Hindawi
Advances in Multimedia
Volume 2022, Article ID 5509637, 14 pages
https://doi.org/10.1155/2022/5509637

Research Article

Analysis of Communication Compression and Transmission in a Multimedia and Internet of Things Environment Integrating Scene Elements

Jia Jia[1] and Guohua Wu[2]

[1]*Henan College of Surveying and Mapping, Zhengzhou 451464, China*
[2]*Zhengzhou University, Zhengzhou 450001, China*

Correspondence should be addressed to Guohua Wu; wgh58@zzu.edu.cn

Received 1 November 2021; Accepted 29 December 2021; Published 3 February 2022

Academic Editor: Qiangyi Li

In order to improve the effect of multimedia data transmission, this paper integrates scene elements to analyze the needs of multiple types of data transmission and uses the technology of the Internet of Things to analyze the compression and transmission of multimedia data communication. Moreover, this paper proposes a reversible information hiding algorithm for encrypted images based on pixel sorting and grouping prediction and discusses how to improve the prediction accuracy through the histogram of prediction errors. In addition, this paper designs a communication compression transmission system in a multimedia and Internet of Things environment integrating scene elements and builds system function modules and system processes. Finally, this paper verifies the performance of the proposed system by means of simulation experiments. From the experimental results, we can see that the system proposed in this paper has a good multimedia information transmission effect.

1. Introduction

With the rapid development of multimedia technology and Internet technology, the relationship between the two is getting closer and closer. Multimedia network technology has gradually penetrated into schools, families, and society, making the connection between people beyond the limits of time and space. Network multimedia, including text, graphics, images, sounds, applications, and animations, can run on computer networks at the same time without affecting each other [1]. Any terminal on the network can share multimedia information and can also store, process, and retransmit the acquired multimedia data. That is to say, network multimedia is developed on the basis of computer, multimedia, and network. It integrates computer, multimedia, and network and can realize various functions such as transmission and exchange of graphics, images, sound, and text in the network [2]. Network multimedia includes two aspects. (1) Different from the general network, network multimedia requires a special equipment environment. (2) Compared with traditional multimedia, the computer equipment used in network multimedia is quite special. The way that network multimedia processes information such as images and sounds is different from that of general multimedia, so the requirements for hardware equipment are higher [3]. The quality of the image is determined by the number of frames transmitted per second and the size of the image. In terms of images, if images are played at a frame rate of thirty frames per second, the human eye can see continuous images. However, in the network multimedia system, the playback effect of 30 frames per second is not ideal. We need to use DIV, JPEG, MPEG, and other encoding and decoding methods [4].

The development of multimedia technology is inseparable from the development of other technologies. Multimedia network transmission technology is essentially the product of a combination of multiple technologies, including multimedia computer technology, network

communication technology, and multimedia data encoding and decoding technology. It is precisely because of the development of network communication technology that computers all over the world are connected to each other to form a global network, which provides a prerequisite for the sharing of resources. Each computer is no longer a separate device, but a node in the entire network.

In the actual multimedia transmission process, the transmission of many multimedia signals, including audio, video, and image data, still uses relatively old technical means based on analog signal transmission. The use of analog signals to transmit multimedia signals has a long history. The main problems it brings are the high cost of the system, the need to lay a long line, the long construction period, and the poor stability. Once the scale of the system becomes larger, it is often necessary to lay more cables. These thick and long cables increase the complexity of the circuit. If the system structure changes, the existing lines need to be changed. This makes it very inconvenient to implement maintenance on the entire system.

With the support of the Internet of Things technology, this paper studies multimedia communication transmission, improves the corresponding algorithm, builds an intelligent model, and improves the multimedia communication compression transmission effect that integrates scene elements.

2. Related Work

The literature [5] applied compressed sensing technology to the field of camera imaging and successfully produced a single-pixel digital camera. The literature [6] used Bayer color filters to enable single-pixel digital cameras to produce color images. The literature [7] used background attenuation technology combined with single-pixel camera data to automatically monitor and track targets. The literature [8] proposed a single-pixel terahertz imaging system based on compressed sensing to successfully avoid mechanical planar image scanning. The literature [9] proposed a sparse imaging technology with the support of compressed sensing theory and a dynamic imaging technology at the same time. In communication technology and network, the literature [10] proposed fast Fourier sampling algorithm based on compressed sensing theory for wireless signal sensing and compression. The literature [11] proposed a cyclic feature detection framework based on compressed sensing for broadband spectrum sensing, which uses second-order statistical information to meet the high sampling rate requirements of traditional cyclic spectrum remote sensing. The literature [12] uses compressed sensing to restore sparse signals through decentralized channels in a large-scale energy-efficient network. The literature [13] proposed a framework to improve the compressed sensing algorithm, making it more suitable for the restoration of overlay network traffic matrix and delay matrix. By calculating the traffic matrix and the delay matrix, the congestion of the overlay network can be estimated, which can reflect the current network security status.

The compressed sensing method proposed in [14] ignores the internal correlation of a single set of signals and only pays attention to the cross-correlation of multiple sets of signals. Ray [15] formally proposed distributed compressed sensing based on the compressed sensing series theory. The main idea of distributed compressed sensing is to extend the original compressed sampling of a single signal to the centralized compressed sampling of the signal group, focusing on the analysis of the internal correlation and cross-correlation of the signal group, and then use these properties to perform joint recovery of the signal group. Compared with traditional compressed sensing, distributed compressed sensing has greatly reduced the requirements for the amount of observation data in the processing of signal groups, which has further expanded the application of compressed sensing theory. In the literature [16], the author proves through experiments that distributed compressed sensing processing related signals can improve the performance by 30%. The joint sparse model of signal group is the theoretical basis of distributed compressed sensing. Observation, the obtained observation data can be accurately restored and reconstructed at the receiving end. Literature [17], based on a comprehensive analysis of the theoretical knowledge and application background of the distributed compressed sensing series, proposed three different joint sparse models for different application scenarios and gave the corresponding signal compression schemes. The paper discusses the selection principle of distributed compressed sensing observation matrix and uses a simple random sparse projection matrix as the observation matrix. The literature [18] focused on the problem of restoring and reconstructing the signal group with the distributed compressed sensing method and adopted a reconstruction error estimation method to improve the existing distributed compressed sensing method.

Literature [19] proposed a distributed compressed sensing model based on spatial correlation in wireless sensor networks. After analyzing the joint sparse model and the spatial correlation between nodes in the wireless sensor network, the DCS encoding and decoding algorithm was adopted, and the energy of the node was used as an evaluation index to perform a series of operations of distributed compressed sensing. Literature [20] applies compressed sensing technology to point-to-point networks and distributed networks in the Internet of Things, studies the existing three types of joint sparse models of distributed compressed sensing, and implements distributed compressed video sensing GPSR (Gradient Projection for Sparse Reconstruction) algorithm focusing on the analysis of Bayesian compressed sensing, using the prior probability distribution characteristics of the original signal, and using the Bayesian principle in the reconstruction algorithm, which makes Bayesian compressed sensing under the same

conditions. Compared with the traditional compressed sensing model, the model can recover and reconstruct the original signal more accurately.

3. IoT Communication Compression Processing Algorithm

The image encryption algorithm can be realized by changing the pixel value or changing the pixel position. Commonly used symmetric encryption algorithms mainly include XOR encryption, scrambling encryption, and mod 256 encryption.

XOR encryption uses an encryption key to generate a stream cipher and then performs XOR operation on each bit of the stream cipher image pixel bit by bit to obtain an encrypted image. For a 256-level grayscale image I with a size of M ∗ N, each pixel can be divided into 8 bits. For the pixel $I(i, j)$, the s-th bit can be expressed as $I(i, j, s)$, as calculated by the following formula [21]:

$$I(i, j, s) = \left\lfloor \frac{I(i, j)}{2^s} \right\rfloor \bmod 2, s = 0, 1, \ldots, 7. \quad (1)$$

Among them, i represents the row where the pixel is located, and j represents the column where the pixel is located.

