

# Real-Time Image Transmission Algorithm in WSN with Limited Bandwidth

<https://doi.org/10.3991/ijoe.v14i12.9494>

Hui Huang, Zhe Li<sup>(✉)</sup>  
Hubei Engineering University, Xiaogan, China  
lizhe\_hbeu@vip.163.com.com

**Abstract**—In this paper, a real-time image transmission algorithm in WSN with limited bandwidth networks is studied. Firstly, a simple and effective monitoring network architecture is established, which allows multiple video monitoring nodes to access the network, and the data transmission is controlled by the synchronization mechanism without collision. Then, the image data is compressed locally at the monitoring nodes (over 85%), so that the image of each node can meet the needs of real-time data transmission, and the overall power consumption of the system is greatly reduced. Finally, based on NVIDIA TX1, four test nodes are constructed to test the algorithm cumulatively, which verifies the effectiveness of the system framework and compression algorithm.

**Keywords**—limited bandwidth networks, real-time image transmission, compression algorithm, synchronous network architecture, NVIDIA TX1

## 1 Introduction

### 1.1 Background

With the decrease of unit cost and the great improvement of performance, front-end embedded high performance computing systems such as NVIDIA TX1 get more and more attention in the Internet of Things and artificial intelligence. The improvement of the front-end processing capability can change the traditional application mode and method of existing wireless sensor networks (WSN) to a considerable extent <sup>[1]</sup>. In general, the sensors in WSN are mainly composed of sensors with high structure and less data, such as temperature, humidity, illumination, current and voltage, which can meet the needs of general applications [2]. Image sensors, especially high definition and multi-data sources (including infrared image sensor and depth sensor Depth Sensor) can provide richer and multi-level information, which can applied to complex application scenarios. However, a large amount of data generated by the network will cause great pressure to the power consumption of the limited wireless network bandwidth and the nodes (Recent work has investigated the application of image coding in WSNs for high-bandwidth networks [3]). In order to reduce the transmission pressure on the wireless communication link and reduce the back-end data processing and application pressure, it is an inevitable choice to use in-situ data

compression or processing, which results in the emerging term of "edge calculation", although it is more pointed to such as NVIDIA TX1 which has stronger computing ability and can meet the needs of a certain complex computing task and better support the development of future systems such as Internet of Things and artificial intelligence. Due to the high complexity and specificity of in-situ data processing, our research focuses on the system application in complex environment, which can realize real-time image transmission under the condition of limited bandwidth and take into account the requirement of low power consumption so that it can be long-term deployed outdoors [4]. Therefore, it is necessary to pay more attention to the image compression algorithm on the front-end node to effectively reduce the network load pressure.



**Fig. 1.** SCADA node formed by NVIDIA TX1 board and CCD camera(supervisory control and data acquisition, SCADA, data acquisition and monitoring system (DAMS))

## 1.2 Research status

The application of wireless image monitoring system is developing rapidly due to its flexible deployment, but its development is restricted by its high demand for power and wireless communication bandwidth [5]. Therefore, more requirements in existing WSNs are based on embedded image sensor systems such as CMUcam1 and CMUcam2 described in document [6]. Such systems typically consist of CMOS image sensors such as OV7660, embedded systems such as STM32 or DPS, and use serial ports using 115200 bps for data communication. When using  $160 * 225$  resolution, CMUcam2 can transmit up to 50FPS. The Cyclops [4] system uses ATmega128 MCU and is equipped with an image sensor with an Agilent resolution of  $352 * 288$  pixels. The main problem with such systems is that they have a small RAM, which is difficult to store multiple full-size images, and have processors of limited computing power. In the initial study, we also build a similar system, TinyBoard, which will be described later in the section.

### **1.3 Motivation**

The main problem that restrict the development of wireless sensor networks (WSN) is that it's difficult to balance limited bandwidth, node computing capability, wireless transmission power consumption and system cost. In order to meet that need for additional computing capabilities of complex systems such as video monitoring, a typical approach is to increase the performance of the node at the expense of cost and power consumption. Another approach is to use a distributed hierarchical network [7] to separate data acquisition and computation, where the front-end monitoring node is only responsible for image data acquisition, and then transmits data to the back-end service system through the wireless network to process the raw data as needed. However, the most important problem of this method is that the wireless network communication bandwidth is limited [8], and the power consumption is proportional to the distance and bandwidth, so that the cost of constructing such a system cannot be effectively controlled [9]. For instance, the institution of the cooperative researchers owns the 430M wireless mapping system, although it can transmit 1080p images in real time [10], its power consumption exceeds 10W and its price exceeds RMB50,000 nodes. The motivation of our research in this paper is to develop a powerful streaming media image data transmission algorithm, and based on which, a small wireless video monitoring network system is built to support long hop distances under limited bandwidth of low power consumption [11].

