The Application of Optimized JPEG-LS Algorithm in Efficient Transmission of Multi-Spectral Images

Huanping Hu, Xing Wang
Information Engineering College, Jiangxi Polytechnic University, Jiujiang, 332000, China

***Abstract*—Currently, multi-spectral image transmission faces challenges such as high storage costs and low transmission efficiency. Although various technologies are attempted to solve these problems recently, such as improving encoding methods in some algorithms, there are still issues such as insufficient compression ratio and slow processing speed. Therefore, the research focuses on optimizing the Joint Photographic Experts Group Lossless Standard (JPEG-LS) algorithm and constructing a multi-spectral image processing system. Regarding the JPEG LS algorithm process, improvements are made to the conventional encoding method by adopting sub-block compression strategy and block compression algorithm based on dynamic image bit width. The results show that the optimized JPEG LS algorithm has an average compression ratio of 5.81, which is higher than the comparison algorithm. The average compression time is 0.35 seconds, the average peak signal-to-noise ratio (PSNR) is 43.6, and the average structural similarity (SSIM) is 0.97, all of which are better than the comparison algorithm. In terms of system performance, stability testing of each module shows that the overall system tends to be stable, and the resource utilization rate of the image compression module is low, with a large resource margin that can meet practical application needs.**

***Keywords*—Multi-spectral; image transmission; JPEG-LS algorithm; compression ratio; signal-to-noise ratio**

I. INTRODUCTION

In the current era of rapid technological development, multi-spectral images play an indispensable role in many cutting-edge fields due to their ability to simultaneously obtain information about target objects in multiple spectral bands [1]. In medical imaging diagnosis, multi-spectral images can help doctors more accurately identify diseased tissues and improve the accuracy of disease diagnosis. In terms of ecological environment monitoring, they can comprehensively evaluate changes in forest cover and water pollution levels, providing strong basis for ecological protection decisions. In the field of intelligent security, their unique spectral characteristics can be utilized to effectively identify disguised targets and enhance the reliability of security systems [2]. However, with the continuous expansion of multi-spectral image application scenarios, the rapid increase in data volume has made its transmission efficiency a bottleneck that restricts further development.

Currently, research on multi-spectral image transmission has been explored in multiple directions. Some scholars have improved the compression efficiency of data to a certain extent by designing new transformation encoding methods. Some studies also attempt to combine machine learning techniques to intelligently extract and process image features in order to optimize the transmission process [3]. However, there are still significant shortcomings in existing research. On the one hand, existing compression algorithms struggle to achieve an ideal balance between compression ratio and image quality, and excessive compression often leads to severe loss of image details, resulting in damage to key information. On the other hand, when facing multi-spectral images with complex spectral features and diverse spatial structures, most algorithms have poor universality and cannot adaptively adjust to different image characteristics.

This study focuses on optimizing the Joint Photographic Experts Group Lossless Standard (JPEG-LS) algorithm in order to overcome the aforementioned challenges. By deeply mining the algorithm core and closely integrating the unique attributes of multi-spectral images, the algorithm is customized and improved. It is expected to significantly improve the compression ratio without compromising image quality, while enhancing the algorithm's adaptability to various types of multi-spectral images, thus laying a solid foundation for the deep application of multi-spectral images in various fields.

The innovation of this study lies in addressing the shortcomings of the JPEG-LS algorithm in multi-spectral image compression. By improving the conventional encoding method, a sub-block compression strategy and a dynamic image bit width-based block compression algorithm are proposed, and the algorithm flow is optimized. Moreover, the study designs a multi-spectral image processing system that integrates optimization algorithms to jointly improve the storage and transmission efficiency of multi-spectral images from both algorithm and system levels.

The research is divided into four sections, with Section II being a summary of the relevant work. Section III is about optimizing algorithms and system design processes. Section IV is the performance analysis of algorithms and systems. Section V is a discussion of the research results, and Section VI is a summary of the entire study.

II. RELATED WORKS

Recently, with the continuous progression of image processing technology, many scholars have devoted themselves to researching how to optimize image transmission algorithms to improve image transmission efficiency [4-5]. Zhang et al. proposed a spatial pilot-assisted fast adaptive framework to address the stability issue of multi-mode fiber image transmission. This framework could adaptively adapt to changes in physical channels and achieve online model updates

during continuous transmission. The experiment outcomes indicated that this approach could achieve a transmission accuracy of over 92% within a few hours, and the pilot frame overhead was about 2% [6]. Wu et al. raised an image transmission method based on semantic segmentation, which could distinguish between regions of interest and non-regions of interest, and achieve high-quality transmission of regions of interest with low communication overhead. The experiment outcomes indicated that this method significantly improved performance compared to existing semantic communication methods and traditional methods [7]. Khandelwal et al. proposed a secure image steganography technique based on discrete wavelet transform and deep learning to improve the quality of steganographic images and extracted secret images. The experiment outcomes indicated that this approach had a PSNR of 51.66 to 38.69 dB and a SSIM index of 0.99, demonstrating high robustness [8]. Gupta et al. proposed an effective approach for encrypting images based on a mixture of watermarking and cryptographic techniques, which was based on two-level security and was used to securely and error free transmit images between devices supporting the Internet of Things. The experiment outcomes indicated that this approach had strong resistance to various types of password attacks [9]. Al Kadhimi et al. proposed a transmission system based on prototype low-density parity check codes and orthogonal frequency division multiplexing for underwater image transmission problems. The experimental results showed that the system outperformed traditional polar cyclic redundancy check and turbo code in terms of performance, and the received image reconstruction effect was better [10].

JPEG-LS is a lossless compression algorithm that predicts images by utilizing adjacent pixels that have already been encoded. It is suitable for scenes that require high image quality. Sun et al. raised a lossless image compression and encryption algorithm that combines JPEG-LS, neural networks, and hyper chaotic mapping to improve the prediction performance of edge texture regions. They also adopted a threshold segmentation method to further improve the image compression ratio. Experiment outcomes indicated that the algorithm had a good compression ratio and could resist various attacks [11]. Hua et al. optimized the JPEG-LS algorithm for compressing the intermediate data layer in neural networks, utilizing computational memory technology for global prediction and efficient compression. The results indicated that the compression ratio reached a high level and the hardware cost was relatively low [12]. Rahman et al. proposed a JPEG-LS algorithm that reduced image dimensionality and utilized prediction techniques, followed by encoding prediction errors using Huffman coding. The experiment outcomes indicated that the algorithm performed well in terms of average code length, compression ratio, encoding time, decoding time, and other aspects [13]. Al Qerom et al. proposed a new LICA-CS algorithm that optimized compression results by strategically minimizing inter channel correlation, and used a new subtraction method to compress image data column by column, successfully solving the problem of similarity and proximity of pixel values in adjacent columns, significantly reducing image size by 71%. Experimental results showed that this algorithm outperformed existing algorithms in terms of compression rate, while exhibiting significant improvements in execution time, with an average compression and decompression process of 1.93 seconds [14]. Hamano et al. applied the JPEG-LS algorithm to encrypted images and analyzed its impact on image classification. The outcomes revealed that the JPEG-LS algorithm could notably reduce the data volume of encrypted images while maintaining classification accuracy. When the quality factor was 85, the classification accuracy could be maintained at over 98%, and the image data volume could be reduced by over 90% [15].

