A Collaborative Coding Framework of RS and H.264

Jiejun Hu and Xilong Che

Department of Computer Science and Technology, Jilin University, China

ABSTRACT

The quality of online multimedia is required higher and higher, yet the uncertainty of network, such as interference, connection failure, still exists. In order to ensure the quality of service in multimedia transmission, we incorporate erasure code with video code together, and present a collaborative coding framework of Reed-Solomon code (RS) and H.264. First of all, we analyze the common coding pattern hidden in RS and H.264. Then, we design the collaborative coding framework of the two codes. Finally, we give a prototype implementation and validate the coding effectiveness using standard dataset. The erasure data slice goes through the redundant data tunnel of H.264, thus it will not involve extra traffic overhead. Experimental results show that the framework can increase the quality of media transmission.

KEYWORDS

RS coding, H.264 coding, erasure code, video code, collaborative coding framework

1. INTRODUCTION

With the development of the Internet, multimedia has become the common traffic in most networks. However some of the networks are lack of sufficient guarantee of quality-of-service (QoS). Researchers start to focus on new technologies of protecting multimedia data. Comparing to the protection of traditional data, Multimedia data protection has more particular features.

First of all, users can bear the latency of opening a file or picture, but they can't endure the pause while playing a movie. It is hard to balance the data reliability and real-time performance of a large volume of multimedia data. Furthermore, the network environment is dynamic and various. Package loss in multimedia transmission happens frequently, since TCP/IP is a protocol which only makes best efforts.

Video coding technology has been rapidly developed. This paper focuses on the standard video coding framework -- H.264/AVC. H.264 provides many new ways to ensure the quality of the video, because of the best-effort nature of the network. In H.264, it also introduces a concept called redundant slice. Redundant slice is an empty channel for some redundant information of video coding. Mostly the attempt of protecting video data focuses on transmit layer. After the video data is packaged, H.264 uses erasure codes to encode the data packets and produce the redundant data packets, afterwards transmit them into the network.

In this paper, we propose a collaborative coding framework of RS codes and H.264 standard. Different from the previous works which generate the redundant data after video encoding, this paper focuses on the deeper combination of RS and H.264 standard which generates the encoded multimedia data and redundant data as the same time. The contribution of this paper is threefold:

(1) we analyze the features of RS codes and H.264 standard to find the common details of both coding technologies;

(2) we apply the common coding details in RS codes and H.264 to realize the collaborative framework. The collaborative framework can generate both the encoded multimedia data and redundant data when video encoding procedure ends;

(3) we implement the simulation environment to test the framework.

2. COLLABORATIVE FRAMEWORK FOR H.264/AVC AND ERASURE CODING

2.1. Mechanism Analysis of RS Coding

In distributed file system, such as Hadoop, erasure code is used in the file system to make sure of the data reliability. The principle of erasure code is firstly coding the blocked data, secondly producing redundant data block. When a client requires data from a distributed system, the system first searches the data. If the data is integrated then the client gets the data. If the system fails to search the integrated data, the system will collect the residual data and the coded redundant data, decode them to recover the original data. Erasure code is a forward error erasure code (FEC code) for binary erasure channel. There are many kinds of erasure codes, for example, Tornado codes, LT codes, Raptor codes and so on. Reed-Solomon code is the most widely used and mature method in erasure codes.
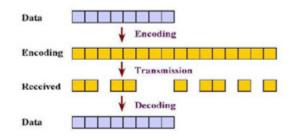

Figure 1. The brief coding procedure of RS codes

In this paper, we introduce Reed-Solomon code into the framework. First of all, we have to declare the principle and procedure of coding. We suppose that there are n devices, let $D_1, D_2 \ldots D_n$ be the Data Devices(DD), each of them stores k bytes data. Let $C_1, C_2 \ldots Cm$ be the Redundant Devices (RD), each of them stores k bytes data. The algorithm encodes the DD to get RD.

