Restoration of Images Compressed by Hybrid Compression, based on Discrete Cosine Transform and Vector Quantization, over a Binary Symmetric Channel

Iman Elawady[1,2] and Caner Ozcan[2]

[1] Department of Electrical Engineering, Ecole Nationale Polytechnique d'Oran Maurice Audin, Oran, Algeria, e-mail: iman.elawady@enp-oran.dz

[2] Department of Software Engineering, Faculty of Engineering, Karabuk University, Demir Celik Campus, 78050, Karabuk, Turkey,
e-mail: canerozcan@karabuk.edu.tr

Abstract: The H.264 digital video compression standard (encoded files are MP4 files, but they can also be AVI or MKV) had a lot of attention and was successful for a long time due to the many features and techniques used to compress a video. Technically, a video represents a sequence of images, for the H.264 standard, the compression is based on reducing the redundancy in the same frame (an image), which is known as "intra prediction". This prediction generates modes used to rebuild the image (decompression) when we apply the same idea again but between the frames, we called it as inter prediction. The inter prediction operation also generates other modes. In terms of compression or transmission, predictions mean dependence on primary information, and it is estimated based on primary data, rather than sending all information. In this paper, we employ these prediction modes in the case of intra prediction to correct a single image compressed by vector quantization. This type of compression makes the restoration process very difficult as it produces block-shaped errors and no information about the original block is removed. The results in experimental studies section show a high restoration and correction ratio based on the PSNR as an evaluation parameter. For BER=0.01, we achieved a restoration average of about 0.6 dB, which produces an image with approximately the same PSNR as the original. The main contribution of our method, is that we open a new concept of non-filtering image correction techniques and improve on this correction method, by introducing new prediction modes, to correct the image.

Keywords: restoration; compression; H.264; intra prediction mode; vector quantization; binary symmetric channel; cyclic redundancy check; discrete cosine transform; median filter

1 Introduction

Data compression is the process of converting analog or very large amounts of discrete data into a lower rate data stream during communication over a digital communication link or storage in digital memory. To truly secure digital communications, applications of theoretical and practical data compression studies have suddenly become much more valuable [1]. While there are systems where bandwidth is generally cheap, e.g., cable TV and fiber optic links, the increasing amount of information users want to communicate or store in most systems requires data to be compressed both securely, reliably and efficiently. To achieve this, various techniques have been proposed [2-3]. The main aim of image compression is to eliminate the redundancy. The redundancy could be found in the level of the image content (spatial redundancy), or it could be found after encoding the image (code redundancy), which take us to the idea of the existence of variable length encoding as mentioned in [4-5] or fixed length encoding as mentioned in [6]. Another question could be asked in this case is the ability of restoring (decompressed) data in lossless way [7] [8] or in a lossy way [9]. The quantization plays an important role in all lossy compression applications [10] [11] in digital signal processing, which is a common procedure for converting a continuous or large set of sample values to a discrete (finite) set of sample values. Instead of direct quantization of pixel intensities in scalar quantization, vector quantization is a classical image compression technique that expands an image matrix into a long vector [12].

The vector quantization (VQ) technique [13] is one of the most interesting compression techniques due to its high compression ratio and the simplicity of its encoding and decoding operations. During the transmission, the vector quantization generally uses fixed-length coding. One of the important features of VQ is that it can reach high compression ratios with smaller block sizes compared to other compression techniques. It is known that using smaller block sizes, in block coding, leads to better subjective quality [14]. Thanks to these features, VQ has been frequently used in lossy data compression [15-18], pattern recognition [19] [20], probability density estimation and clustering [21] [22] applications.

Another advantage of the VQ technique is that we will not experience loss of synchronization, during transmission. So, the wrong packet will not affect the rest of the packets, and this is not the case with the variable compression technique, which corrupts the whole decoding process. It usually forces us to use too many redundant bits to avoid any problems in case of corruption. However, the noise in vector quantization will be in the form of blocks, which will really slow down the restoration process as it contains no information that can give us an idea of the contents of the missing block. In the H.264 standard, it is important to have some kind of reproduction of blocks based on 9 expectation modes, these 9 mods can predict the missing block based on its neighborhood producing a missing block very similar to the original [23-25]. However, we decided to use a blurred image to get minimal information about the bad blocks, then we created a mod, based on this information and applied it in the image.