In summary, there have been various methods to improve the stability, security, and efficiency of image transmission technology by optimizing the JPEG-LS algorithm or other image processing techniques. However, these methods still need further optimization in the efficient transmission application of multi-spectral images. Therefore, the research optimizes the JPEG-LS algorithm and applies it to efficient transmission of multi-spectral images, in order to improve transmission efficiency and stability while ensuring image quality. The innovation of the research lies in the use of sub block compression strategy and dynamic image bit width improvement to improve compression efficiency of the JPEG-LS algorithm.

III. Methods and Materials

A. Optimizing the Design of the JPEG-LS Algorithm

The JPEG-LS algorithm is suitable for compressing grayscale images and multi-spectral images, and its process is in Fig. 1. The JPEG-LS algorithm first calculates the gradient of the image, and then determines whether it is a flat area based on the gradient. If it is a flat area, conventional encoding is performed and output [16-17]. If it is not a flat area, it enters the adaptive correction step, predicts through the median predictor, processes by the context modeler, and finally performs Golomb encoding and run length encoding to output the compressed image.

The JPEG-LS algorithm mainly includes two methods: run length encoding and conventional encoding. However, due to the high amount of image noise and large fluctuations in grayscale values, run length encoding is not suitable in this situation. Therefore, the research mainly focuses on improving conventional encoding methods. In the conventional encoding process, context modeling constructs a model based on the surrounding pixel values to better predict the current pixel value. Firstly, each pixel in the image is sampled, and its surrounding pixel values are referenced to establish a probability distribution model for predicting the possible values of the current pixel. The local gradient calculation is in Eq. (1).

$$\begin{cases} D_1 = x_4 - x_2 \\ D_2 = x_2 - x_3 \\ D_3 = x_3 - x_1 \end{cases} \quad (1)$$

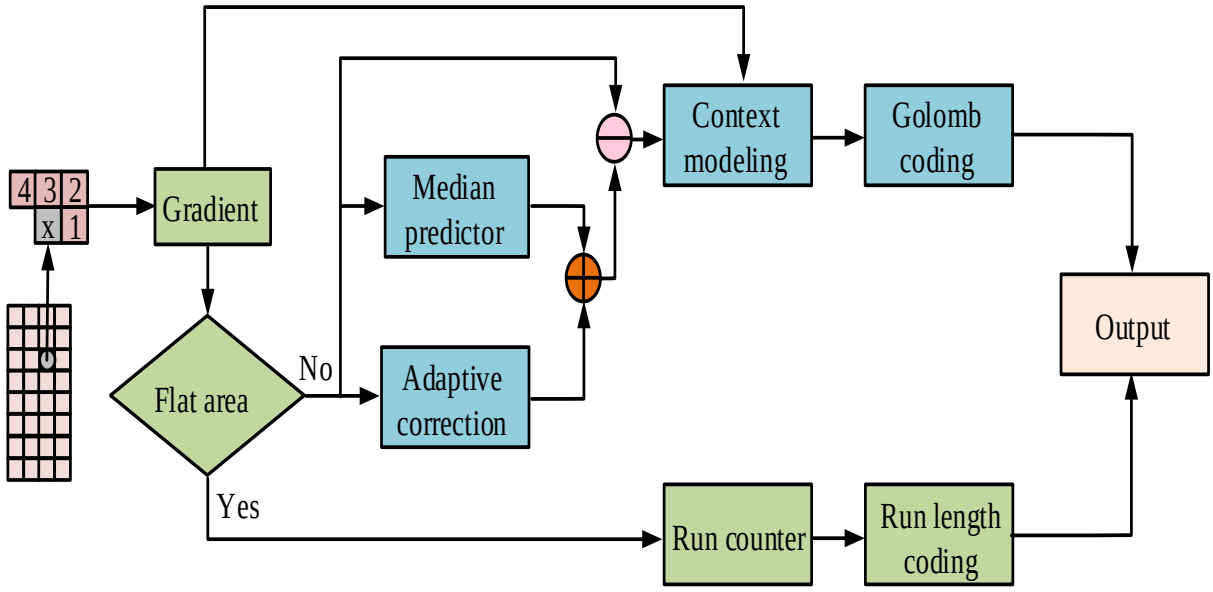

Fig. 1. JPEG-LS algorithm flow.

In Eq. (1), the local gradient values are D_1, D_2, and D_3, and the pixel values at the four pixel positions are x_1, x_2, x_3, and x_4, respectively. The quantization standard for local gradient values is shown in equation (2).

$$Q_i = sign(D_i).\begin{cases} 0, & D_i = 0 \\ 1, & 0 < |D_i| < T_1 \\ 2, & T_1 < |D_i| < T_2 \\ 3, & T_2 < |D_i| < T_3 \\ 4, & T <_3 |D_i| \end{cases}, i = 1,2,3 \quad (2)$$

In Eq. (2), the quantized gradient value is Q_i, and the non-negative threshold is $[T_1, T_2, T_3]$. The calculation for non-negative threshold is shown in Eq. (3).

$$\begin{cases} T_1 = near + 3(bpp - 7) \\ T_2 = 2near + 7(bpp - 7) \\ T_3 = 3near + 21(bpp - 7) \end{cases} \quad (3)$$

In Eq. (3), the micro loss is $near$ and the image bit width is bpp. After quantifying each gradient, it can be fused into a whole. Meanwhile, a context parameter address index needs to be set, which represents a specific symbol. This address index can be used to define predictive information. Secondly, prediction is based on the context model to calculate the predicted value of the current pixel value, and the prediction difference quantization process is shown in Eq. (4).

$$Err' = \begin{cases} \dfrac{Err + near}{2near + 1}, & Err > 0 \\ -\dfrac{Err + near}{2near + 1}, & Err \le 0 \end{cases} \quad (4)$$

In Eq. (4), the predicted difference is Err, and its quantized value is Err'. Predictive encoding is the process of predicting the current pixel value based on known pixel values, and then encoding the prediction error. When performing predictive encoding, it is necessary to utilize context related parameters to better predict and encode the next pixel. These context-related parameters can include known pixel values, neighborhood information of pixels, and so on. By updating these parameters, the precision and effectiveness of predictive coding can be enhanced. When encoding prediction errors, it is necessary to convert them into a one-sided exponential distribution. This is because the unilateral exponential distribution has a smaller variance, which can better represent the noise and detail information in the image [18]. Meanwhile, by taking the modulus of the error, negative modulus can be avoided, thereby ensuring the stability of the encoding. Finally, the Gloomb encoding method is used to convert the error of the one-sided exponential distribution into a bitstream for storage and transmission. After encoding the current pixel, it is necessary to update the context-related parameters in order to better predict and encode the next pixel.

The JPEG-LS encoding method has the problem of error sensitivity. To solve this problem, a sub-block compression strategy is proposed in the encoding process, which divides the image into independent non-overlapping sub-blocks for compression. However, this method may have an impact on compression performance and requires further optimization [19]. In traditional methods, dynamic range is generally calculated by quantifying bit width. However, the research has proposed a block compression algorithm based on dynamic image bit width, which expands statistics on the local dynamic range of each image sub-block to improve compression performance. The process of local dynamic range statistics is shown in Fig. 2.