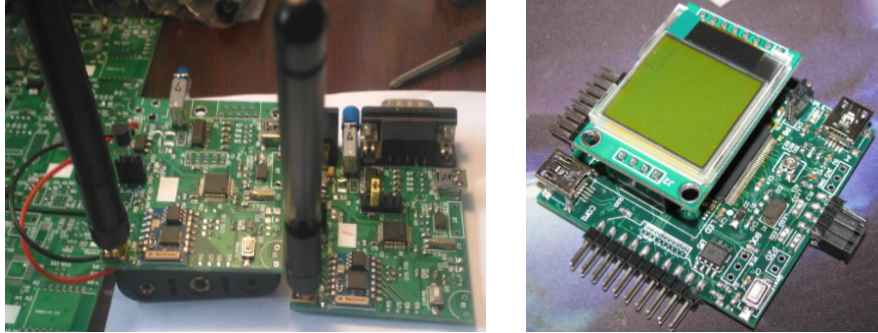
The main contributions of this paper are as follows: Constructing a synchronous wireless sensor network framework that allows monitoring nodes to be dynamically added or deleted; A front-end image compression algorithm and a corresponding restoration algorithm to meet the needs of real-time image transmission; A general simulation test and migration method, which can quickly deploy this research to a new platform, and perform off-line testing to verify the effectiveness of migration [12].

## **2 Hardware Platform**

### **2.1 TinyBoard**

TinyBoard (figure 2) is an outdoor wireless sensor network monitoring platform, which is mainly used in urban and agricultural environmental monitoring, and was developed by a research team in 2014 (for a literature). It is mainly composed of STM32C8T6 master controller, the default main frequency is 72MHz, and has 128KB Flash and 20KB SRAM; A standard ESP8266 WIFI wireless communication module can achieve a transmission rate of 1Mbps (700mW) in shorter distance; In order to support longer distance communication, a block of NRF905 module is connected through the SPI interface on board, and the default transmission rate is 50kbps (maximum 100kbps). The system consists of temperature, humidity and air pressure sensors to form the function of environmental monitoring. By default, a lithium battery is used for power supply [13]. In low power mode, a single node can operate for at least 2 months, and the whole system has been in operation in the park for 3 years. The

system is planned to be extended to address requirements such as environmental security monitoring through the addition of image sensors.



(a) TinyBoard I

(b) TinyBoard II has a LED display

**Fig. 2.** "TinyBoard" WSN node

Considering the difficulty and cycle of implementation, and the need for more complex applications such as human abnormal behavior analysis and immediate warning, this study first builds a simulation test system on TX1 platform which is more powerful and easy to test, and uses TinyBoard as a comparative subsystem to access the simulation platform for the test.

## 2.2 TX1 Board and Camera board

TX1 board is responsible not only for local image data processing, but also for sending encapsulated data packets to the server using wireless communication. TX1 platform includes a four-core ARM CPU A57 of 2GHz main frequency, equipped with 256 CUDA cores GPU, 4GB LPDDR memory, including UART, SPI, I2C, I2S, GPIO, USB 3.0 and other IO interfaces; in addition to deploy operating system and development environment, on-board 16GB eMMC memory can also store a certain amount of image data and provide 6-way CSI2 camera interface. IEEE 802.11 ac2x2 Wifi on board has the max transmission rate of 800Mbps. It has the computation capability of 1 Teraflops that is difficult for other front-end processing systems to achieve.