The algorithm only proceeds on the word level. So DD must divide into words, and each word contains w bits. $d_1, d_2 \ldots dn$ is presented the words in one device, so as $c_1, c_2 \ldots c_m$. We define the encode function B:

$$c_i = B_i(d_1, d_2 \ldots dn)$$

Vector D and C represent data words and redundant words. B_i is used as a row to compose matrix B, so matrix B will satisfy the equation F·D=C. As RS coding is based on Galois Field, we define B as Galois Matrix. Vector D and vector C are composed of a matrix E.
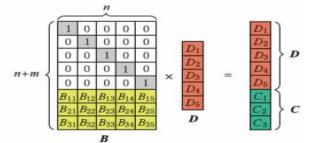

Figure 2. The multiplication of Galois matrix

When one of the devices malfunctions, then we delete the corresponding rows of B and D. As a result of the deleting, it will become a new matrix, and satisfy the equation: $B' \cdot D' = E'$. When m devices breakdown, there will be a $n * n$ matrix, at the same time, any rows in the matrix is linearly independent. Because of the property of the matrix, the value of D is calculated by Gaussian elimination of $B' \cdot D = E'$. If the data is totally recovered, the redundant data will be calculated again.
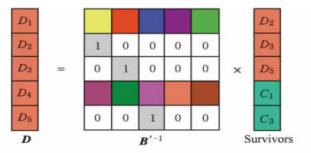

Figure 3. The recovery of the loss data

2.2. Mechanism analysis of H.264 coding

In this section, we will discuss the coding details in H.264/AVC and entropy coding.
H.264/AVC provides two parts of the framework, including video coding layer (VCL) and network abstraction layer (NAL). VCL is used for the effective coding of video. At the same time it defines the video data. VCL uses hybrid coding scheme and describes the picture by macroblock. The basic idea of H.264 is motion estimation and time-frequency transform in macroblock. It depends on time and space statistics for residual predicted coding. H.264/AVC contains several important parts, including motion compensation, residual transform coding based on DCT, quantification, zig-zag scan and entropy coding. NAL is used for transmitting the coded data packets into the network.

H.264 does the coding of 4×4 residual block in this framework. It provides low computing complexity and high quality of video data by integering transformed the residual block. After transformation and quantification, the video data has some features:

(1) 4×4 residual block is a sparse matrix;

(2) After zig-zag scanning, the coefficients of DC are non-zero, and most of the coefficients are +1 or -1;

(3) The coefficients of the nearby 4×4 residual blocks are relative;

As the features above, we do the entropy coding after the quantification. There are two methods of entropy coding: CAVLC (Context Adaptive Variable Length Coding) and CABAC (Context Adaptive Binary Arithmetic Coding). CAVLC is less complicated and more efficient than CABAC. Thus we use CAVLC to coding the 4×4 residual blocks in this paper.

Before entropy coding, the 4×4 residual blocks should be re-ordered by the zig-zag scan, they are restored into a one-dimensional array, as it shows in Fig. 4. An example for a typical zig-zag scan of quantized transform coefficients could be given as follows, all the coefficients in the matrix are restored in VC_array={0,3,0,1,-1,-1,0,1,0,0,…}, as it shows in Fig. 5.
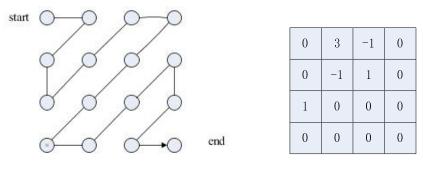

Figure 4. Zig-zag scan   Figure 5. A residual block

The simpler entropy coding method uses a single infinite-extent codeword table for all syntax elements except the quantized transform coefficients. Thus, instead of designing a different VLC table for each syntax element, only the mapping to the single codeword table is customized according to the data statistics. The single codeword table chosen is an exp-Golomb code with very simple and regular decoding properties.

In this scheme, VLC tables for various syntax elements are switched depending on already transmitted syntax elements. Since the VLC tables are designed to match the corresponding conditioned statistics, the entropy coding performance is improved in comparison to schemes using a single VLC table.

2.3. Collaborative coding framework

In this part we propose the architecture of the collaborative framework of RS codes and H.264 standard. The framework is composed by two main parts: a coding part, which collaborates RS codes and H.264 standard; a transmit part, which transmits redundant data with the coded video data.
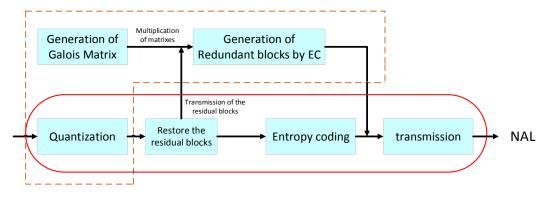
Figure 6. The collaborative framework of RS codes and H.264 standard

In this part we propose the architecture of the collaborative framework of RS codes and H.264 standard. The framework is composed by two main parts: a coding part, which collaborates RS codes and H.264 standard; a transmit part, which transmits redundant data with the coded video data.