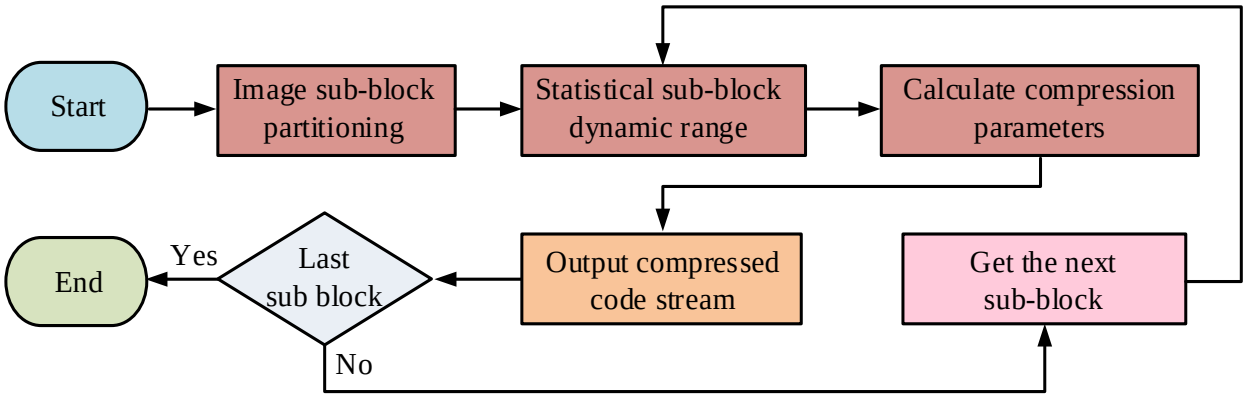

Fig. 2. Local dynamic range statistical process.

In Fig. 2, in the initial stage, sub-block partitioning is first carried out for the image. After completion, statistical sub-block dynamic range is analyzed. After the statistical work is completed, the compression parameters are calculated. Then, it is necessary to determine whether the processed sub-block is the last sub-block. If the determination result is not the last sub-block, then the next sub-block is extracted and the operation continues according to the steps described earlier. If it is determined as the last sub-block, the compressed stream is output and the entire process ends. The calculation of dynamic range parameters is shown in Eq. (5).

$$bpp = \log_2(\max G - \min G) \tag{5}$$

In Eq. (5), after the image is segmented, its maximum grayscale value is $\max G$ and its minimum grayscale value is $\min G$. The minimum grayscale value can be considered as 0, resulting in a simplified dynamic range parameter as shown in Eq. (6).

$$bpp = floor(\log_2(\max G) + 1) \tag{6}$$

In Eq. (6), the rounding down operation is $floor$. After the dynamic range of the segmented image is calculated, the maximum value of independently-encoded pixel values is calculated as shown in Eq. (7).

$$\max P = 2^{bpp} - 1 \tag{7}$$

In Eq. (7), the independently-encoded pixel value is P. The calculation of the quantization range of prediction error is shown in Eq. (8).

$$Range = \left| \frac{\max P + 2near}{2near + 1} \right| + 1 \tag{8}$$

In equation (8), the quantization range of prediction error is $Range$. In the Golomb encoding algorithm, the encoding length limit is shown in Eq. (9).

$$Limit = 2(bpp + (8, bpp)) \tag{9}$$

In Eq. (9), the parameter $Limit$ plays an important role in controlling the encoding length and optimizing the encoding efficiency in Golomb limited length encoding. The initial value of the context is calculated as shown in Eq. (10).

$$A_0 = \max\left(1 + \left| \frac{32 + Range}{64} \right|\right) \tag{10}$$

In Eq. (10), the initial value of the context is A_0. The optimized JPEG-LS algorithm flow is shown in Fig. 3.

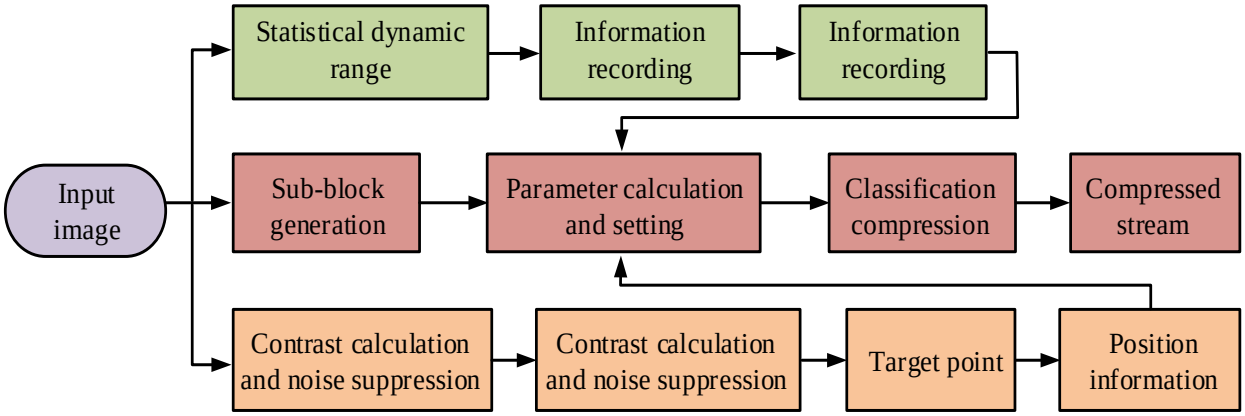

Fig. 3. Optimized JPEG-LS algorithm process.

In the optimized JPEG-LS algorithm process, first, one frame of image is input. Next, the image is divided into sub-blocks, the dynamic range of the sub-blocks is calculated, and the dynamic range sub-block information is recorded. Afterwards, image sub-blocks are generated and parameter calculations and settings are performed based on them. The calculation for image sub-blocks is shown in Eq. (11).

$$K = \frac{M \times N}{m \times n} \tag{11}$$

In Eq. (11), the image size is $M \times N$, the sub-block size is $m \times n$, and the number of sub-blocks is K. In another parallel branch, contrast calculation and noise suppression processing are performed on the input image to calculate micro contrast. The contrast calculation of image sub-blocks is shown in Eq. (12).

$$C = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \left(P_{ij} - \bar{P} \right)^2}{m \times n} \tag{12}$$

In Eq. (12), the sub-block contrast is C, the pixel value of pixel (i, j) in the sub-block is P_{ij}, and the average pixel value of the sub-block is $\bar{P}$. Then, the suspected target points are determined through multi-scale and multi-threshold segmentation, and the positions of the suspected target points are recorded. Then the JPEG-LS algorithm is applied to classify and compress the target and background, and finally a compressed stream is output. The sub-block occupancy of false alarm images is in Eq. (13).

$$F_p = \frac{k_f}{K} \tag{13}$$

In Eq. (13), the proportion of false alarm image sub-blocks is F_p, and the number of false alarm sub-blocks is k_f. The target sub-block is represented by Eq. (14).

$$T_p = \frac{k_t}{K} \tag{14}$$

In Eq. (14), the proportion of target sub-blocks is T_p, and the number of target sub-blocks is k_t. The performance of compression algorithms is affected by object detection algorithms, and low false alarm rates help improve compression ratios.