After obtaining the 8 bits of the original image pixels, an 8 bit pseudorandom sequence P with a size of M ∗ N can be generated according to the encryption key. $P(i, j)$ and pixel $I(i, j)$ are XORed to obtain encrypted pixel $E(i, j)$, and finally encrypted image E is obtained. The following formulas are specific calculation processes:

$$E(i, j, s) = I(i, j, s) \oplus P(i, j, s), \quad (2)$$

$$E(i, j) = \sum_{s=0}^{7} (i, j, s) \times^s. \quad (3)$$

Figure 1 shows the changes of Lena images before and after encryption. It can be seen that the encrypted image is the same as the noise, and the original image content is completely invisible to the human eye, thus achieving the purpose of protecting the image content. The reason why the XOR encryption algorithm can achieve the encryption effect is mainly that the random sequence P is randomly generated according to the key, without any rules, and is evenly distributed. Moreover, the encrypted pixels obtained after the XOR become evenly distributed, without retaining any features of the original image, and greatly weaken the correlation between the pixels of the original image. Figure 1(a) is the original image histogram. We can see the approximate distribution of pixel values in the original image. Figure 1(b) is an encrypted image histogram. Compared with the original image histogram, the pixel values of the encrypted image histogram are evenly distributed without retaining any distribution characteristics of the original image. Therefore, the original image cannot be found through the encrypted image, indicating that the exclusive OR encryption algorithm is safe and can achieve the purpose of protecting the security of the image.

For a grayscale image I with a size of MN, the range of pixels is [0, 255]. According to the encryption key, a random matrix R with the same size as the image is generated, the image I and the random matrix R are added, and then modulo 256 is added to obtain the encrypted image E. The specific encryption process is shown in the following formula [22]:

$$E(i, j) = (I(i, j) + R(i, j)) \bmod 256. \quad (4)$$

When the image is decrypted, the encrypted image E is subtracted from the random matrix R and then modulo 256 to obtain the original image I. The specific calculation process is as shown in the following formula:

$$I(i, j) = (E(i, j) - R(i, j)) \bmod 256. \quad (5)$$

Mod 256 encryption can be completely decrypted, and there is no unrecoverable boundary point. For example, when the original pixel is $l(ij) = 255$, the random number is $R(i, j) = 1$ and the encrypted pixel is $E(ij) = 0$, and when decrypted, $I(i, j)=(0-1)$ and mod $256 = 255$. For the original pixel $I(ij) = 0$, the random number is $R(i, j) = 255$, then the encrypted pixel is $E(ij) = 255$, the decryption is $E(ij)$-$R(ij) = 0$, the pixel is $I(i, j) = 0$, and mod $256 = 0$. In the same way, other pixel values can be also completely decrypted.

Arnold transformation is a typical scrambling encryption algorithm. The main idea is to perform multiple elementary matrix transformations on the image matrix, scramble the position of the image pixels, and then realize image encryption. Arnold transformation is mainly for square images, and for rectangular images, it needs to be filled into a square matrix. Arnold transformation has a transformation cycle, which is related to the pixel position transformation cycle during encryption and decryption. For images of different sizes, the transformation period is different. For images of size NN, the period T is shown in Table 1 [23].

If it is assumed that the period of the Arnold transform is $t1$ and the period of the inverse Arnold transform is $t2$, then $T = t1 + t2$. Arnold transformation is transformed by elementary matrix, and the specific calculation process is shown in the following formula:

$$\begin{pmatrix} i' \\ j' \end{pmatrix} = \begin{pmatrix} 1 & b \\ a & ab+1 \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} \bmod N. \quad (6)$$

Among them, a and b are variable values, but the matrix must satisfy the determinant value of 1, such as $a-b = 1$. (i,j) represents the position coordinates of the pixel, (i',j') represents the pixel coordinates after transformation, and the N ∗ N grayscale image is transformed $t1$ times according to (5) to obtain the encrypted image. In the Arnold inverse transformation, we only need to change the matrix to the inverse matrix of the original elementary matrix. The specific calculation is as follows:

$$\begin{pmatrix} i \\ j \end{pmatrix} = \begin{pmatrix} ab+1 & -b \\ -a & 1 \end{pmatrix} \begin{pmatrix} i' \\ j' \end{pmatrix} \bmod N. \quad (7)$$

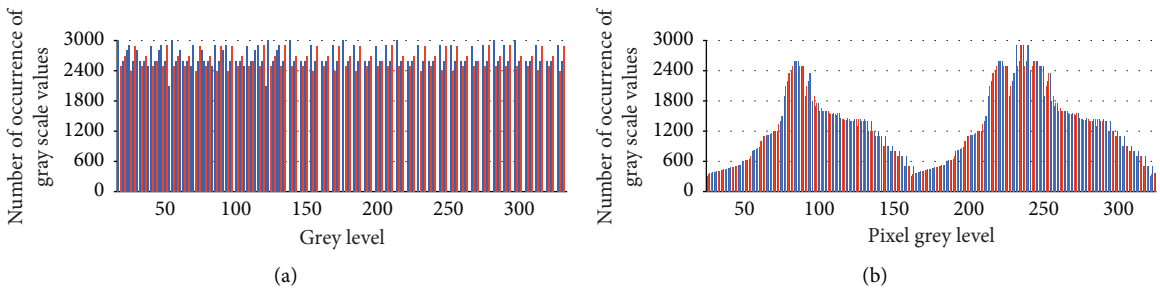


Figure 1: Comparison of histograms before and after Lena XOR encryption: (a) original histogram; (b) encrypted image histogram.

Table 1: Arnold transformation period of images of different sizes.

N	2	4	6	12	25	64	128	256	512
T	3	3	12	12	50	48	96	192	384

Figure 2 shows the original histogram of the Lena image and the histogram of the scrambled encrypted image. It can be seen that the changing trends of the two histograms are the same, indicating that the encrypted image retains the characteristics of the original image. Therefore, the encryption method has security problems, which may cause the original image information to be leaked.

Figure 3(a) is an original image block, using the encryption key to generate a random number, which is set to 56. Figure 3(b) is the image block after XOR encryption, and Figure 3(c) is the image block after mod 256 encryption. Figures 3(d) and 3(e) show the corresponding difference (pixels in the block minus the pixel at the first position) image block. It can be seen from Figure 3 that Figure 3(f) and Figure 3(d) are closer. It shows that the mod 256 encryption algorithm can better preserve the correlation of the image, so the mod 256-scrambling encryption algorithm is more suitable for introducing the traditional reversible information hiding algorithm into the encrypted image.

Figure 4 shows the encryption effect of the Lena image under different size blocks, and Figures 4(a)–4(c) show the effect diagram encrypted with the mod 256 algorithm. It can be seen that as the block becomes larger, the image encryption effect becomes worse, and the basic outline of the original image can even be seen. Therefore, only using mod 256 encryption is insecure. Figures 4(d) and 4(e) show the encrypted image after block scrambling. It can be seen that compared with Figures 4(a)–4(c), no original image content can be seen. Therefore, the encryption effect of using mod 256-scrambling encryption algorithm is better than that of using mod 256 encryption algorithm alone.

Figure 5 shows the encrypted image histogram of the block mod 256-block dislocation encryption algorithm for different block sizes. From Figures 5(b)–5(d), it can be seen that the encrypted image has a uniform pixel distribution and does not retain any features of the original image. Thus, the original image cannot be obtained by attacking the histogram of the encrypted image, indicating that the mod 256-block dislocation encryption algorithm is secure.