TX1 board carries a high-performance image sensor, with a resolution of 1280 \* 720 @ 30FPS, accessing to the motherboard through the CIS expansion board. We can already control it through the CAM\_I2C MUX interface of the motherboard, and can set its time of start and end to match the synchronous control algorithm. In order to simulate the actual application and contrast, we use a USB camera of 640 \* 480 @ 30FPS to carry out the follow-up experiment; A 128 GB SSD hard disk is also configured for each node to store all uncompressed raw and compressed images for subsequent compression performance analysis. The hardware structure of the system is shown in Figure 3:

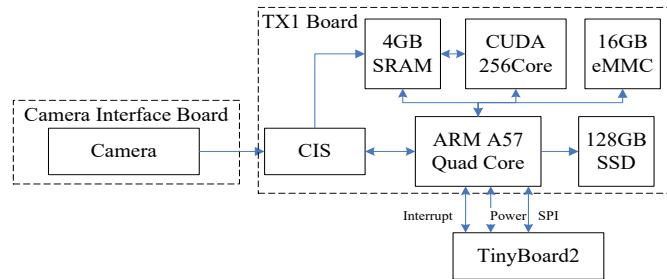


Fig. 3. Boxdiagram of system hardware structure

### 2.3 Controllable bandwidth simulation system

As mentioned in the introduction, TX1 has a powerful computing capability that is not available in traditional embedded systems or DSP systems, and the high bandwidth supporting 5G networks, which is contrary to the research theme of this article, "bandwidth constraints." The main consideration of selecting this system is that more development experience has been accumulated on the platform in the past two years, and the available programs limit the bandwidth to fully simulate the actual network environment and its changes, including bandwidth, multi-hop, time delay, data packet loss and so on. For this reason, a controllable bandwidth simulation system software framework is constructed with the framework shown in Figure 4,5:

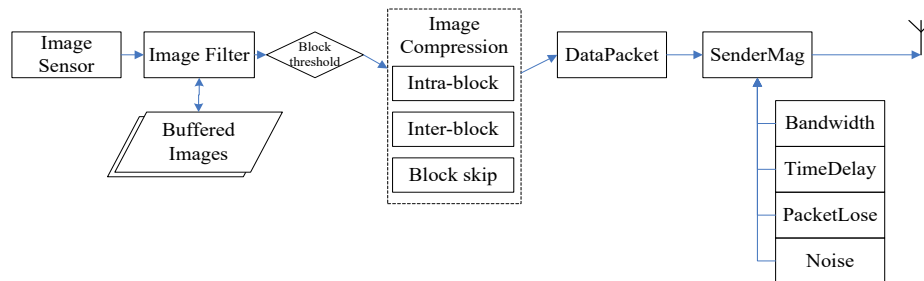


Fig. 4. Software Framework of Controllable bandwidth simulation system



Fig. 5. Referencing input and output image samples (Use the black blocks to cover sensitive information in the sample)

### 3 Overall Design of the System

#### 3.1 Overall framework

The overall architecture of the system is shown in Figure 6. The whole system is mainly composed of four parts: image sensor node (Client Node), wireless network pipe, data processing part and back-end database. The image sensor is supervised by Client Node in the form of star network, and its data is transferred to wireless gateway through bus (including CAN, SPI and USB). The image is compressed here, then transferred to the host at the receiving end to be restored and finally stored in the database. In addition, the wireless gateway node is responsible for manage the camera and ensuring that there are no conflicts in image transmission by operating the event trigger in a synchronized manner.

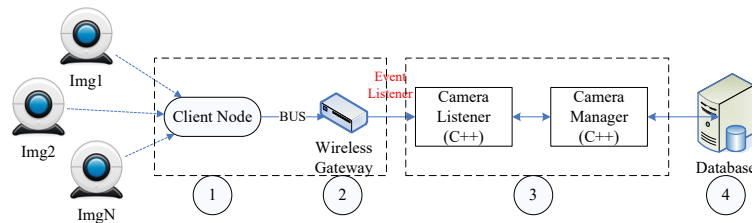


Fig. 6. Architecture of the system

#### 3.2 Synchronization controller of image sensor

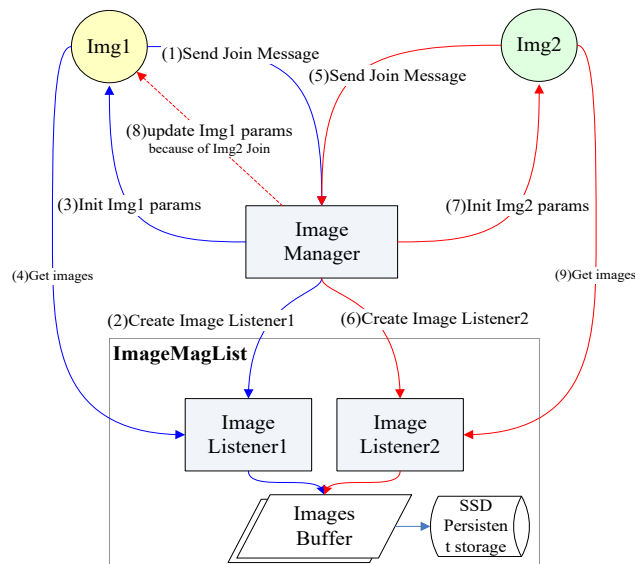


Fig. 7. Frame diagram of image sensor synchronization management flow

The synchronization controller of image sensor is one of the core of the whole system, and its synchronization management flow is shown in Figure 7.