In summary, the optimized JPEG-LS algorithm is designed to address the characteristics of multi-spectral images. Due to the high level of image noise and large fluctuations in grayscale values, the research focuses on improving conventional encoding methods. A probability distribution model is constructed through context modeling to predict pixel values, and the prediction error is converted into a one-sided exponential distribution during encoding to ensure stability. To address the issue of error sensitivity, a sub-block compression strategy is adopted, and a block compression algorithm based on dynamic image bit width is proposed to improve compression performance. The optimization process also includes sub-block partitioning of the input image, dynamic range statistics, contrast calculation, and noise suppression processing, ultimately compressing the output stream for target and background classification.

B. Design of Multi-Spectral Image Processing System

After optimizing the JPEG-LS algorithm, a multi-spectral image processing system is further designed to compress and encode images using the optimized JPEG-LS algorithm, in order to reduce storage and transmission costs. The framework of the multi-spectral image processing system is in Fig. 4. The system consists of control, image acquisition, server, image processing, and client modules. The control module obtains real-time status information and sends control signals. The image acquisition module receives the signal and collects image data, which is then transmitted to the server. The image processing module integrates optimization algorithms to compress the image and transmits it to the client via USB. The client interacts with the server to ensure correct reception and processing of the data. The image processing module includes data transmission, storage, and compression sub-modules, relying on relevant chips and platforms, combined with reversible component transformation and algorithms to achieve lossless image processing.

In Fig. 4, the control module plays a role in obtaining and controlling real-time status information, including position, angle, and operating status, through sockets. Based on these status information, the control module will issue corresponding control signals for adjusting posture and other operations. Meanwhile, there is data exchange between the control module and the image acquisition module, which sends control signals to the image acquisition module. After receiving the control signal from the control module, the image acquisition module begins to collect image data [20]. The image data it collects will be transmitted to the server. The image processing module plays a critical processing role in the entire system, integrating the optimized JPEG-LS algorithm. It receives control signals from the server and processes image data based on these control signals. The image processing module compresses the image data to reduce the amount of data transmitted during the transmission process. The compressed stream-processed image data are transmitted to the client through a USB interface. The server receives image data from the image acquisition module and sends control signals to the image processing module [21-22]. The client is the terminal of the system, which receives compressed stream image data transmitted through USB from the image processing module. Meanwhile, the client and server interact through real-time transmission protocol control signals to ensure that the client can correctly receive and process image data. The process of the control module is shown in Fig. 5.

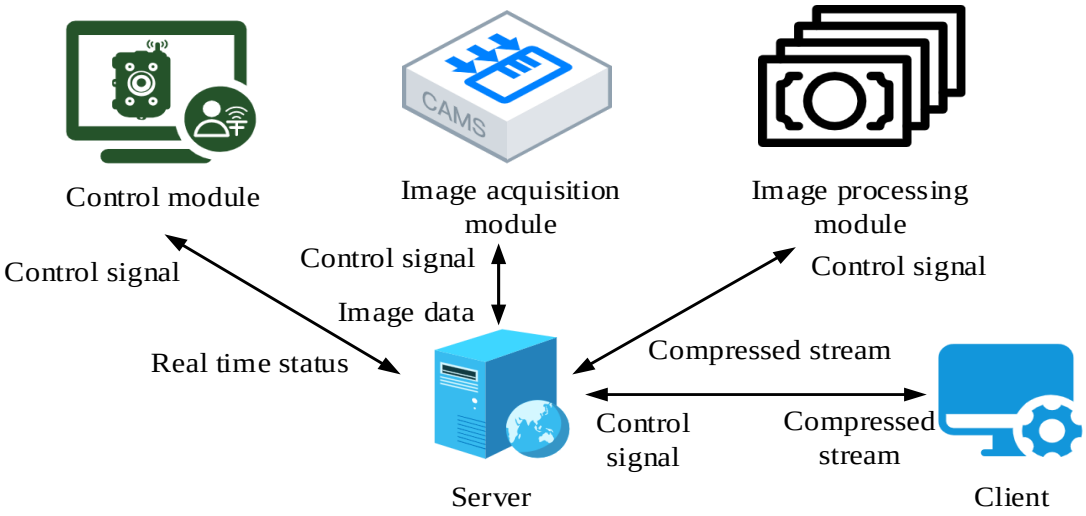

Fig. 4. Framework of the system.

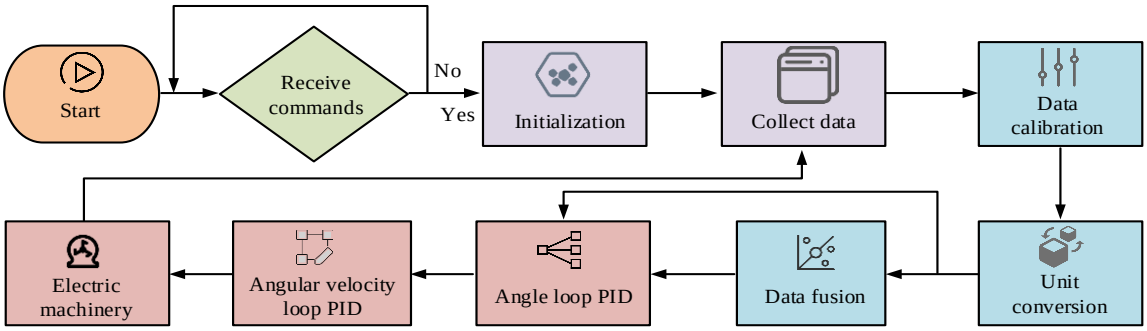

Fig. 5. Process of control module.

The process begins with determining whether an upper level command has been received, and if not, it remains in a waiting state. If received, it enters the initialization phase, during which offset calculation will be performed. After initialization is complete, data collection begins, followed by calibration of acceleration and angular velocity data. Next is the unit conversion stage, where data fusion is performed after conversion using the Mahony filtering method? After data fusion, the expected angle and current angle are obtained separately, and the expected angular velocity is calculated through the angle loop PID. Based on the current angular velocity, the pitch angle motor, roll angle motor, and yaw angle motor are finally controlled through PWM and GPIO after PID processing in the angular velocity loop.

The image processing module can be mainly divided into three sub-modules: data transmission, data storage, and image compression. In the data transmission module, the USB interface and CY7C68013A chip are fully utilized to efficiently transmit image data between the server and FPGA. This chip supports USB 2.0 protocol and its development toolkit is also very complete, providing reliable guarantee for stable data transmission. The data storage module uses DDR2 SDRAM, which can properly store multi-spectral images and compressed bitstreams, ensuring secure storage and easy access to data at any time. The image compression module is shown in Fig. 6.

The image compression module relies on FPGA high-performance platform, combined with reversible component transformation and optimized JPEG-LS algorithm, to perform lossless compression of multi-spectral images in space and spectrum. After decoding, the image can be completely consistent with the original image, thus achieving lossless processing of the image.