For the coding part, the collaborative coding framework begins after the quantization of the 4×4 residual blocks. We find that entropy coding in H.264 has many things in common with RS coding by analyzing H.264 standard. For example, they both apply algorithms on a matrix. So the 4×4 residual blocks are suitable for the collaborative coding framework. In the erasure coding part of the whole framework, the first thing to do is to decide the degree of protection of the residual blocks. For example, if the data is important such as an online video meeting, then we have to increase the degree of protection of the data, so as to increase the redundant data. Secondly, according to the degree of protection, RS codes generates Galois matrix. Third, RS codes does the multiplication of Galois matrix and the residual blocks, so we get the redundant data of the video.

For the transmission part, the redundant data is packed into the "similar-NALU". When the coded video data is ready to transmit, then they will transmit with redundant data of the video together. Because the redundant data is different from the original data, we proposed a new structure to transmit the redundant data into the network. We take NALU (the structure for transmitting the packaged video data) of the video data as a reference. As a result, we propose "similar-NALU" to transmit the redundant data. In addition we use redundant slice in H.264 to transmit the "similar-NALU", so it won't increase the traffic of the network.

As it shows in Fig. 5, this matrix is the 4×4 residual block after quantify. We call it residual_block. It is stored in a one-dimension array in zig-zag order, the one-dimension array called VC_array.

We use residual_block to multiply the Galois_matrix. According to the feature of the multiply the matrix, we know that the multiply is between the column vector of the Galois_matrix and the row vector of the residual_block. So when the multiply happens, the resisual_block must be restored in a new one-dimension array in the row order, so that the procedure will end normally.
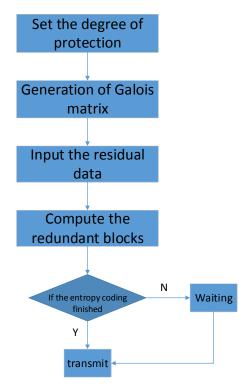
Figure 7. The procedure of RS coding

As it shows in Fig. 5, this matrix is the 4×4 residual block after quantify. We call it residual_block. It is stored in a one-dimension array in zig-zag order, the one-dimension array called VC_array.

We use residual_block to multiply the Galois_matrix. According to the feature of the multiply the matrix, we know that the multiply is between the column vector of the Galois_matrix and the row vector of the residual_block. So when the multiply happens, the resisual_block must be restored in a new one-dimension array in the row order, so that the procedure will end normally.

In order that our framework will be realized, first we do a little trick of the residual blocks. We divide VC_array into 4 groups, every group has 4 elements. Second we put every group into a new matrix, we call that "new_residual_block", in the row order. As the features of the residual_matrix, the new_residual_block has a new character: the matrix is sparse, even the last few rows are zero. By this way, we make the RS codes and H.264 standard collaborate together. At the same time, we simplified the calculation of the matrix multiplication.

We now give an example of the collaborative coding framework. The residual matrix shows in Fig. 5. First, it is scanned by the zig-zag order, so we get a sequence like: 0, 3, 0, 1, -1, -1, 0, 1, 0, 0,…. Second we store the elements into the VC_array = {0,3,0,1,-1,-1,0,1,0,0,…}. As what we talked above, the new_residual_matrix shows in the Fig. 8. Then the Galois_matrix and new_residual_block do the multiplication .
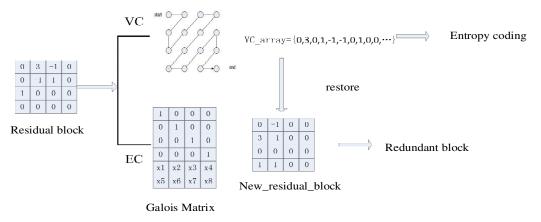
Figure 8. The example of the collaboration coding framework

Although the character of the new_residual_block increases the efficiency of the coding, it can't recover the real residual block. So we introduce the procedure called "inverse zig-zag scan" to recover the original video data, as it shows in Fig. 9.
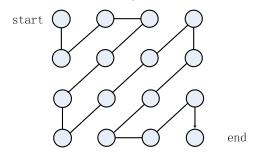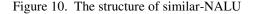