As the block becomes larger, it can be seen that the distribution of the encrypted image histogram gradually changes, and it begins to become less uniform. Therefore, when using this encryption method, in order to ensure the security of the original image, it is necessary to use as small image blocks as possible.

The detailed operation steps of the histogram modification algorithm are as follows.

Step 1: the algorithm scans the image, counts each pixel value, and obtains the histogram of the image as shown in Figure 6. The peak point pixel value $P = 121$ is obtained from Figure 6, the zero point is the pixel value in the range of [209, 255], and the pixel value $Z = 209$ is the minimum zero point on the right.

Step 2: the algorithm finds the peak point P and the zero point Z in the histogram. The following takes the case of $P < Z$ as an example to introduce the histogram modification algorithm. $P < Z$ indicates that the zero point is on the right of the peak point. Before embedding additional data, the pixel value point falling in the interval (PZ) is shifted to the right by one unit. At this time, the number of pixel value points P+1 is zero, which can be used to hide the additional data. The specific operation is shown in (8), where b is the additional data, x is the original pixel value, and x' is the modified pixel value.

$$x' = \begin{cases} x + b, & \text{if } x = p, \\ x + 1, & \text{if } p < x < Z, \\ x, & \text{otherwise.} \end{cases} \tag{8}$$

Step 3: when extracting additional data, the algorithm first obtains the histogram of the image with additional data according to Step 1. The recovery process is the opposite of the hiding process. Since the location of the peak point will not change, the algorithm first finds the peak point P. The additional data 0 is extracted at the pixel point equal to P, and the additional data 1 is extracted at the pixel point equal to P+1. The following formula is the additional data extraction process:

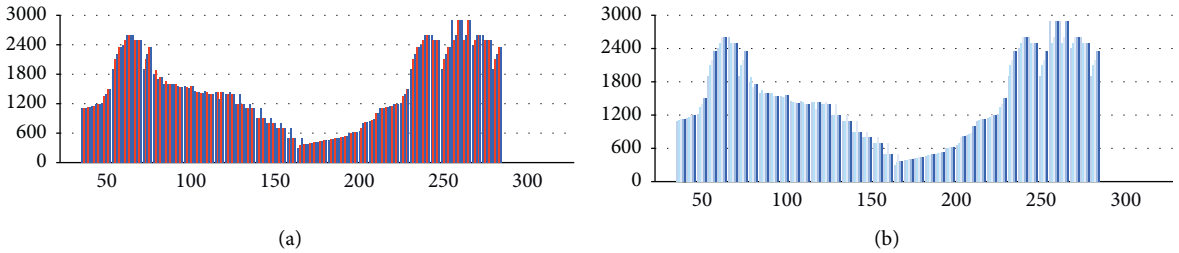


Figure 2: Arnold transform histogram: (a) original histogram; (b) scrambling encryption histogram.

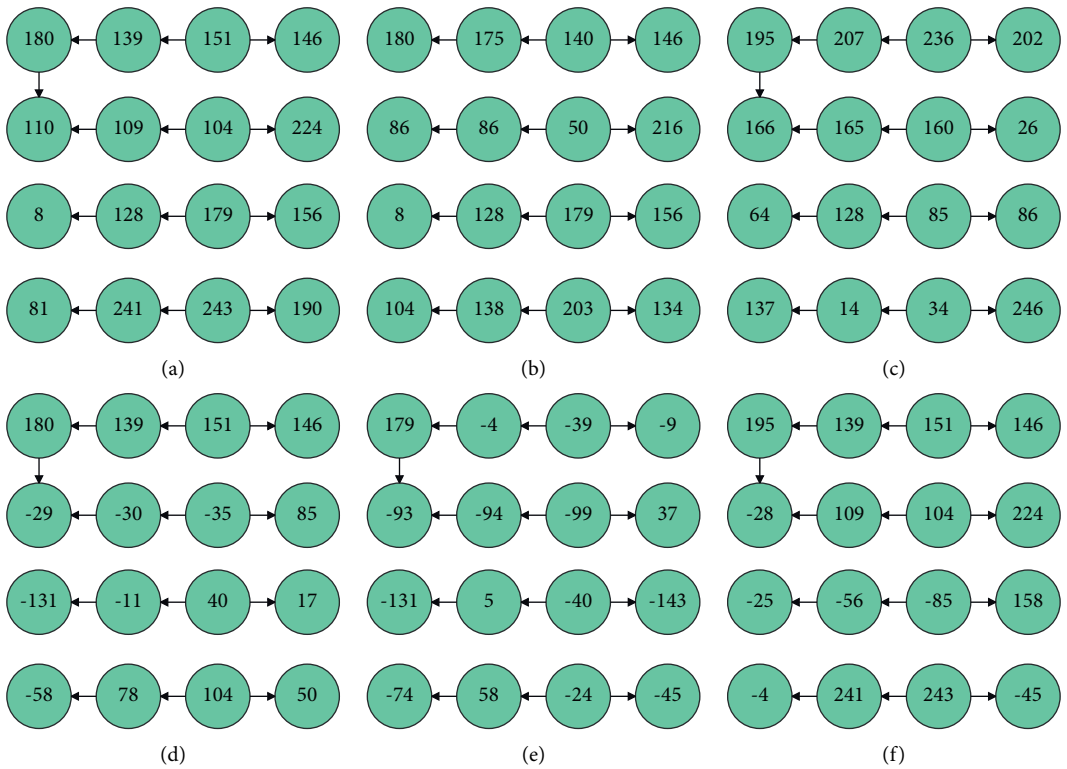


Figure 3: Comparison of the relevance of encryption methods. (a) The original image block. (b) Encrypted image block. (c) Mod 256 encryption image block. (d) The original image block D-value. (e) Encrypted image block D-value. (f) Mod 256 encryption image block D-value.

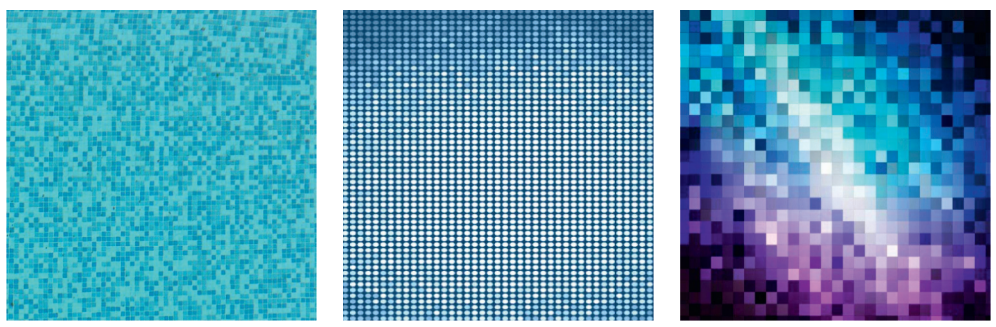
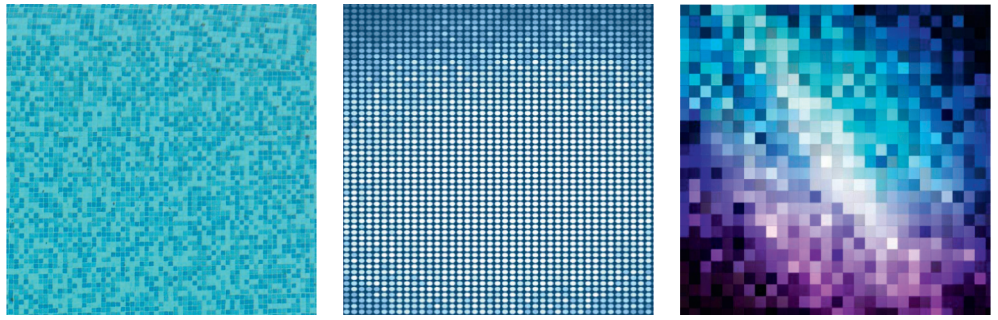


Figure 4: Continued.