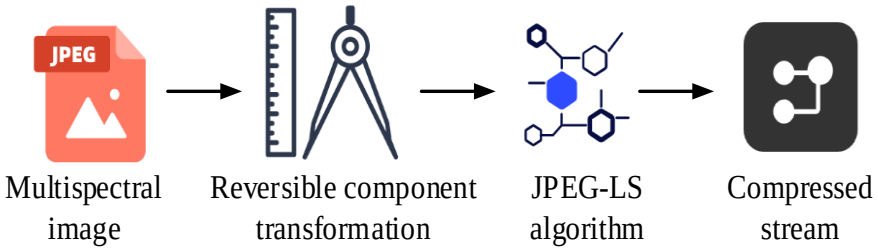

Fig. 6. Image compression module.

IV. RESULTS

A. Performance Analysis of Optimizing the JPEG-LS Algorithm

The hardware environment configuration for algorithm performance analysis adopted Intel (R) Core (TM) i7-6700HQ core, with a CPU frequency of 2.6GHz, 8GB memory, 1TB hard drive, and ran on the Windows 10 operating system. The software environment was configured as MATLAB 2022. During the block compression process of the JPEG-LS algorithm, the compression ratio and prediction error curves are in Fig. 7. Fig. 7 (a) indicates the compression ratio under different block sizes and micro loss degrees. As the block size increased, the compression ratio of the image gradually increased. As the degree of micro loss increased, the compression ratio of the image also gradually increased. When the micro loss was 1, the compression ratios for block sizes of 8×32, 64×64, and 515×512 were 1.61, 3.13, and 5.55, respectively. When the block size was 512×512, the image compression ratios corresponding to micro loss degrees of 0, 1, 2, and 3 were 3.12, 5.55, 6.51, and 7.32, respectively. Fig. 7 (b) shows the prediction error curves for full image compression and block compression. The prediction error range for full image compression was [-40,40], and for block compression was [-45,45]. The above data indicated that block compression had a larger fluctuation range of prediction error compared to full image compression. In the process of block compression, the boundaries of each block and the local characteristics within each block may introduce more uncertainty, making the distribution of prediction errors more dispersed and wider. However, full image compression may be relatively more stable during the prediction process due to considering the global characteristics of the entire image, and the range of prediction errors may be relatively small. Therefore, it is necessary to introduce dynamic range parameters in the block compression process of the JPEG-LS algorithm to obtain an optimized version of the algorithm.

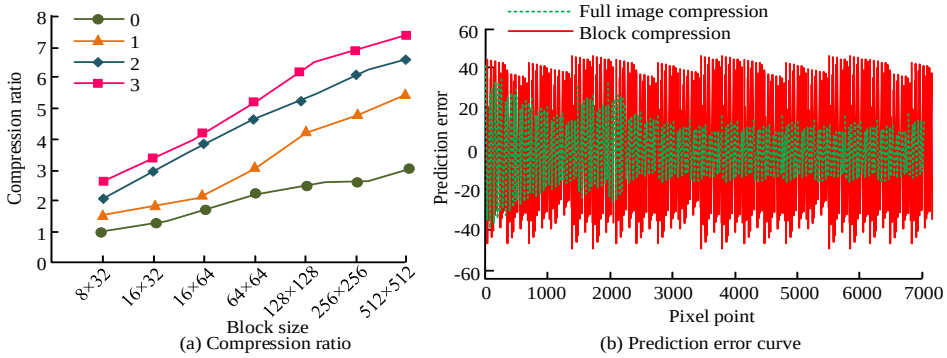

Fig. 7. Compression ratio and prediction error curve.

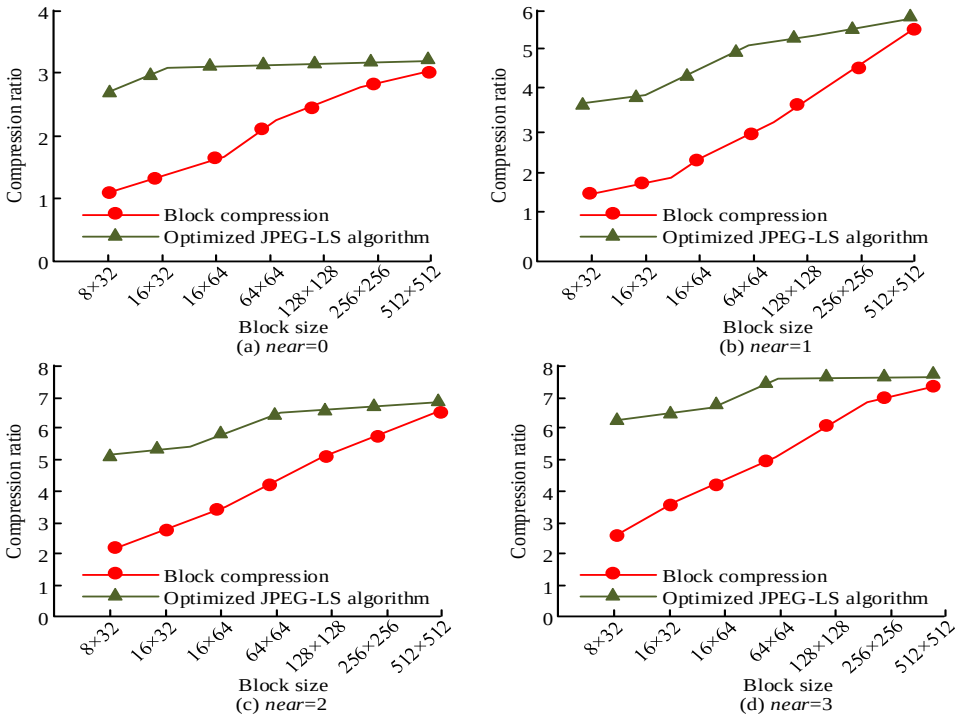

Fig. 8. Comparison of compression effects of optimized JPEG-LS algorithm.

The compression effect comparison of the optimized JPEG-LS algorithm is in Fig. 8. In Fig. 8, as the block size increased, the compression ratio of block compression and optimized JPEG-LS algorithm gradually increased, and the compression ratio of optimized JPEG-LS algorithm was larger, but the distance between the two gradually decreased. In Fig. 8 (a), the micro loss was 0. When the block size was 8 × 32, the compression ratios of block compression and optimized JPEG-LS algorithm were 1.05 and 2.73, respectively. When the block size was 128 × 128, the compression ratios of block compression and optimized JPEG-LS algorithm were 2.75 and 3.11, respectively. In Fig. 8 (b), the micro loss was 1. When the block size was 64 × 64, the compression ratios for block compression were 3.13, and the optimized compression ratio for the JPEG-LS algorithm was 5.06. In Fig. 8 (c), the micro loss was 2. When the block size was 8×32, the compression ratios of block compression and optimized JPEG-LS algorithm were 2.08 and 5.15, respectively. In Fig. 8 (d), when the micro loss was 3 and the block size was 16×64, the compression ratios of block compression and optimized JPEG-LS algorithm were 5.31 and 7.48, respectively. The results indicated that the optimized JPEG-LS algorithm had a high compression ratio and exhibited relatively stable performance with changes in block size.