Whenever a network joins a new image sensor, synchronization control information needs to be diffused in a broadcast mode to other nodes in the network. When these messages are returned to the Client Node, the controller will generate a new frame monitoring management object for the image sensor, and be responsible for its image data synchronization. In this part, a sensor management container ImageMagList is set up. By dynamically adding or deleting image sensors in the container management object, the current sensor and network operation state can be correctly reflected. With certain lag, the synchronization of the state is very important. Since the communication bandwidth of a single node is limited as the sensor increases, the bandwidth share of each image sensor will gradually decrease. In order to meet the need of real-time transmission of all sensors, the controller will dynamically adjust the proportion of the compression algorithm according to the number of sensors and the state information, and send the control information to the corresponding sensors.

In addition, it should be noted that in the outdoor battery-powered application scheme, it is unlikely to support long-term operation of full-scale real-time images at full speed. Generally, a large compression ratio/working ratio is adopted. From the point of view of power consumption, the nodes are actually in a state of lower power consumption, that's, the power consumption is proportional to the working ratio. Since the ratio is lower, the addition of more image sensors within a certain range does not cause a larger pressure to the system.

### 3.3 Software environment

At present, all the software runs on the TX1 mainboard. Its operating system is Ubuntu, software development environment uses CMake, image compression, restoration and transmission algorithms are written by C / C ++. The image processing library uses the OpenCV4Tegra version dedicated to the TX1 platform.

## 4 Algorithm of Dynamic Image Compression

Because the bandwidth of wireless communication network is limited (< 50/100kbps), the real-time transmission of images with original size cannot be satisfied. Therefore, image compression at the node is the focus of this paper. The image compression algorithm will consume additional amounts of computation and corresponding energy, but this can be offset to some extent by the reduced power consumption of wireless data transmission. Of course, the key to compression algorithms is to ensure real-time image transmission, and allow more sensors to join the network to cover a wider range.

#### 4.1 Image compression coding

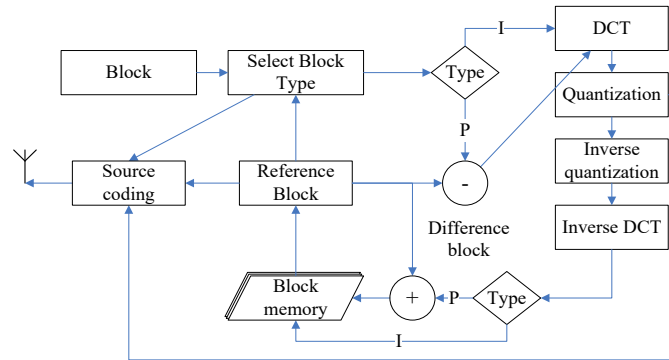


Fig. 8. Diagram of simplified block compression algorithm

Figure 8 shows the main steps of the node compression algorithm. An uncompressed RGB color image is first taken from the image sensor and sent to the SRAM for comparison with the saved image. In order to accelerate this process and make full use of TX1's advantages in parallel computation (such as CUDA), the image is decomposed into  $8 * 8$  image blocks for comparison. This compression process has two basic strategies in the part of the image sensor node, namely skipping the current frame or block and compressing the current frame or block, which can be further divided into three cases:

- 1) Skip the block: If the mean-square error (MSE) between a block in the current image and the corresponding block of the saved image is less than a certain threshold, the block image is considered to have no significant change, and thus not sent;
- 2) No reference code: Compressing the blocks in the current frame image when there is no reference; generally, it is used because the current image is the first frame, or the buffered reference frame is invalid;
- 3) Reference code: Typically, a difference algorithm is used for the current block is compressed based on the buffered reference frame.