(d) (e) (f)

Figure 4: Comparison of encryption effects of different encryption algorithms. (a) 4 ∗ 4 mod2 56s. (b) 8 ∗ 8 mod 256. (c) 16 ∗ 16 mod 256. (d) 4 ∗ 4 mod 256 scrambling. (e) 8 ∗ 8 mod 256 scrambling. (f) 16 ∗ 16 mod 256 scrambling.

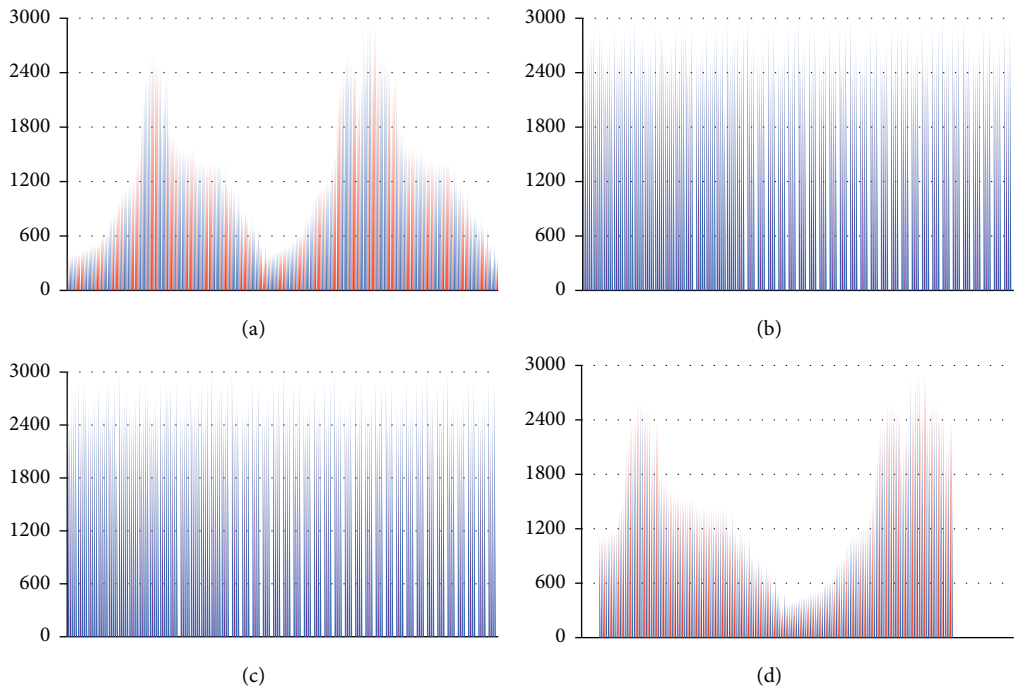


Figure 5: (a) Original and encrypted Lena image histogram. Mod 256-scrambling encrypted image histogram: (b) 4 ∗ 4 block; (c) 8 ∗ 8 block; (d) 16 ∗ 16 block.

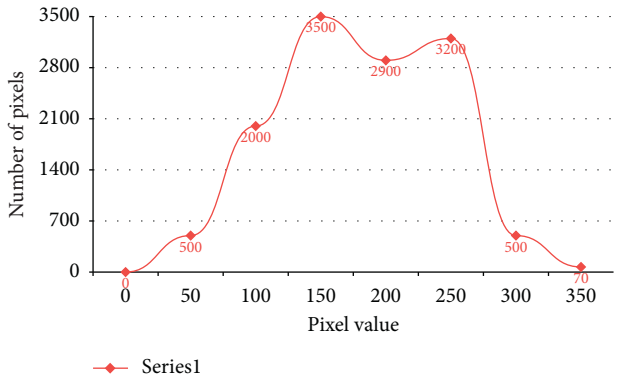


Figure 6: Image histogram.

$$b = \begin{cases} 0, & \text{if } x' = P, \\ 1, & \text{if } x' = P + 1. \end{cases} \tag{9}$$

Step 4: after the additional data is extracted, the algorithm needs to restore the image. If the pixels in the range of [P+1, Z] are shifted by one unit to the left, the original image can be obtained. The specific operation process is as follows:

$$x = \begin{cases} x', & \text{if } x' = P, \\ x' - 1, & \text{if } P + 1 < x < Z + 1, \\ x', & \text{otherwise.} \end{cases} \tag{10}$$

After determining the pixel prediction method, the algorithm uses three-neighbor pixels to predict the pixel. Formula (11) shows the prediction process, where a, b, and c are the pixels at the right, bottom, and right diagonal positions of the predicted pixel x, respectively.

$$x' = \begin{cases} \min(a, b), & \text{if } c \geq \max(a, b), \\ \max(a, b), & \text{if } c \leq \min(a, b), \\ a + b - c, & \text{otherwise.} \end{cases} \tag{11}$$

After the pixel predicted value x' is obtained, the original pixel x is subtracted from its predicted value x' to obtain the prediction error e, as shown in (12), to obtain the space for embedding additional data. The specific operation process is as in (13). After the additional data is embedded in the prediction error, the pixel value X after the embedded additional data can be obtained by formula (14).

$$e = x - x', \tag{12}$$

$$e' = \begin{cases} e - 1, & \text{if } e < -1, \\ e - b, & \text{if } e = -1, \\ e + b, & \text{if } e = 0, \\ e + 1, & \text{if } e > 0, \end{cases} \tag{13}$$

$$X = x' + e'. \tag{14}$$

The prediction error histogram is Gaussian distributed, with the peak generally at prediction error 0. Figure 7 shows the process of panning the prediction error histogram to hide the additional data. Figure 7(a) shows the original prediction error histogram. If we choose to hide the additional data at prediction error −1 and 0, the prediction error region to the left of prediction error −1 is shifted one unit to the left to free up prediction error −2 as the space for hiding the additional data. Similarly, the prediction error 1 is vacated to the right of the prediction error 0 as the space of hidden additional data, the histogram of prediction error after translation is obtained as in Figure 7(b), and 7(c) shows the histogram of the prediction error after embedding the additional data. The errors −2 and 1 are obtained when the additional data hidden in the prediction error value −1 and 0 is 1. If the additional data hidden is 0, the errors −1 and 0 remain unchanged.

The prediction value x' can be obtained accurately using the same prediction method, and the prediction error e' is obtained by subtracting the pixel X containing additional data from the prediction value x'. According to the principle of modifying the prediction error to hide the additional data, the range of the prediction error with additional data can be inferred. It is also obvious from Figure 7(c) that the prediction error with additional data is in the range of [−2, 1]. Then, the additional data is extracted, and the prediction error is recovered according to the prediction error range, as follows:

$$b = \begin{cases} 0, & \text{if } e' = -1 \text{ or } 0, \\ 1, & \text{if } e' = 1 \text{ or } -2, \end{cases} \tag{15}$$

$$e' = \begin{cases} e' - 1, & \text{if } e \geq 1, \\ e' + 1, & \text{if } e' \leq -2, \\ e', & \text{otherwise.} \end{cases} \tag{16}$$