In order to verify the stability of the algorithm, the experiment was tested by changing the image type, image resolution, and noise level. The test results are shown in Table I. From the perspective of image type, the compression ratio of texture image was the highest (5.01), followed by landscape image (4.23) and figure image (3.87). In terms of image resolution, as the resolution increased, the compression ratio increased from 3.56 at 640*480 to 4.78 at 1920*1080. In terms of noise level, the lower the noise, the higher the compression ratio, which was 3.98 at low noise and 3.21 at high noise. The results showed that the compression performance of JPEG-LS algorithm was affected by many factors. In terms of image types, images with rich textures were easier to obtain higher compression ratio. The higher the image resolution was, the higher the compression ratio could be achieved. The noise level was negatively correlated with the compression ratio, and the lower the image noise, the higher the compression ratio. In practical application, the compression effect of JPEG-LS algorithm could be estimated and optimized according to image characteristics such as type, resolution, and noise.

TABLE I. TEST RESULTS OF THE ALGORITHM UNDER DIFFERENT PARAMETERS

Parameter type	Parameter value	Compression ratio	Prediction error range
Image type	Landscape	4.23	[-35, 35]
	Character	3.87	[-30, 30]
	Texture	5.01	[-40, 40]
Image resolution	640*480	3.56	[-32, 32]
	1280*720	4.12	[-38, 38]
	1920*1080	4.78	[-42, 42]
Noise level	Low noise (mean 0, variance 0.01)	3.98	[-33, 33]
	Medium noise (mean 0, variance 0.05)	3.65	[-36, 36]
	High noise (mean 0, variance 0.10)	3.21	[-40, 40]

To confirm the progressiveness of the optimized JPEG-LS algorithm proposed by the research, the experiment compared the algorithms in study [12], study [13], study [23] and reference [24], and the comparison of compression ratio and compression time of different algorithms is shown in Fig. 9. Fig. 9 (a) shows a comparison of compression ratios for various algorithms. The optimized JPEG-LS algorithm had an average compression ratio (ACR) of 5.81, the algorithm in study [12] had an ACR of 5.56, and the algorithm in study [13] had an ACR of 5.46. The ACR of the algorithm in reference [23] was 5.76, and that of the algorithm in study [24] was 5.74. Fig. 9 (b) shows a comparison of compression times for different algorithms. The optimized JPEG-LS algorithm had an average compression time (ACT) of 0.35s, the algorithm in study [12] had an ACT of 0.37, and the algorithm in study [13] had an ACT of 0.35s. The ACT of the algorithm in study [23] and the algorithm in study [24] was 0.36.

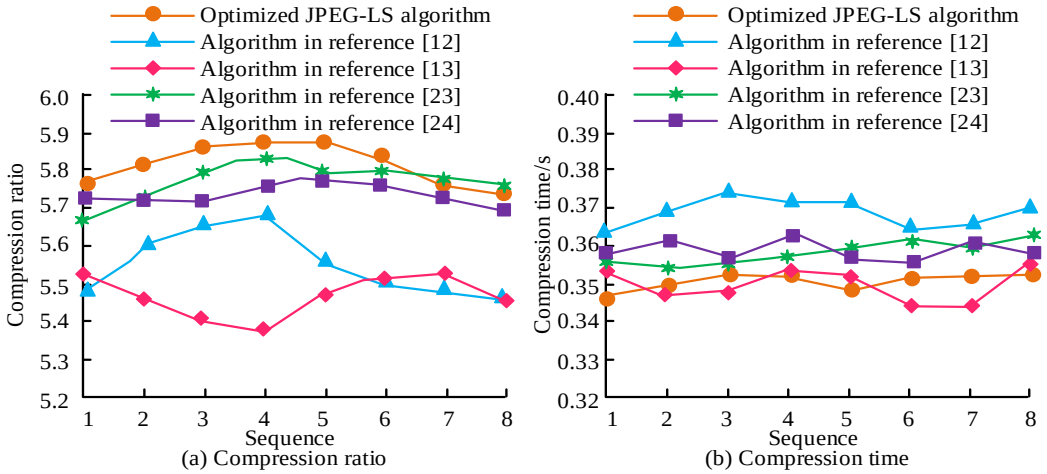

Fig. 9. Comparison of compression ratios and compression times for different algorithms.

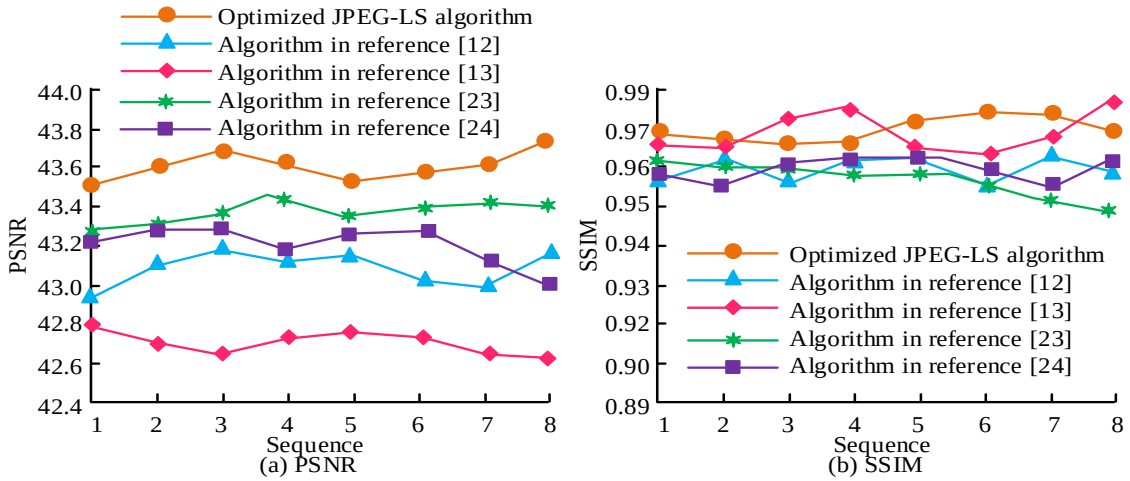

Fig. 10. Comparison of PSNR and SSIM for different algorithms.

The comparison of PSNR and SSIM of different algorithms is shown in Fig. 10. Fig. 10 (a) shows the comparison of PSNR. The PSNR of the optimized JPEG-LS algorithm was higher than 43, with an average of 43.6. The average PSNR of the algorithm in study [12] was 43.1, and the average PSNR of the algorithm in study [13] was 42.7. The average PSNR of the algorithm in study [23] was 43.4, and the average PSNR of the algorithm in study [24] was 43.2. Fig. 10 (b) shows the comparison of SSIM. The average SSIM of the optimized JPEG-LS algorithm was 0.97, the average SSIM of the algorithm in study [12] was 0.97, and the average SSIM of the algorithm in study [13] was 0.96. The average SSIM of the algorithm in study [23] was 0.95, and the average SSIM of the algorithm in study [24] was 0.96. Compared with existing algorithms in studies [12], [13], [23], and [24], the optimized JPEG-LS algorithm exhibited many advantages. In terms of compression ratio, this algorithm was higher than existing algorithms. In terms of compression time, it was comparable to various comparison algorithms. In terms of image quality assessment, its PSNR was superior to existing algorithms, and its SSIM was not inferior or even better. Overall, the optimized JPEG-LS algorithm performed well in compression performance and image quality preservation, making data processing more efficient.

