods.

As shown in Figure 9, when the K-means algorithm is applied to the set of image feature points, the selection of the initial number of seed points needs to be discussed. First, the number of seed points determines the number of classes after the aggregation of feature points. A larger number of seed points can be used to generate a larger number of classes, which will ensure the accuracy of the ROI region extracted because each class occupies too small area, and too

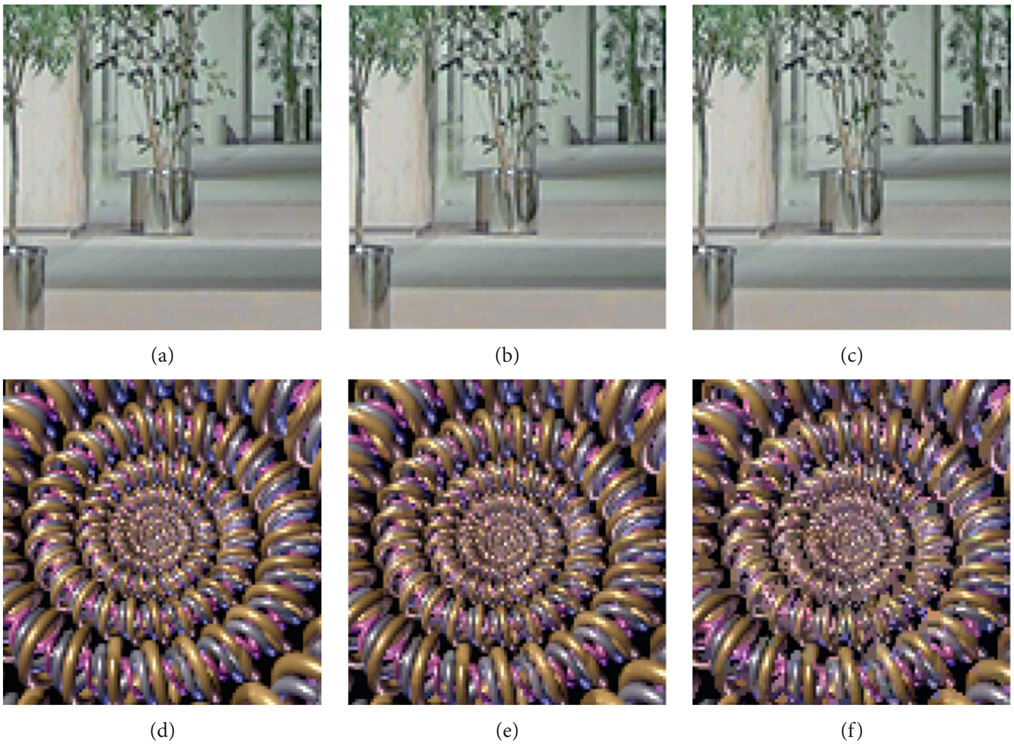


Figure 9: Compression results of physical images at different angles.

few seed points will cause the number of classes. Each class covers the area of the zone too interested in the results of the accuracy which is not high. Second, because of the information on the 3*D* model surface, the surface feature points do not reflect the importance of surface area. Only the set of feature points were distributed in the middle of the area, this area can be seen as area of interest, and after three-dimensional space to two-dimensional space mapping, some feature points in the UV coordinates are coincidence points; the overlap will affect the result of the experiment, need to be removed. Experiments have shown that an initial number of 300 seed points is usually appropriate.

5. Conclusions

Most of the information people get from the surrounding environment comes from vision, and the study of visual images has always been an important direction in the field of scientific research. After entering the information age, people's demand for the quality of digital pictures is getting higher and higher, so digital pictures are developing towards clearer and higher resolution methods to increase the amount of picture data. In recent years, the growth rate of image data exceeds the technological development of storage and transmission speed. It is unrealistic to rely on increasing storage capacity and network bandwidth. We can only consider the image content itself and design image compression algorithm reasonably. When the visual distortion of the image is small, the redundant information of the image should be removed as much as possible to reduce the consumption of storage and bandwidth.