Step 5: The algorithm extracts additional data b through Step 3, and then obtains the initial prediction error e. The original pixel x can be obtained by adding the prediction error e to the pixel x containing the additional data, as shown in the following formula:

$$x = X + e. \tag{17}$$

The additional data embedding rate indicates how many bits of additional data can be hidden in a pixel, and is an evaluation of the image's ability to carry additional data. The higher the additional data embedding rate, the better the algorithm, and the lower the additional data embedding rate, the worse the algorithm. The following formula calculates the additional data embedding rate with an image size of M ∗ N and an embedding capacity of num:

$$\text{rate (bpp)} = \frac{\text{num}}{M \times N}. \tag{18}$$

For an 8 bit 256-level grayscale image with a size of M ∗ N, its PSNR value is calculated as follows:

$$\text{PSNR} = 10 \times \log 10\left(\frac{255 \times 255}{\text{MSE}}\right). \tag{19}$$

Among them, MSE is the mean square error between the directly decrypted image and the original image, as defined in (20), where I is the original image and Γ is the directly decrypted image.

$$\text{NSE} = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(I(i, j) - I'(i, j)\right)^2}{M \times N}. \tag{20}$$

For an image block containing n pixels, the pixels are sorted in ascending order; nL, nH, and nR are used to represent the lengths of sets L, H, and R; and each set represents a category of pixels. Pixel $\{x(1), x(2), \ldots, x(n)\}$ in the block is sorted in ascending order to get $\{h(1), h(2), \ldots, h(n)\}$, and $L = \{h(1)\}, R = \{h(n)\}, i = 1, j = n, nL = 1, nR = 1$, are initialized. The specific classification process is described in (1), where the parameter EL is the

maximum value of the prediction error interval used to embed additional data, and i and j are used for classification and counting.

$$\begin{cases} h(i) \in L, i = i + 1, \text{if h}(i) - h(1) \le \text{EL and i} \le j \le n, \\ h(j) \in R, j = j - 1, \text{if h}(n) - h(j) \le \text{EL and j} \le i \ge 1. \end{cases} \tag{21}$$

After the classification, the remaining pixels in the block form a set H. In pixel prediction, different prediction methods are designed for different sets according to their characteristics. The predicted value of the pixel in the set L is the maximum value max(L) in the set, and the predicted value of the pixel in the set R is the minimum value min(R) in the set. The pixels in the set H are predicted by judging the relationship between the difference between the maximum values max(H) and max(L) of the pixels in the set H and the parameter EL. The specific prediction process for the pixels in the set H is as follows:

$$f(i) = \begin{cases} \max(L), & \text{if } i = nH, \\ \max(L), & \text{if } 1 \le i \le nH, \max(H) - \max(L) \le \text{EL}, \\ \max(L) + EL, & \text{if } 1 \le i \le nH, EL < \max(H) - \max(L) \le 2\text{EL}, \\ \max(L) - EL, & \text{if } 1 \le i \le nH, \max(H) - \max(L) > 2\text{EL}. \end{cases} \tag{22}$$

An image I of size M ∗ N is divided into nonoverlapping blocks $\{B_1, B_2, \ldots, B_t\}$ of $u \times v$ size, where $t = (M * N)/(u * v)$ represents the number of blocks, and the specific image encryption steps are as follows.

Step 1: The algorithm generates a pseudorandom sequence R of length t according to encryption key 1 and adds mod 256 to the random number in the sequence and the pixels in the block. Moreover, the same random number is added to the pixels in each block, and the specific operation is shown in the following formula to obtain the encrypted image block $\{E_1, E_2, \ldots, E_t\}$.

$$\begin{aligned} E_s(i, j) &= \left(B_s(i, j) + R_s\right) \text{mod} 256 \\ &1 \le i \le u, 1 \le j \le v, 1 \le s \le t. \end{aligned} \tag{23}$$

Step 2: The algorithm scrambles the position of the encrypted image block and generates a random natural number sequence w from 1 to t according to encryption key 2. The position of the block is exchanged according to the sequence W, the obtained block is $\{E_{w1}, E_{w2}, \ldots, E_{wN}\}$, and then the block is reorganized to obtain the encrypted image C.

In order to make the scrambling process clearer, Figure 8 is an example of scrambling encryption. The sequence $W =$ {5, 3, 9, 7, 2, 1, 4, 8, 6} is generated according to encryption key 2. This sequence corresponds to the original sequence {1, 2, 3, 4, 5, 6, 7, 8, 9}; for example, 5 corresponds to 1 and the positions are swapped. Figure 8(a) is the position before scrambling, and Figure 8(b) after scrambling. Figure 8 shows that the position after scrambling is completely different from the original position, and the scrambling effect is achieved.

Additional data hiding is divided into three parts: (1) encrypted image overflow processing; (2) pixel selection; (3) embedded additional data. Figure 9 is a general flow chart of the additional data embedding algorithm. The process of recording the location is shown in as follows:

$$\text{LN}(i, j) = \begin{cases} 1, & \text{if } C(i, j) > 255 - (\text{EL} + 1), \\ & \text{and } C(i, j) < \text{EL} + 1, \\ 0, & \text{if } C(i, j) \in [255 - 2\text{EL} - 1255 - \text{EL} - 1], \\ & \text{and } C(i, j) < [\text{EL} + 1, 2\text{EL} + 1]. \end{cases} \tag{24}$$

The specific calculation process of the processed encrypted image E is shown as follows:

$$e(i, j) = \begin{cases} c(i, j) - \text{EL} - 1, & \text{if } c(i, j) > 255 - \text{EL} - 1, \\ c(i, j) + \text{EL} + 1, & \text{if } c(i, j) < \text{EL} + 1, \\ c(i, j), \text{ otherwise.} \end{cases} \tag{25}$$

In order to accurately extract the additional data, the com value in the last row and the last column of the image is recorded as 0. According to the three pixels on the right, bottom, and right diagonal positions, the com value is obtained by calculating the absolute value of the difference between the right pixel and the bottom pixel and the right diagonal pixel. The specific calculation is as follows:

$$\begin{aligned} \text{com}(i, j) = &|e(i + 1, j) - e(i + 1, j + 1)| \\ &+ |e(i, j + 1) - e(i + 1, j + 1)|. \end{aligned} \tag{26}$$

The predicted value of e1 is obtained according to (22); the predicted value is specifically calculated as (27); and the predicted value of pixel e1l is subtracted from the predicted value to obtain the prediction error Pel of $e1$, which is calculated as formula (28).

$$\widetilde{e} = \begin{cases} \max(L), e \le \max(L), \\ \text{MIN}(R), e \ge \min(R), \\ f, \text{ other,} \end{cases} \tag{27}$$