For the non-skip block compression process, we refer to the JPEG compression algorithm. The block is converted into a frequency domain space using a 2D discrete cosine transform (2D DCT). Since only one or several values often occur in repeated blocks, we choose Run-length encoding (RLE) to perform intra-frame and inter-frame compression, and use Huffman encoding to dynamically construct the codewords with the shortest average length to reduce the amount of subsequent data transmission as much as possible.

According to the compression algorithm, for the input image shown in Figure 3, the block-based compression decomposition view is shown in Figure 9:



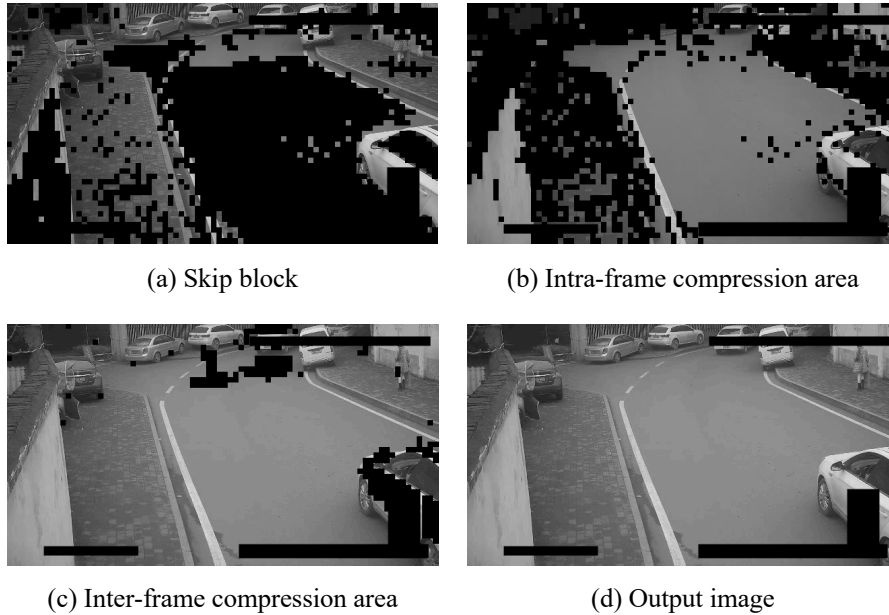


Fig. 9. Block based compression algorithm: Decomposition view

#### 4.2 Datagram structure

After the compressed image data is serialized, it needs to be sent into our customized datagram structure for packaging, as shown in Figure 10:

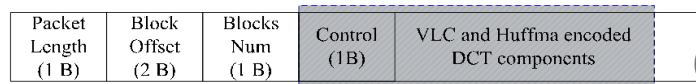


Fig. 10. A datagram structure for transmitting the serialized compressed image

The data packet is mainly composed of five parts, of which the control information part only occupies 5B, and the maximum length of a single packet can be up to 256, which has the advantage of using integer data blocks to facilitate the data parsing of the sending and receiving parties, and also is suitable for NRF905 with less data buffer (sending buffer area is 32B. This coding format refers to the literature [x]. The control block occupies 8 bits, where the first 2 bits are used to represent the block type (00 denotes skip, 01 denotes intra-frame compression, and 10 denotes inter-frame compression), and the second 6 bits are used for other control information and future extended applications. Such a data packet of 32B may contains several data blocks.

### 4.3 Decoding algorithm

The decoding algorithm framework is shown in Figure 11. We implement this algorithm in Client Node for testing based on C / C ++ and display the updated image on the screen. Due to the use of predictive coding, if a packet loss problem occurs in the data transmission, an error will occur in the current frame, which appears as an image miscode, such as the case we see in the wireless image transmission.

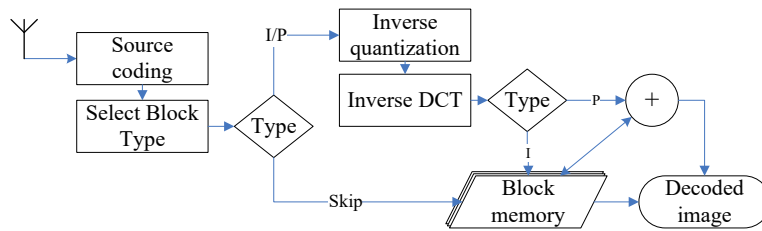


Fig. 11. Framework diagram of decoding algorithm

Moreover, the error will continue to affect subsequent frames, eventually causing the transmitted image to become unusable. An internal block is used to solve this problem. Whether to provide decoder feedback information will result in two internal block insertion algorithms. If the feedback information cannot be provided, a block is periodically inserted through a set timer, and a maximum intra-block interval is given; if feedback information can be provided, the information of that lost block is sent to the encoder by the decode, and then the intra-frame compression mode is selected to pass out of the block again, but in certain cases, this method results in rapid deterioration of the communication quality.