B. *Performance Analysis of Multi-Spectral Image Processing System*

After the construction of the multi-spectral image processing system was completed, its performance was tested and analyzed. The stability test results of each module are in Fig. 11. Fig. 11 (a) shows the stability test results of the control module. As the number of tests increased, the stability time of the control module fluctuated, with an average stability time of 0.21s. Due to the impact of scheduling and resource allocation of different tasks within the system when processing various control instructions, the stability time fluctuated. Fig. 11 (b) shows the stability test results of the image acquisition module. As the number of tests increased, the stabilization time gradually decreased. After 30 tests, the image acquisition module took 0.18 seconds to stabilize. As the testing progressed, the module gradually adapted to the working environment and workflow, resulting in improved collection efficiency and reduced time consumption. Fig. 11 (c) shows the stability test results of the image processing module. As the number of tests increased, the stability time showed a fluctuating downward trend. In the first 30 tests, the average stability time of the image processing module was 0.19 seconds. The image processing process involved multiple algorithms and complex operations, and its stability was affected by various factors such as data volume and algorithm complexity, resulting in fluctuations in stability time. However, the overall downward trend may be due to the system's adaptive adjustment of resource management and algorithm execution during operation, which improved processing efficiency. Overall, all modules tended to stabilize to a certain extent, providing a certain guarantee for the normal operation of the multi-spectral image processing system.

The resource utilization of FPGA in the image compression module is shown in Table II. The usage of Lookup Table (LUT) was 27582, the total allowed resources of FPGA system was 203800, and the percentage of FPGA system was 13.5%. The usage of Block Random Access Memory (BRAM) was 38, the total allowed resources were 455, and the percentage of FPGA system was 8.4%. The usage of Digital Signal Processor (DSP) was 69, the total allowed resources were 840, and the percentage of FPGA system was 8.2%. The usage of input/output (I/O) was 57, the total allowed resources ere 500, and the percentage of FPGA system was 11.4%. The usage of Hybrid Memory Cube (HMC) was 2, the total allowed resources were 12, and the percentage of FPGA system was 16.6%. The results indicated that in the image compression module, FPGA had a low utilization rate of various resources and a large resource margin to meet the further expansion needs of the system. Meanwhile, it also indicated that the current system was relatively reasonable in resource utilization, and there was no excessive use of resources.

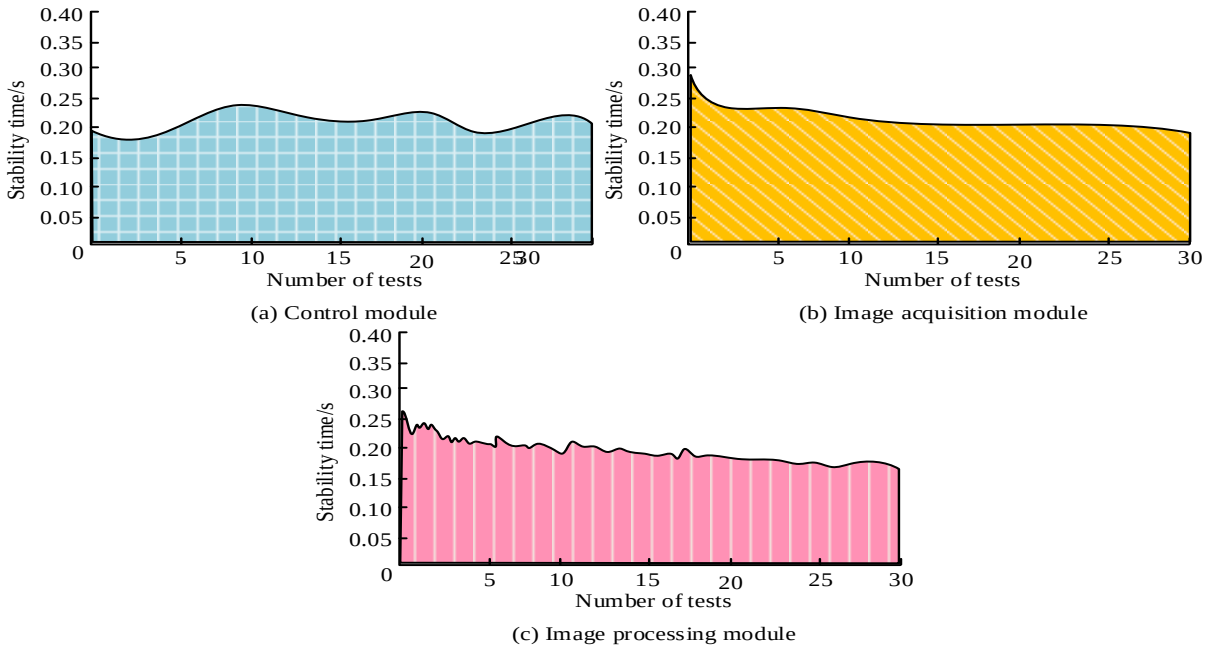

Fig. 11. Stability test results of each module.

TABLE II. RESOURCE UTILIZATION OF FPGA

Index	Resource type				
	LUT	BRAM	DSP	I/O	HMC
Usage amount	27582	38	69	57	2
Total resource quantity	20380	455	840	500	12
Proportion	13.5%	8.4%	8.2%	11.4%	16.6%

V. DISCUSSION

According to the characteristics of multi-spectral images, the JPEG-LS algorithm was optimized and the corresponding processing system was designed. From the application point of view, the optimized JPEG-LS algorithm had significant application potential in the field of multi-spectral image compression. For example, in the field of remote sensing, the amount of multi-spectral image data was huge, and the high compression ratio of the optimization algorithm could effectively reduce the cost of data storage and transmission, so that a large number of image data collected by satellites and other equipment could be processed and transmitted more efficiently. In the field of medical imaging, multi-spectral images were used for disease diagnosis, and the optimization algorithm could improve the compression ratio on the premise of ensuring image quality, which helped to store and transmit medical images quickly and facilitate doctors to obtain accurate information in time for diagnosis [25].

The advantages of this research work are more prominent. In terms of algorithm optimization, by improving the traditional coding method, a probability distribution model was constructed to predict pixel values, and the prediction error was converted into a unilateral exponential distribution, which effectively solved the problem of poor adaptability of the original algorithm to image noise and gray value fluctuations, and improved the stability of the algorithm. Meanwhile, the sub-block compression strategy and the block compression algorithm based on dynamic image bit width were adopted to significantly improve the compression performance. In the aspect of system design, the modules of the multi-spectral image processing system had clear division of labor and work together, which could efficiently complete the tasks of image acquisition, processing, compression and transmission. Among them, the image compression module was based on FPGA high-performance platform, combined with reversible component transformation and optimization algorithm to achieve lossless image compression, and ensure the image quality.