$$\text{Pe} = e - \widetilde{e}. \tag{28}$$

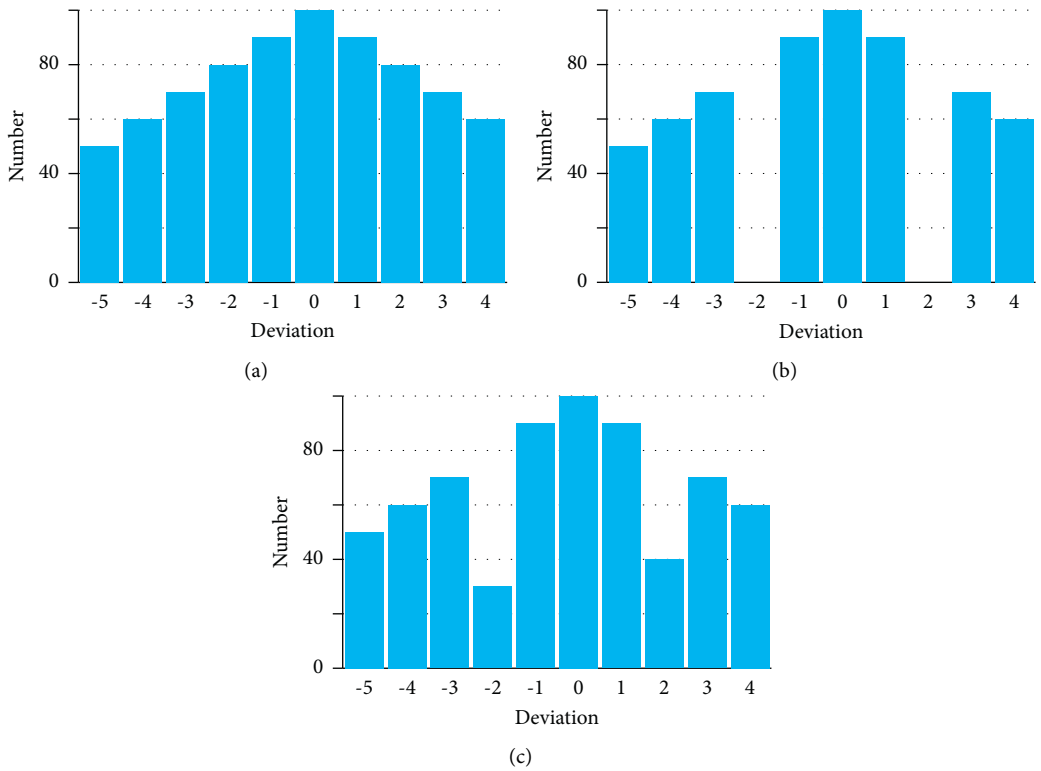


Figure 7: The translation process of the prediction error histogram: (a) histogram of the prediction error; (b) histogram of the prediction error after translation; (c) prediction error histogram with additional data.

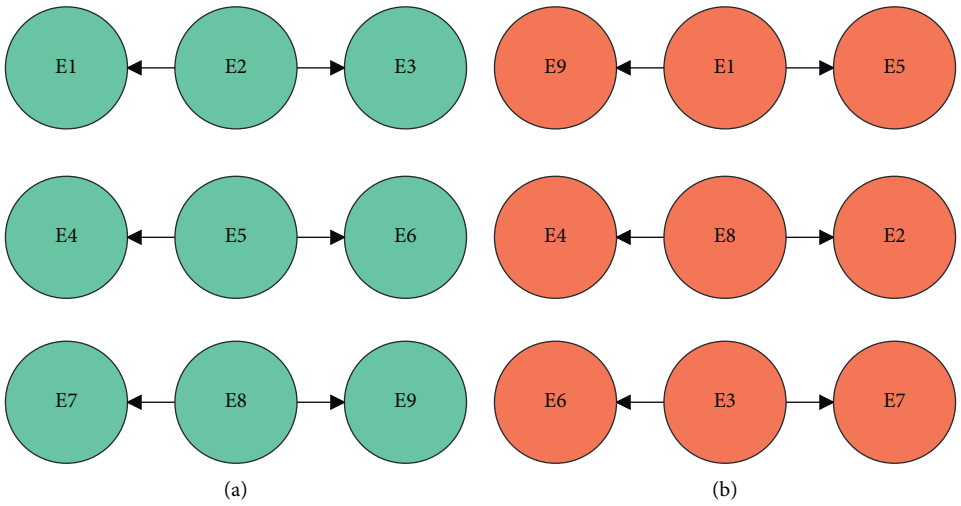


Figure 8: Changes before and after image block scrambling: (a) images block before scrambling; (b) images block after scrambling.

The specific calculation of the additional data embedding is shown in formula (29), where e is the pixel value after embedding the additional data and b is the additional data.

$$e' = \begin{cases} e - (\mathrm{EL} + 1), & \text{if } \mathrm{Pe} < -\mathrm{EL}, \\ e + \mathrm{Pe} - b, & \text{if } -\mathrm{EL} \le \mathrm{Pe} < 0, \\ e + \mathrm{Pe} + b, & \text{if } 0 \le \mathrm{Pe} < \mathrm{EL}, \\ e + (\mathrm{EL} + 1), & \text{if } \mathrm{Pe} > \mathrm{EL}. \end{cases} \tag{29}$$

The additional data is extracted in three steps: (1) pixel selection; (2) pixel prediction; (3) extraction of additional data. Because this part scans the image and the image block in reverse order, it can accurately calculate the complexity of the pixel and select the pixel to obtain the sets L, H, R. According to the size relationship between the pixel $e4'$ and the set critical values max(L) and min(R), it is determined which set the pixel e4 belongs to, and the predicted value of the pixel $e4'$ is predicted. The specific operation is shown in

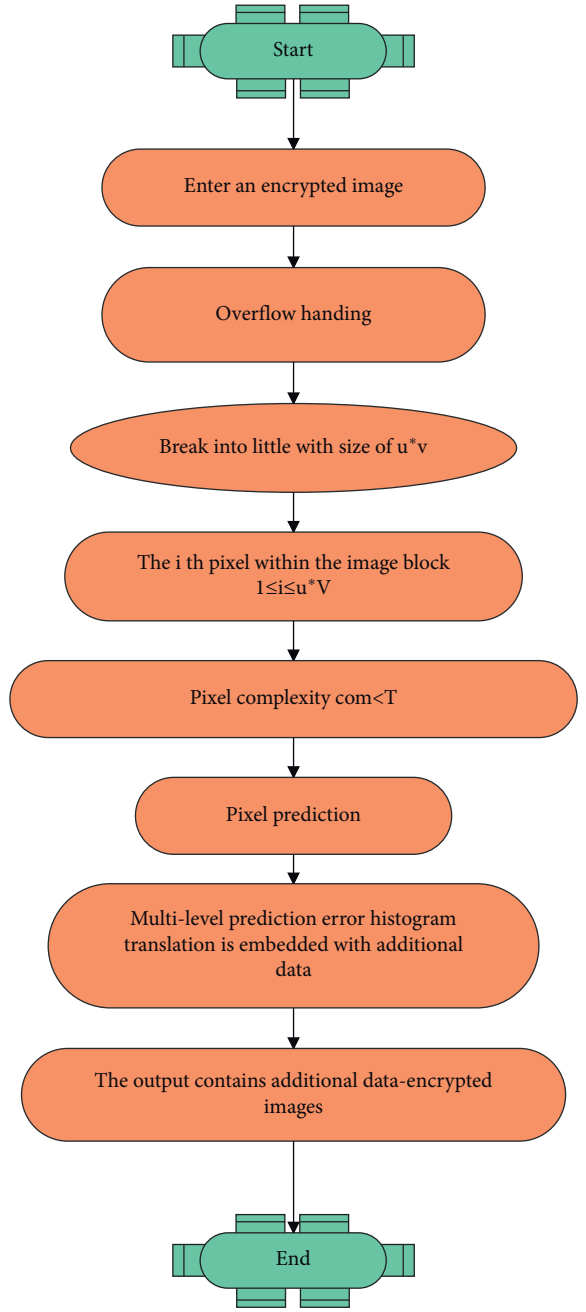


Figure 9: Flow chart of the additional data embedding algorithm.