Figure 9. Inverse zig-zag scan

After the collaborative coding framework, all the data blocks are ready to transmit into the network. Here the problem comes, H.264 has already had a structure to pack the video data block, it uses a structure called NALU to realize this procedure. What to do with the redundant data coded by erasure codes? We designed a new structure called "similar-NALU" to pack the redundant data. Similar-NALU is a twin of the original NALU of H.264.

```
            +---------------------------+

            |0 | 1| 2| 3| 4| 5| 6| 7|

            +-+-+-+-+-+-+-+-+-+-+-+-+-+

             |M| IM |   E_TYPE   |

            +-----------------------------+
```

Figure 10. The structure of similar-NALU

In similar-NALU, there are three fields:

(1) M(Mark): this is the flag of a similar-NALU, because the flag of NALU normally sets "0", so we use "1" to represent similar-NALU.

(2) IM(IMportant): IM represents the importance of the similar-NALU. Usually we set "00".

(3) E_TYPE(erasure_type): E_TYPE is the type of the erasure coded data, it has three kinds of data: Y data, U data and V data, according to the video data types.

There is a function called "redundant slice" in H.264. This function uses residual bandwidth of video data to transmit the redundant data. However, redundant slice is not really realized. So in our work, we use the residual bandwidth of the redundant slice to transmit the similar-NALU. In this way, on one hand we increase the reliability of the video data, on the other hand, the redundant data doesn't increase any extra bandwidth.

Now we come to the last part of our collaborative framework: the decode procedure. Firstly, we need to check if the NALUs of the video is integral. If so, the video data applies for the decode procedure of H.264. If not, the framework uses the redundant data to recover the original data. Using RS codes to recover the original data as follows:

(1) Generate the corresponding Galois_matrix for the redundant data;

(2) Use Berlekamp-Massey algorithm and Forney algorithm to estimate the location of the errors;

(3) Recover the error elements.

Once we get the original NALU, we use H.264 to decode the video data.

3. IMPLEMENTATION AND EVALUATION

3.1 Implementation

In order to implement the collaborative coding framework, we extend the JM(Joint Model, version 13.2) and apply for RS codes in to JM framework. The combination of two coding schemes is after quantization. The collaborative coding framework is realized in C. We also make use of the "foreman_part_qcif.yuv" data set, which is a standard testing data set. In H.264, we apply the baseline mode of coding to the video data.

We also implemented a simulator on NIST Net to evaluate the collaborative framework's performance in different network conditions. NIST Net is a tool to facilitate testing and experimentation with network code through emulation. The version of NIST Net is 2.0.12b. It works on the Linux OS, it acts like a black box. We can use NIST sNet to simulate many kinds of network conditions, such as: packet loss, delay, jitter, congestion and limited bandwidth. In our framework, we only realize the packet loss function.

The simulation of the network including 3 parts:

(1) A sender which runs the collaborative framework to encode;

(2) A router which is a virtual machine runs NIST Net, and the router must have eth0 and eth1 to communicate with the sender and the receiver;

(3) A receiver which also runs the collaborative framework to decode.
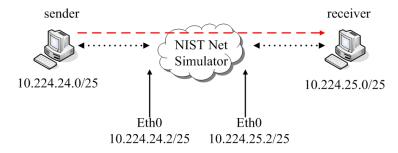
The topology is as the following picture shows:

Figure 11. The topology of the simulation

In addition, Gilbert-Eliott model(GE model) is introduced into NIST Net simulator. In GE model, the channel is classified into two states: "good " and "bad". We donate P_G as the possibility of good state and P_B as the possibility of bad state. The possibility of changing good state to bad state is P_{GB}, and P_{BG} is the opposite. Suppose the channel is in steady state, then:

The average loss possibility of the whole EB model:

$$P_{average} = P_G * P_{BG}/(P_{BG} + P_{GB}) + P_B * P_{GB}/(P_{BG} + P_{GB})$$
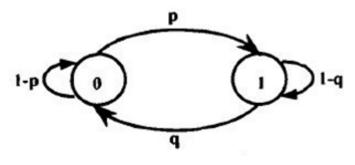
the state shows in the figure：

Figure 12. GE Model

By adjusting the parameters of the model, the NIST Net simulates a network condition with 5%, 10% and 15% packet loss.