## 5 Test and Result Analysis

### 5.1 Data packet transmission test

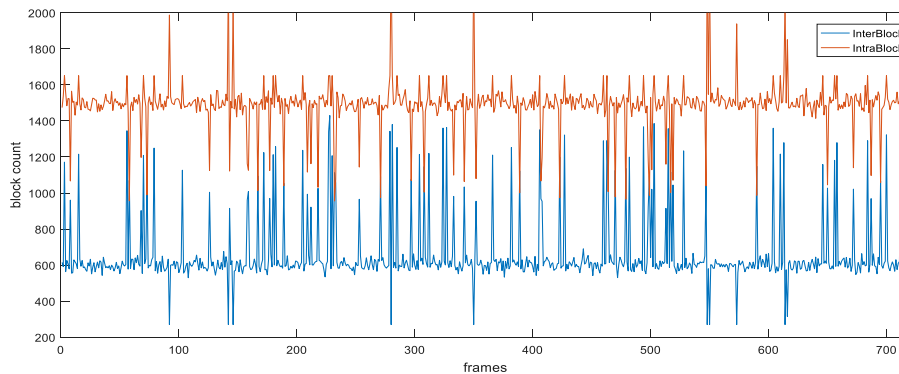
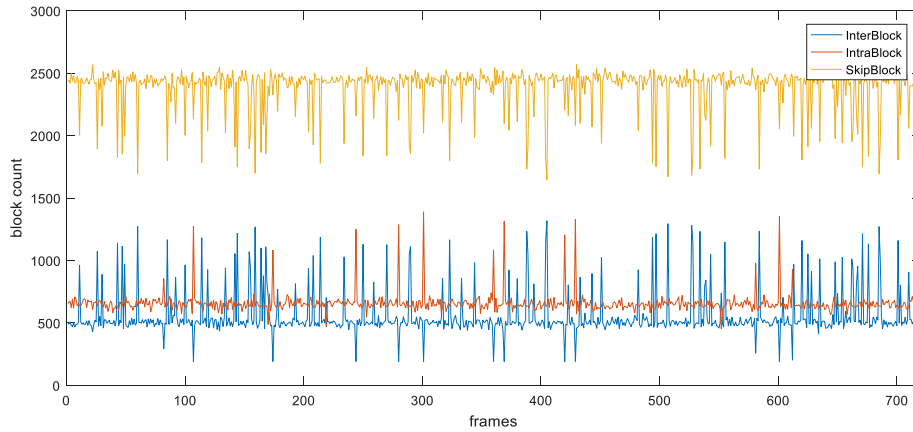


Fig. 12. data packet transmission contrast map in long image sequence

Figure 12 shows the number of packets actually transmitted by each image in a long sequence of images, the environment in the sample is essentially static, while foreground objects such as people occasionally appear.

The two cases are emphatically compared, where the upper curve is the number of intra-coded blocks and the lower curve is the number of inter-coded blocks. Wherein the average packet number is 1,490 packets/s or 42.3kbps. For 3 bytes per pixel of a standard RGB format image, the compression ratio at this time is about 20.7%. The intra-frame compression method has strong robustness, in which the lost data block / data packet area will be refreshed in the next period, and from the display result, occasionally some black blocks or random color blocks (actually the corresponding block of the previous frame) will appear on the screen.

When using inter-frame differential coding, a larger compression ratio can be obtained, the average transmission is 659 packets/s, 18.7 kbps, and the compression ratio is about 9.15%, as shown in Figure 13:



**Fig. 13.** Comparison of the number of three different blocks when inter-frame differential compression allowed

Then two nodes are used to compare the above two kinds of cases. The transmission rate of node 1 is about 605 packets/s, which can reach 1FPS. The average number of the three data blocks is 66.5% (block jump), 18.2% (intra-frame compression) and 15.3% (inter-frame compression). Figure 14 displays three images of a fixed camera node in a day at an interval of 10 minutes.



**Fig. 14.** Comparison of three images of the fixed cameras at 10-minute interval

This particular sequence of images clearly shows the process of plastic waste and other objects ,as well as the change in light (shadow).