(17). The predicted value $\tilde{e}4$ of the pixel $e4'$ is subtracted to obtain the prediction error $Pe4'$, as shown in the following:

$$Pe' = e' - \tilde{e}. \tag{30}$$

The peak area of the embedded additional data is [−EL, EL], and the maximum change value of the pixel is (EL+1). Therefore, if the prediction error $Pe4'$ is in the range of [−2EL−1, 2EL+1] when the additional data is extracted, this means that the pixel $e4'$ contains additional data, and the additional data b is extracted according to the prediction error Pel'. The specific calculation is as follows:

$$b = \begin{cases} 0, & \text{if } Pe' \bmod 2 = 0, \\ 1, & \text{if } Pe' \bmod 2 = 1. \end{cases} \tag{31}$$

After the additional data is extracted, the pixel $e4'$ is restored according to the extracted additional data b to obtain e4. If the additional data is not included, the pixel $e4''$ is directly modified according to the prediction error $Pe4'$ to

obtain e4. The specific calculation process is shown in the following formula:

$$e = \begin{cases} e' + (\mathrm{EL}+1), & \text{if } \mathrm{Pe}' < -2\mathrm{EL}-1, \\ \widetilde{e} + \mathrm{fix}\left(\dfrac{\mathrm{Pe}' - b}{2}\right), & \text{if } 0 \le \mathrm{Pe}' \le 2\mathrm{EL}+1, \\ \widetilde{e} + \mathrm{fix}\left(\dfrac{\mathrm{Pe}' + b}{2}\right), & \text{if } -2\mathrm{EL}-1 \le \mathrm{Pe}' < 0, \\ e' - (\mathrm{EL}+1), & \text{if } \mathrm{Pe}' > 2\mathrm{EL}+1. \end{cases} \tag{32}$$

The algorithm decompresses through arithmetic coding to obtain the original overflow position map LM, processes the image E according to the LM, removes the preprocessing operation before embedding additional data, and obtains the original encrypted image C. The specific operation process is as follows:

$$C(i,j) = \begin{cases} C'(i,j) + \mathrm{EL} + 1, & \text{if } \mathrm{LM}(i,j) = 1, c'(i,j) \in [255 - 2\mathrm{EL} - 1, 255 - \mathrm{EL} - 1], \\ C'(i,j) + \mathrm{EL} - 1, & \text{if } \mathrm{LM}(i,j) = 1, c'(i,j) \in [\mathrm{EL}+1, 2\mathrm{EL}+1], \\ C'(i,j), & \text{otherwise.} \end{cases} \tag{33}$$

Image decryption is the reverse process of image encryption. In order to ensure lossless image decryption, the algorithm first generates a random sequence W of numbers 1 to t according to encryption key 2. After that, the algorithm scrambles the image block, and the image block returns to the original position. The algorithm generates a pseudo-random sequence R of length t according to encryption key 1 and then subtracts mod 256 from the image block $\{E, E, \ldots,\}$. As calculated by formula (34), the decrypted image block $\{B_1, B_2, \ldots, B_t\}$ is obtained, and the original image I is finally obtained.

$$\begin{gathered} B_S(i,j) = \left(E_s(i,j) - R_s\right) \mathrm{mod} 256 \\ 1 \le i \le u, 1 \le j \le v, 1 \le s \le t. \end{gathered} \tag{34}$$

4. Analysis of Communication Compression and Transmission in a Multimedia and Internet of Things Environment Integrating Scene Elements

The working principle of network multimedia communication system is as follows: when the caller initiates a call and the two sides establish a connection, the two sides can make voice calls normally, and when graphic communication is needed, the two sides can make graphic communication through the touch screen. First, the sender makes the graphic information input on the touch screen, and the graphic signal processing module collects the graphic signal and sends it to the microprocessor for processing. The microprocessor first judges whether the graphic data meets the sending criteria. If it does, the data is compressed and coded for one frame and then sent. The principle block diagram of the transmitting side of the network multimedia system is shown in Figure 10.

According to the idea of this algorithm, an adaptive transmission mode is designed for video transmission. The frame diagram is shown in Figure 11. A rate adjustment module is added to the video compression data sending end, and a network data monitoring module is added to the receiving end accordingly.

Next, this paper evaluates the communication compression transmission effect of the model proposed in this paper. This article collects multi-scene element data and combines it with the system for multimedia data transmission. After constructing the intelligent Internet of Things system, this paper evaluates the effect of communication compression transmission. The set evaluation parameters are the data compression effect and the communication transmission effect, and the results obtained are shown in Table 2 and Figure 12.

Through the above experimental research results, it can be seen that the communication compression transmission system in the multimedia and Internet of Things environment integrating scene elements proposed in this paper is more effective.

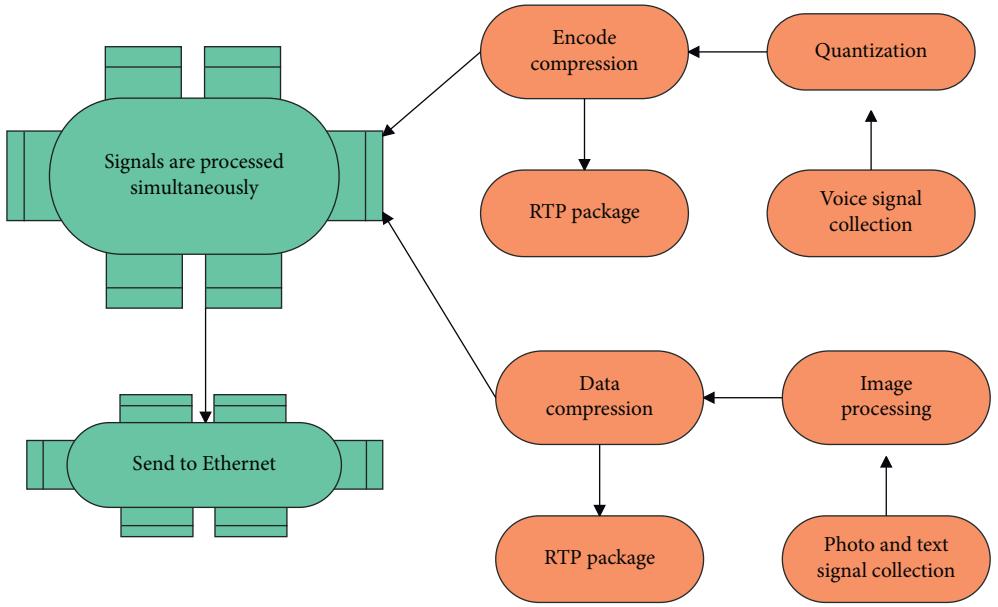


Figure 10: Block diagram of the system transmitter.

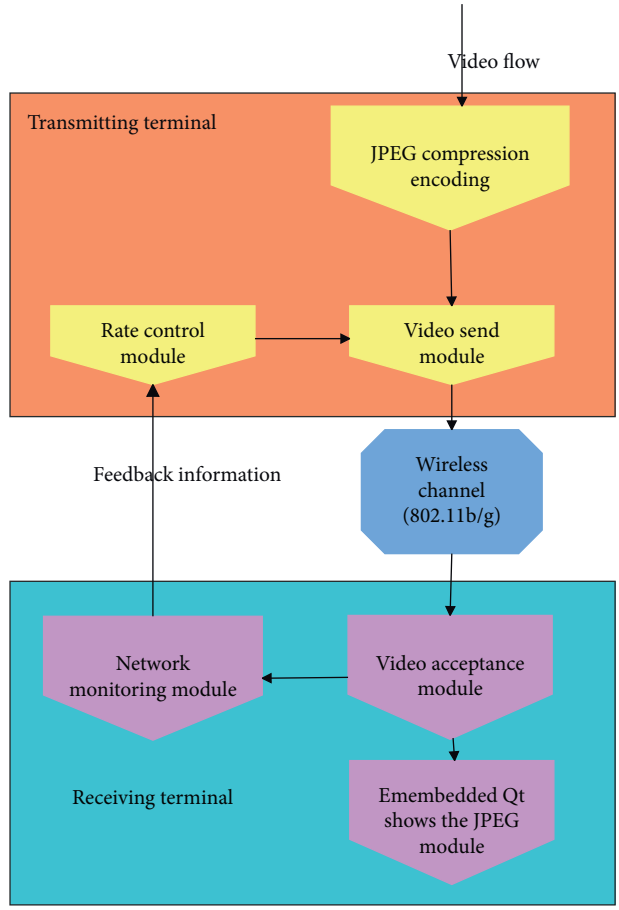


Figure 11: Video transmission system.