3.2. Evaluation

This section describes the evaluation of the collaborative framework using NIST Net simulator. Generally, Signal-to-noise ratio (abbreviated SNR) is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. The definition of SNR as follows:

$$SNR = 10 * Log_{10} \frac{Signal\_power}{Noise\_power}$$

When the typical reference threshold value of SNR is 25-55db, the noise of the picture is low and the quality is good. When SNR is 60db, the picture is the best and without noise.

We focus this evaluation on three aspects.

(1) Without this collaborative framework, what will the video quality be when there is packet loss?

The first experiment we test is only the H.264 standard in different network condition. The purpose of the first experiment is to see how these parameters change during the different packet loss rate. We use 3 different packet loss rates to test. Figure 6.7 shows the SNR of Ydata, Udata and Vdata in video data. As the increasing of the packet loss rate, SNRs decrease.
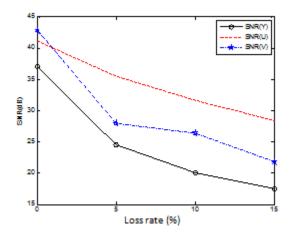

Figure 13. SNR under different data loss rates witåhout collaborative framework

After knowing the result of the first experiment, we want to find out if our collaborative framework help or not. Thus here comes the second experiment.

(2) Does the collaborative framework help when the packet loss rate increases?

The purpose of the second experiment is to show that the collaborative framework does make an effort in a poor network condition. In this experiment, the erasure codes set as RS(15,9), that means after the RS encoding there will be 15 data blocks, and there are 9 original data blocks, so the redundant data block is 6. At the same time, these 6 redundant data blocks use the "redundant slice" to transmit into the network. Figure 6.8 shows the effects of the collaborative framework. When the packet loss rate increases, the SNR is increased by the collaborative framework nearly 1db.
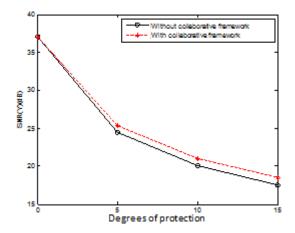
Figure 14. SNR under different data loss rates with collaborative framework

The result shows that the framework helps when the packet loss rate increases. This result turns out that the collaborative framework of RS codes and H.264 standard can do as a supplement to ensure the quality of the video. With the third experiment we will adjust the parameters of the framework to see the result.

(3) How will the quality of the video change when we modify the parameters in the collaborative framework?

In the third experiment, we adjust the parameters of the RS code in the collaborative framework with the same bad network conditions -- 5% packet loss. We apply RS(15.9), RS(15,7) and RS(15,5). We create more redundant data blocks in the encoding procedure. In figure 6.9, we find that with the increasing of the redundant data blocks, the SNR increases nearly 2db.
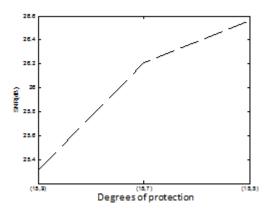

Figure 15. SNR under different degrees of protection with collaborative framework

4. RELATED WORKS

We have talked about all the details and the result of the simulation and experiments. There are two special aspects of the collaborative framework of RS codes and H.264 standard. On one hand, the whole collaborative framework makes two coding schemes together and achieves the

quality of the video. On the other hand "similar-NALU" makes it possible to avoid increasing any traffic while transmitting the redundant data. Related work falls into three aspects: using erasure coding in the transmitting stage, analyzing the importance of the video data and then using erasure codes and using different erasure codes in transmitting stage.

Xingjun Zhang et al. proposed a method called ULP—Unequal Loss Protection scheme for data transmission over lousy networks. This is developed from the ELP – Equal Loss Protection. By analyzing the scheme of H.264/AVC, they found out that the packet transmitting in network has unequal importance. They classify the data into three types – data partition A, B, C. In decoding, PA is the most important data partition of these three, and then PB, PC. And they apply Reed-Solomon code with different recoverability to the data partitions. The result shows that at the same loss rate, the video quality of ULP is much better than ELP.

R.Razavi et al. proposed a method that using growth codes to protect data partitions of H.264 video streaming. Using a growth codes was especially for the wireless networks. The method provided a hierarchy of protection, both for I frames and P frames. They used Raptor channel coding which allowed incremental protection of video data partitions in network. It also used UEP channel coding. By Raptor codes, this method of protecting data was up to 10db in PSNR(video quality parameter). Different from these two works above, our work pay attention to the common coding scheme between the RS codes and H.264 standard.