VI. CONCLUSION

Aiming at the problems of error sensitivity and compression performance in the efficient transmission of multi-spectral images using the JPEG-LS algorithm, this study proposed to optimize the JPEG-LS algorithm. By adopting sub-block compression strategy, dynamic image bit width and other improvement measures, the goal of reducing error sensitivity and improving compression performance was achieved, and a multi-spectral image processing system was constructed. Experimental results showed that the optimized JPEG-LS algorithm performed well under different parameters. When the micro loss was 1 and the block size was 512×512, the compression ratio could reach 5.55. Compared with other algorithms, the average compression ratio of the optimized

algorithm was 5.81, which was higher than that of study [12] (5.56), study [13] (5.46), study [23] (5.76) and study [24] (5.74), and the average compression time was 0.35s, which was comparable to other algorithms. The average value of PSNR was 43.6, which was higher than other comparison algorithms, and the average value of SSIM was 0.97, which was equivalent or better than some algorithms. In terms of the performance of the multi-spectral image processing system, the stability test results of each module were good, the average stability time of the control module was 0.21s, the stability time of the image acquisition module was reduced to 0.18s after 30 tests, and the average stability time of the image processing module was 0.19s in the first 30 tests. In the image compression module, the utilization rate of FPGA to all kinds of resources was low, the utilization rate of LUT was 13.5%, BRAM was 8.4%, DSP was 8.2%, I/O was 11.4%, HMC was 16.6%, and there was a large resource margin. The research method effectively improved the compression performance of multi-spectral images, reduced the storage and transmission costs while ensuring the image quality, and provided a more efficient solution for the application of multi-spectral images in many fields. However, there are some shortcomings in this study. In the process of algorithm optimization, although the influence of various factors on the compression performance was considered, the compression effect of images in complex scenes still needs to be further improved, and the computational complexity of the algorithm increased to a certain extent. In terms of system design, there is room for improvement in the communication efficiency between modules. The future research work can further optimize the algorithm, reduce the computational complexity, and improve the image compression effect in complex scenes. Then, by improving the communication mechanism between system modules, the overall operation efficiency is improved. The application of optimization algorithms and systems can be explored in more fields, such as intelligent security, industrial testing, etc., to expand its application range.

REFERENCES

[1] Gertsiy O. Research on graphic data formats for compact representation and comparison of images Transport systems and technologies, 2024 (43): 173-187.

[2] Turcza P, Duplaga M. Low-power low-area near-lossless image compressor for wireless capsule endoscopy Circuits, Systems, and Signal Processing, 2023, 42(2): 683-704.

[3] Li X, Wang K, Gu X, Deng F, Wang F Y. Paralleleye pipeline: An effective method to synthesize images for improving the visual intelligence of intelligent vehicles IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2023, 53(9): 5545-5556.

[4] Yuan Z, Zeng J, Wei Z, Jin L, Zhao S, Liu X, Zhou G. CLAHE-based low-light image enhancement for robust object detection in overhead power transmission system IEEE Transactions on Power Delivery, 2023, 38(3): 2240-2243.

[5] Zhou J, Pang L, Zhang D, Zhang W. Underwater image enhancement method via multi-interval subhistogram perspective equalization IEEE Journal of Oceanic Engineering, 2023, 48(2): 474-488.

[6] Zhang S, Wang Q, Zhou W, Yan A, Zhang J, Shi J, Li Z. Spatial pilot-aided fast-adapted framework for stable image transmission over long multi-mode fiber Optics Express, 2023, 31(23): 37968-37979.

[7] Wu J, Wu C, Lin Y, Yoshinaga T, Zhong L, Chen X, Ji Y. Semantic segmentation-based semantic communication system for image transmission Digital Communications and Networks, 2024, 10(3): 519-527.

[8] Khandelwal J, Sharma V K. W-VDSR: Wavelet-based secure image transmission using machine learning VDSR neural network Multimedia Tools and Applications, 2023, 82(27): 42147-42172.

[9] Gupta M, Singh V P, Gupta K K, Shukla P K. An efficient image encryption technique based on two-level security for internet of things Multimedia Tools and Applications, 2023, 82(4): 5091-5111.

[10] Al-Kadhimi A M, Abdelkareem A E, Tsimenidis C C. P-LDPC coded image transmission with OFDM over underwater acoustic channel Acta Polytechnica, 2024, 64(2): 68-76.

[11] Sun X, Chen Z, Wang L, He C. A lossless image compression and encryption algorithm combining JPEG-LS, neural network and hyperchaotic system Nonlinear Dynamics, 2023, 111(16): 15445-15475.

[12] Hua J, Xu H, Du Y, Du L. Improved JPEG Lossless Compression for Compression of Intermediate Layers in Neural Networks Based on Compute-In-Memory Electronics, 2024, 13(19): 3872-3882.

[13] Rahman M A, Hamada M. A prediction-based lossless image compression procedure using dimension reduction and Huffman coding Multimedia Tools and Applications, 2023, 82(3): 4081-4105.

[14] Al Qerom M, Otair M, Meziane F, AbdulRahman S, Alzubi M. LICA-CS: Efficient Lossless Image Compression Algorithm via Column Subtraction Model Journal of Robotics and Control (JRC), 2024, 5(5): 1311-1321.

[15] Hamano G, Imaizumi S, Kiya H. Effects of jpeg compression on vision transformer image classification for encryption-then-compression images Sensors, 2023, 23(7): 3400-3418.

[16] Mahajan H B, Junnarkar A A. Smart healthcare system using integrated and lightweight ECC with private blockchain for multimedia medical data processing Multimedia Tools and Applications, 2023, 82(28): 44335-44358.

[17] Zhou S, Qiu Y, Wang X, Zhang Y. Novel image cryptosystem based on new 2D hyperchaotic map and dynamical chaotic S-box Nonlinear Dynamics, 2023, 111(10): 9571-9589.

[18] Gümüş S, Kamisli F. A learned pixel-by-pixel lossless image compression method with 59K parameters and parallel decoding Multimedia Tools and Applications, 2024, 83(8): 22975-22993.

[19] Ungureanu V I, Negirla P, Korodi A. Image-Compression Techniques: Classical and "Region-of-Interest-Based" Approaches Presented in Recent Papers Sensors, 2024, 24(3): 791-817.

[20] Ahmad I, Choi W, Shin S. Comprehensive Analysis of Compressible Perceptual Encryption Methods—Compression and Encryption Perspectives Sensors, 2023, 23(8): 4057-4100.

[21] Joseph S M, Sathidevi P S. Microarray Image Lossless Compression Using General Entropy Coders and Image Compression Standards Circuits, Systems, and Signal Processing, 2023, 42(8): 5013-5040.

[22] Choudhuri S, Adeniye S, Sen A. Distribution Alignment Using Complement Entropy Objective and Adaptive Consensus-Based Label Refinement for Partial Domain Adaptation. Artificial Intelligence and Applications. 2023, 1(1): 43-51.

[23] Wang Y, Liang F, Wang S, Chen H, Cao Q, Fu H, Chen Z. Towards an Efficient Remote Sensing Image Compression Network with Visual State Space Model. Remote Sensing, 2025, 17(3): 425-444.

[24] Gao L, Zhang Y, Jiao A, Zhang L. A Road Extraction Algorithm for the Guided Fusion of Spatial and Channel Features from Multi-Spectral Images. Applied Sciences, 2025, 15(4): 1684-1703.

[25] Liu F, Li G, Wang J. Advanced analytical methods for multi-spectral transmission imaging optimization: enhancing breast tissue heterogeneity detection and tumor screening with hybrid image processing and deep learning. Analytical Methods, 2025, 17(1): 104-123.