Table 2: Evaluation of the effect of the communication compression transmission system in the multimedia IoT environment integrating scene elements.

Number	Data compression	Data transmission	Number	Data compression	Data transmission	Number	Data compression	Data transmission
1	96.9	97.3	16	95.5	98.3	31	96.7	97.2
2	96.9	98.6	17	95.9	99.5	32	97.0	99.1
3	98.0	97.5	18	96.8	99.3	33	98.2	97.3
4	96.6	99.6	19	95.1	98.8	34	96.3	98.1
5	95.3	99.9	20	97.1	97.8	35	94.5	99.6
6	95.2	98.6	21	98.9	99.1	36	94.5	99.1
7	98.1	99.7	22	95.0	98.7	37	97.8	100.0
8	97.5	99.3	23	97.7	99.3	38	97.9	98.2
9	96.9	97.7	24	95.0	97.9	39	98.8	98.5
10	96.4	99.1	25	96.1	100.0	40	95.1	97.2
11	94.7	98.7	26	98.3	97.8	41	98.3	98.4
12	94.6	98.9	27	94.8	97.8	42	97.3	99.3
13	99.0	97.2	28	96.1	99.9	43	98.4	99.6
14	94.2	98.0	29	97.6	98.3	44	97.1	98.4
15	96.8	98.3	30	97.8	97.1	45	95.0	99.7

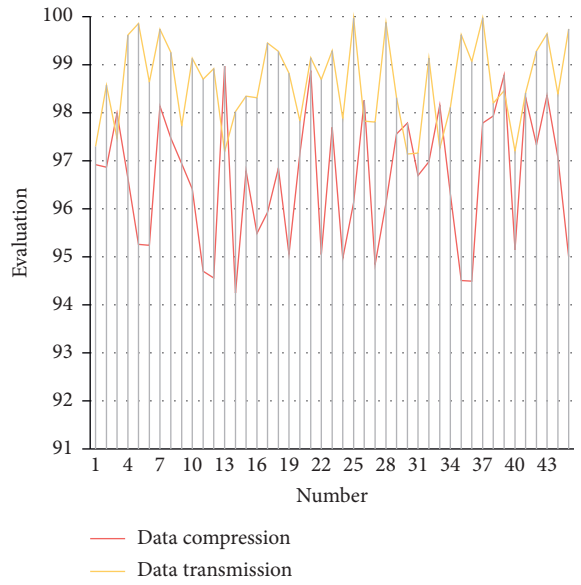


Figure 12: Statistical diagram of test data.

5. Conclusion

The development of multimedia technology cannot be separated from the development of other technologies. Multimedia network transmission technology is essentially the product of a combination of technologies, including multimedia computer technology and network communication technology, as well as multimedia data coding and decoding technology. In the actual multimedia transmission process, the transmission of many multimedia signals, including audio, video, and image data, still uses relatively old technical means based on analog signal transmission. The use of analog signals to transmit multimedia signals has a long history, and it brings the main problems of the high cost of the system, the need to lay a long line, the long construction cycle, the poor stability. Once the system becomes larger, more cable runs are often required, and these thick and long cable runs add to the complexity of the wiring. If the system structure changes, existing wiring also needs to be changed. This leads to very inconvenient maintenance of the whole system. With the support of Internet of Things technology, this paper studies multimedia communication transmission, improves the corresponding algorithm, builds an intelligent model, and enhances the compressed transmission of multimedia communication with fused scene elements.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This study was sponsored by Zhengzhou University.

References

[1] J. H. Abawajy and M. M. Hassan, "Federated internet of things and cloud computing pervasive patient health monitoring system," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 48–53, 2017.

[2] D. He, R. Ye, S. Chan, M. Guizani, and Y. Xu, "Privacy in the internet of things for smart healthcare," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 38–44, 2018.

[3] P. P. Ray, "Internet of things for smart agriculture: technologies, practices and future direction," *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 4, pp. 395–420, 2017.

[4] Y. A. Qadri, A. Nauman, Y. B. Zikria, A. V. Vasilakos, and S. W. Kim, "The future of healthcare internet of things: a survey of emerging technologies," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1121–1167, 2020.

[5] B. H. Dobkin, "A rehabilitation-internet-of-things in the home to augment motor skills and exercise training," *Neurorehabilitation and Neural Repair*, vol. 31, no. 3, pp. 217–227, 2017.

[6] J. Yao and N. Ansari, "Caching in energy harvesting aided Internet of Things: a game-theoretic approach," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3194–3201, 2018.

[7] J. E. Siegel, S. Kumar, and S. E. Sarma, "The future internet of things: secure, efficient, and model-based," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2386–2398, 2017.

[8] M. A. A. Elmagid, N. Pappas, and H. S. Dhillon, "On the role of age of information in the Internet of Things," *IEEE Communications Magazine*, vol. 57, no. 12, pp. 72–77, 2019.

[9] A. Sheth, U. Jaimini, and H. Y. Yip, "How will the internet of things enable augmented personalized health?" *IEEE Intelligent Systems*, vol. 33, no. 1, pp. 89–97, 2018.

[10] G. J. Joyia, R. M. Liaqat, A. Farooq, and S. Rehman, "Internet of Medical Things (IOMT): applications, benefits and future challenges in healthcare domain," *Journal of Communications*, vol. 12, no. 4, pp. 240–247, 2017.

[11] N. Kshetri, "The evolution of the internet of things industry and market in China: an interplay of institutions, demands and supply," *Telecommunications Policy*, vol. 41, no. 1, pp. 49–67, 2017.

[12] S. Siboni, V. Sachidananda, Y. Meidan et al., "Security testbed for Internet-of-Things devices," *IEEE Transactions on Reliability*, vol. 68, no. 1, pp. 23–44, 2019.

[13] Y. Yang, M. Zhong, H. Yao, F. Yu, X. Fu, and O. Postolache, "Internet of things for smart ports: technologies and challenges," *IEEE Instrumentation and Measurement Magazine*, vol. 21, no. 1, pp. 34–43, 2018.

[14] Z. Li, Y. Liu, A. Liu, S. Wang, and H. Liu, "Minimizing convergecast time and energy consumption in green Internet of Things," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 3, pp. 797–813, 2018.

[15] P. P. Ray, "A survey on Internet of Things architectures," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, 2018.

[16] M. Mayer and A. J. Baeumner, "A megatrend challenging analytical chemistry: biosensor and chemosensor concepts ready for the internet of things," *Chemical Reviews*, vol. 119, no. 13, pp. 7996–8027, 2019.

[17] M. Saez, F. P. Maturana, K. Barton, and D. M. Tilbury, "Real-time manufacturing machine and system performance monitoring using internet of things," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1735–1748, 2018.

[18] V. Jagadeeswari, V. Subramaniyaswamy, R. Logesh, and V Vijayakumar, "A study on medical Internet of Things and Big Data in personalized healthcare system," *Health Information Science and Systems*, vol. 6, no. 1, pp. 14–20, 2018.

[19] S. Smys, A. Basar, and H. Wang, "Hybrid intrusion detection system for internet of Things (IoT)," *Journal of ISMAC*, vol. 2, no. 4, pp. 190–199, 2020.

[20] I. Butun, P. Österberg, and H. Song, "Security of the internet of things: vulnerabilities, attacks, and countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 616–644, 2019.

[21] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How can heterogeneous internet of things build our future: a survey," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 3, pp. 2011–2027, 2018.

[22] A. Heiskanen, "The technology of trust: how the Internet of Things and blockchain could usher in a new era of construction productivity," *Construction Research and Innovation*, vol. 8, no. 2, pp. 66–70, 2017.

[23] M. Wolf and D. Serpanos, "Safety and security in cyber-physical systems and internet-of-things systems," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 9–20, 2017.