## 5.2 Power test

Compressing an image in situ at a node can effectively reduce the amount of data transmission, and can reduce the power consumption during data transmission. But the image compression algorithm will increase the amount of computation, thus increasing the power consumption of the system. In order to evaluate the relationship between the compression algorithm and the power consumption of data transmission, we compare the image transmission system with the compression algorithm with the one without the compression algorithm. The test results are shown in Tables 1 and 2.

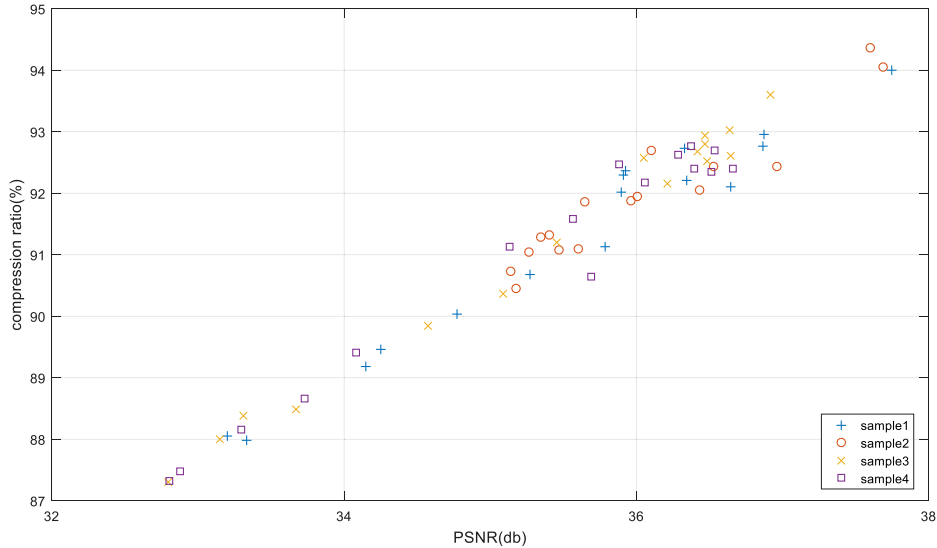
**Table 1.** Table I System power comparison when switching on/off compression algorithms

	Camera	CPU	Wireless Trans	AVG Power	Energy Cost
Compression On	25mA	330 mA	285 mA	2112mW	836mJ
Compression Off	25mA	240 mA	390mA	2161.5mW	4280mJ

According to the test results, when the image compression algorithm is turned on, the average power consumption of the system is 836mJ; when it is not turned on, the average power consumption is 4,280mJ. That’s, the average power consumption differs by 5 times. Obviously, image compression in situ at nodes will reduce the power consumption of wireless data transmission to a large extent, which is enough to offset the extra power consumption generated by adding new computation, fully demonstrating the validity of the compression algorithm in this paper.

## 5.3 Performance test of image compression algorithm

To further evaluate the actual performance of the compression algorithm, we deploy one of the nodes outdoors in the scenario shown in Figure 8. Because the scene contains a more complex background environment, people, and vehicles, with limited lighting conditions at night and excessive noise, the test images only under sufficient daylight during the day are selected. For each compressed frame, we calculate the total amount of compression and the peak Signal to Noise Ratio (PNSR) between the original image (stored in a node (SSD) hard disk) and the final decoded image (stored in a server), and the test result is shown in Figure 15:



**Fig. 15.** PSNR comparison data for test nodes in four days

The results shown in Figure 15 indicate that there is a significant linear relationship between PSNR and compression algorithm. When the image taken by the camera changes strongly, the difference between the current frame and the reference frame increases, thereby increasing the total transmission amount of the data block, reducing the compression ratio and the PSNR value. Under the condition of limited bandwidth, the average compression ratio of the proposed compression algorithm is 93.2% (87.6% ~ 94.5%), and the average PSNR is 34.2db (33 ~ 37.8db) with good performance.

## 6 Summary

In this study, a wireless image transmission algorithm with wide adaptive range and limited bandwidth is proposed, and can be applied to upgrade existing WSN. The algorithm and management framework allow the network to dynamically add or delete image sensor nodes. By performing data compression locally on the nodes, the total amount of data transmission can be reduced to a considerable extent, and the power consumption of the system resulting from wireless data transmission can be greatly reduced. The future research focuses on the integration of the existing research prototype system, making it suitable for all kinds of platforms; in the network management architecture, it supports the function of routing and transferring, and realizes the image data transmission of longer distance.