Yevs Dhondt et al. proposed a flexible macroblock scheme for unequal error protection. In H.264/AVC, Flexible Macroblock Ordering (FMO) is a new tool. This paper improved the original FMO. The importance of macroblock will be recalculated and re-packed in a slice. By this way, it has achieve an increase of PSNR of 4 dB on average.

Many researchers pay most attention on the phase, which the video has been coded by H.264 and ready to transmit to the channel. By using this way, they achieve the better quality of the video data. However the time efficiency of erasure coding is low. At the same time, the video data they used to do the erasure coding is in large granularity, it can't use the redundant slice to transmit, so it increases the redundant data in the channel. In our work, firstly it provides a framework that makes the erasure coding procedure "go-through" the video coding under H.264. Secondly we produce the smaller granularity redundant data to recover the lost data, the decoding procedure will be simple. Finally we use the redundant slice to transmit the redundant data, and pack the redundant data into the similar-NALU. These all makes our work different from the former ones.

5. CONCLUSION

This research proposed a collaborative coding framework of RS code and H.264 code. Our efforts include three main parts. First, we analyzed the coding mechanism of RS code and H.264 code, and found out that both coding schemes are based on using of media matrix, which stands as the basis for collaborative coding. Second referencing to the traditional NALU, we gave the twin NALU called "similar-NAUL" to transmit the redundant data with the original bandwidth. Finally we simulated the package loss codition in network environment and validated the effective net of collaborative coding framework proposed. Experimental results show that the combination of erasure coding and video coding did improve the value of SNR.

REFERENCE

[1] Karp R, Luby M, Shokrollahi A. Finite length analysis of LT codes[C]//Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on. IEEE, 2004: 39

[2] Xianghong L, Jiwu S. Summary of research for erasure code in storage system[J]. Journal of Computer Research and Development, 2012, 49(1): 1-11..

[3] Wang Y, Zhu Q F. Error control and concealment for video communication: A review[J]. Proceedings of the IEEE, 1998, 86(5): 974-997.

[4] Wiegand T, Sullivan G J, Bjontegaard G, et al. Overview of the H. 264/AVC video coding standard[J]. Circuits and Systems for Video Technology, IEEE Transactions on, 2003, 13(7): 560-576.

[5] Wang Y, Ostermann J, Zhang Y Q. Video processing and communications[M]. Upper Saddle River: Prentice Hall, 2002.

[6] Dhondt Y, Lambert P, Van de Walle R. A flexible macroblock scheme for unequal error protection[C]//Image Processing, 2006 IEEE International Conference on. IEEE, 2006: 829-832.

[7] Zhang X, Peng X, Fowler S, et al. Robust H. 264/AVC video transmission using data partitioning and unequal loss protection[C]//Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on. IEEE, 2010: 2471-2477.

[8] Razavi R, Fleury M, Altaf M, et al. H. 264 video streaming with data-partitioning and growth codes[C]//Image Processing (ICIP), 2009 16th IEEE International Conference on. IEEE, 2009: 909-912.

[9] Schierl T, Schwarz H, Marpe D, et al. Wireless broadcasting using the scalable extension of H. 264/AVC[C]//Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on. IEEE, 2005: 884-887.

[10] Lin W K, Chiu D M, Lee Y B. Erasure Code Replication Revisited[C]//Peer-to-Peer Computing. 2004: 90-97.

[11] Mohr A E, Riskin E A, Ladner R E. Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction[J]. Selected Areas in Communications, IEEE Journal on, 2000, 18(6): 819-828.

[12] Girod B, Aaron A M, Rane S, et al. Distributed video coding[J]. Proceedings of the IEEE, 2005, 93(1): 71-83.

AUTHORS

Jiejun Hu, received her master degree in Computer Science from Jilin University in 2012. Currently she is a PhD student in Jilin University, China. Her current research areas are computer network, future network and technology, resource allocation in network, and related applications.

Xilong Che received his MEng and PhD degrees in Computer Science from Jilin University in 2006 and 2009. Currently, he is a lecturer at the College of Computer Science and Technology, Jilin University, China. His current research areas are parallel and distributed computing, machine learning, and related applications. He is a member of the IEEE.