Neural network model can fully excavate the essential information of the image and separate the important features and unimportant redundant information in the image and has a strong potential in the field of image compression. This topic uses deep convolutional network to deal with the problem of image codec, so that the image has a better compression effect. Although the method in this paper aims at the image-based cubic panoramic texture image data, however, the spherical and cylindrical panoramic texture image data can also have a good application in the field of graphic image-based modeling, which provides a new idea for efficient rendering of texture image in 3D scene visualization. The application of the new method in spherical and cylindrical panoramic modeling will be further studied. For deep learning processing of image compression, there are still many aspects to be further studied, such as entropy rate control module in full roll machine network coding and image quality evaluation criteria when using adversarial network.

Data Availability

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

Conflicts of Interest

The authors declare that they have conflicts of interest or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] N. Zikiou, M. Lahdir, and D. Helbert, “Support vector regression-based 3D-wavelet texture learning for hyperspectral image compression,” *The Visual Computer*, vol. 36, no. 7, pp. 1473–1490, 2020.

[2] R. Suresh Kumar and P. Manimegalai, “Near lossless image compression using parallel fractal texture identification,” *Biomedical Signal Processing and Control*, vol. 58, p. 101862, 2020.

[3] S. UmaMaheswari and V. SrinivasaRaghavan, “Lossless medical image compression algorithm using tetrolet transformation,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 4127–4135, 2021.

[4] T. Barbu, “Segmentation-based non-texture image compression framework using anisotropic diffusion models,” *Proceedings of the Romanian Academy - Series A: Mathematics, Physics, Technical Sciences, Information Science*, vol. 20, no. 2, pp. 122–130, 2019.

[5] J. Li, Y. Fu, G. Li, and Z. Liu, “Remote sensing image compression in visible/near-infrared range using heterogeneous compressive sensing,” *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 12, pp. 4932–4938, 2018.

[6] S. Wang, S. Wang, and W. Yang, *Towards Analysis-Friendly Face Representation with Scalable Feature and Texture compression*, IEEE Transactions on Multimedia, vol. 5, no. 2, pp. 11–24, 2021.

[7] J. Uthayakumar, M. Elhoseny, and K. Shankar, “Highly reliable and low-complexity image compression scheme using neighborhood correlation sequence algorithm in WSN,” *IEEE Transactions on Reliability*, vol. 69, no. 4, pp. 1398–1423, 2020.

[8] A. Bouida, M. Beladgham, A. Bassou, and I. Benyahia, “Quality and texture analysis of biometric images compressed with second-generation wavelet transforms and SPIHT-Z encoder,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 3, pp. 1325–1339, 2020.

[9] T. Dumas, A. Roumy, and C. Guillemot, “Context-adaptive neural network-based prediction for image compression,” *IEEE Transactions on Image Processing*, vol. 29, pp. 679–693, 2019.

[10] Z. Chen, Y. Hu, and Y. Zhang, “Effects of compression on remote sensing image classification based on fractal analysis,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4577–4590, 2019.

[11] F. Chen, Y. Yuan, and H. He, “Multi-MSB compression based reversible data hiding scheme in encrypted images,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 905–916, 2020.

[12] J. Li, B. Fu, and Z. Liu, “Panchromatic image compression based on improved post-transform for space optical remote sensors,” *Signal Processing*, vol. 159, pp. 72–88, 2019.

[13] X. Huang, P. An, L. Shen, and R. Ma, “Efficient light field images compression method based on depth estimation and optimization,” *IEEE Access*, vol. 6, pp. 48984–48993, 2018.

[14] Y. Song, Z. Zhu, W. Zhang, L. Guo, X. Yang, and H. Yu, “Joint image compression-encryption scheme using entropy coding and compressive sensing,” *Nonlinear Dynamics*, vol. 95, no. 3, pp. 2235–2261, 2019.

[15] Y. Akbulut, Y. Guo, A. Şengür, and M. Aslan, “An effective color texture image segmentation algorithm based on hermite transform,” *Applied Soft Computing*, vol. 67, pp. 494–504, 2018.