## 7 Acknowledgement

Public Welfare Scientific Research Program is funded by the Hubei Provincial Department of Education (B2017504, B2017506); Scientific research project is funded by Wenzhou City (Grant No. LY G20160020).

## 8 References

- [1] Krishnamachari, B. (2005). *Networking Wireless Sensors*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511541025>
- [2] Sikka, P., Corke, P., Overs, L. (2004). Wireless sensor devices for animal tracking and control. *IEEE International Conference on Local Computer Networks*, 446-454. <https://doi.org/10.1109/LCN.2004.141>
- [3] Lajara, J.R., Pelegrisebastia, J., Solano, J.J. (2010). Power Consumption Analysis of Operating Systems for Wireless Sensor Networks. *Sensors*, 10(6): 5809-5826. <https://doi.org/10.3390/s100605809>
- [4] Akyildiz, I.F., Melodia, T., Chowdhury, K.R. (2007). A survey on wireless multimedia sensor networks. *Computer Networks*, 51(4): 921-960. <https://doi.org/10.1016/j.comnet.2006.10.002>
- [5] Li, H.J., Xu, G.N., Zhou, L.D. (2017). Mechanical parameters acquisition test based on wireless sensor network, *Academic Journal of Manufacturing Engineering*, 15(4), 88-93.
- [6] Wark, T., Corke, P., Sikka, P., Klingbeil, L., Guo, Y., Crossman, C., Valencia, P., Swain, D., Bishop, H.G. (2007). Transforming agriculture through pervasive wireless sensor networks. *IEEE Pervasive Computing*, 6(2): 50-57. <https://doi.org/10.1109/MPRV.2007.47>
- [7] Rowe, A., Rosenberg, C., Nourbakhsh, I. (2002). A low cost embedded color vision system. *IEEE/RSJ International Conference on Intelligent Robots & Systems*, 1: 208-213. <https://doi.org/10.1109/IRDS.2002.1041390>
- [8] Rahimi, M., Baer, R., Iroezzi, O.I., Garcia, J.C., Warrior, J., Estrin, D., Srivastava, M. (2005). Cyclops: In situ image sensing and interpretation in wireless sensor networks. *International Conference on Embedded Networked Sensor Systems*, 245(1): 192-204. <https://doi.org/10.1145/1098918.1098939>
- [9] Hu, W., Tran, V.N., Bulusu, N., Chou, C.T., Jha, S., Taylor, A. (2005). The design and evaluation of a hybrid sensor network for cane-toad monitoring. *International Symposium on Information Processing in Sensor Networks*, 5(1): 503-508. <https://doi.org/10.1145/1464420.1464424>
- [10] Polastre, J., Szewczyk, R., Culler, D. (2005). Telos: Enabling ultra-low power wireless research. *Ipsn Fourth International Symposium on Information Processing in Sensor Networks*, 364-369. <https://doi.org/10.1109/IPSIN.2005.1440950>
- [11] Corke, P., Valencia, P., Sikka, P., Wark, T., Overs, L. (2007). Long-duration solar-powered wireless sensor networks. *Workshop on Embedded Networked Sensors*, 33-37. <https://doi.org/10.1145/1278972.1278980>
- [12] Liu, C. (2017). Design and Implementation of Real-time Brain-computer Interface System Based on LabVIEW. *NeuroQuantology*, 16(3): 97-104.
- [13] Banerjee, S., Ghosh, A., Mitra, S.K. (2017). A modified mathematical model for life-time enhancement in wireless sensor network. *Mathematical Modelling of Engineering Problems*, 4(2): 84-90. <https://doi.org/10.18280/mmep.040204>

## 9 Authors

**Hui Huang** is an Associate Professor of College of Technology, Hubei Engineering University, Xiaogan 432000, in China. His research interests include image process, embedded system, computer science and big data and data mining.

**Zhe Li** received the Ph.D. degree. He is currently a Lecturer with the School of Economics and Management, Hubei Engineering University, Xiaogan 432000, China. His research interests include the consistency control of multiple agents in complex environments, multi-rates with multi-agent systems, and distributed control systems.

Article submitted 05 September 2018. Resubmitted 14 October 2018. Final acceptance 16 October 2018. Final version published as submitted by the authors.