[16] C. Shi and L. Wang, “Remote sensing image compression based on adaptive directional wavelet transform with content-dependent binary tree codec,” *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 3, pp. 934–949, 2019.

[17] V. Lukin, I. Brysina, and V. Makarichev, “Discrete atomic compression of digital images: a way to reduce memory expenses,” *Integrated Computer Technologies in Mechanical Engineering*, vol. 4, pp. 492–502, 2020.

[18] B. Sushma and P. Aparna, “Distributed video coding based on classification of frequency bands with block texture conditioned key frame encoder for wireless capsule endoscopy,” *Biomedical Signal Processing and Control*, vol. 60, Article ID 101940, 2020.

[19] P. Jain and U. Ghanekar, “Robust watermarking technique for textured images,” *Procedia Computer Science*, vol. 125, pp. 179–186, 2018.

[20] Z. Jin, M. Z. Iqbal, and W. Zou, “Dual-stream multi-path recursive residual network for JPEG image compression artifacts reduction,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 2, pp. 467–479, 2020.

[21] S. A. Amiri and H. Hassanpour, “Image compression using JPEG with reduced blocking effects via adaptive down-sampling and self-learning image sparse representation,” *Multimedia Tools and Applications*, vol. 77, no. 7, pp. 8677–8693, 2018.

[22] B. Balakrishnan, S. H. Darsana, J. Mathews, and M. S. Nair, “Satellite/aerial image compression using adaptive block truncation coding technique,” *Journal of the Indian Society of Remote Sensing*, vol. 46, no. 11, pp. 1761–1771, 2018.

[23] S. Ghosh, R. G. Gavaskar, D. Panda, and N. C. Kunal, “Fast scale-adaptive bilateral texture smoothing,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2015–2026, 2019.

[24] R. Meng, S. G. Rice, J. Wang, and X. Sun, “A fusion steganographic algorithm based on faster R-CNN,” *Computers, Materials & Continua*, vol. 55, no. 1, pp. 1–16, 2018.

[25] P. K. Sran, S. Gupta, and S. Singh, “Segmentation based image compression of brain magnetic resonance images using visual saliency,” *Biomedical Signal Processing and Control*, vol. 62, Article ID 102089, 2020.

[26] M. S. Farid, M. Lucenteforte, and M. Grangetto, “Evaluating virtual image quality using the side-views information fusion and depth maps,” *Information Fusion*, vol. 43, pp. 47–56, 2018.

[27] F. Ernawan, M. N. Kabir, and J. M. Zain, “Bit allocation strategy based on Psychovisual threshold in image compression,” *Multimedia Tools and Applications*, vol. 77, no. 11, pp. 13923–13946, 2018.

[28] Z. Pan, H. Fan, and L. Zhang, “Texture classification using local pattern based on vector quantization,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5379–5388, 2015.

[29] L. Armi and S. Fekri-Ershad, “Texture image analysis and texture classification methods-A review,” 2019, https://www.researchgate.net/publication/332370066_Texture_image_analysis_and_texture_classification_methods_-_A_Review.

[30] T. Qiao, R. Shi, and X. Luo, “Statistical model-based detector via texture weight map: application in re-sampling authentication,” *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1077–1092, 2018.

[31] C. Raghavendra, S. Sivasubramanian, and A. Kumaravel, “Improved image compression using effective lossless compression technique,” *Cluster Computing*, vol. 22, no. 2, pp. 3911–3916, 2019.

[32] D. Liu, P. An, and R. Ma, "Content-based light field image compression method with Gaussian process regression," *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 846–859, 2019.

[33] M. Armando, J. S. Franco, and E. Boyer, "Adaptive mesh texture for multi-view appearance modeling," in *Proceedings of the 2019 International Conference on 3D Vision (3DV)*, pp. 700–708, IEEE, Canada, 2019.

[34] Y. Dong, J. Wan, and L. Si, "Deriving polarimetry feature parameters to characterize microstructural features in histological sections of breast tissues," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 3, pp. 881–892, 2020.