There have been huge developments in the field of video compression that have produced many standards, like MPEG4, MPEG2, MPEG-1, JPEG2000, and JPEG [26]. The main aim of video compression is to lessen the amount of data desired to display a visual source and video coding used. The human visual system's (HVS) redundancy and the statistical organization of the video data are the two key aspects that support this. Due to redundancy in the HVS, some of the information in a video can be removed without changing how the viewer perceives the scene. Redundancy in both space and time can be found in the statistical structure of the video data. By using color subsampling and quantization to take advantage of this redundancy, high frequency coefficients are eliminated without altering the viewer's perception. Video compression is theoretically possible, without compromising visual quality, by lowering the quantity of data required [27].

Multiple standard image size reduction algorithms have good performance in case of noiseless transmission. However, these typical image size reduction algorithms usually cannot be easily adopted for wireless image transmission since the compressed images are hypersensitive to channel noise due to unfixed length coding that they based on. Our model based on the development of video coding techniques to compress the image we used H.264 transform to compress the image and the intra prediction modes in the restoration since it can correct the erroneous blocks by calculating the minimum difference between the neighborhoods blocks. This allows to predict the value of original block pixels.

The main contribution of our work can be summarized as follows:

- The study introduces a novel concept to unfiltered image correction techniques.
- This correction method is improved, by introducing new estimation modes to correct the image.
- Offers high restoration and correction ratios, based on the peak signal-to-noise ratio (PSNR), as an evaluation parameter is provided.
- A restoration average of about 0.6 dB, which produces an image with approximately the same PSNR, as the original is achieved.

Our article is organized in 4 sections. In the first section, information about discrete cosine transform-vector quantization (DCT-QV), intra prediction, CRC code and binary symmetric channel are given, respectively. Second, the details of the proposed method are explained. Third, the experimental results are presented. In the fourth, and final section, the evaluation and future work is given as conclusions.

2 Proposed Approach

In this section, we first explained the DCT-QV compression, intra prediction, CRC code, binary symmetric channel, and then we explained the details of our proposed intra prediction restoration.

2.1 Hybrid Compression Approach Based on DCT-QV

In our case, we applied DCT to the image before compression. This provides a good representation of the image in the frequency domain, which gives us a good quality of reconstructed (decompressed) image. The hybrid compression using DCT, and vector quantization diagram is illustrated below:

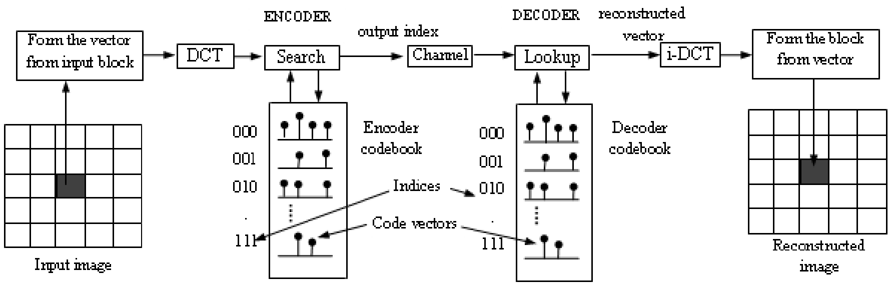

Figure 1
Hybrid compression (DCT with Vector quantization) procedure

The DCT-VQ is excellent for low-bit rate applications due to its great coding efficiency and straightforward decoder architecture. The general DCT-VQ algorithm consists of three key phases [28]:

- Create blocks from the image (usually they are 2x2, 4x4, 8x8, or 16x16).
- Create and index a codebook with the most accurate block estimation.
- The best estimate code index from the codebook is finally used to replace the original picture blocks.

In order to minimize distortion between the input vector and the selected code-vector, the fundamental tenet of vector quantization is to pair each input vector with a code-vector from the codebook [29]. Since the process of quantization is irreversible, it is impossible to determine the original value from the quantized value [30].

2.2 Intra Prediction Modes in H264 Video Compression

Three main types of intra prediction are defined, each of which, takes advantage of spatial correlation among pixels:

- The full macroblock prediction for luma sizes of 16x16
- The luma prediction size of 8x8
- Luma prediction sizes of 4x4 are also possible

The same image's macroblocks are used for intra prediction (INTRA). For the luminance component, there are two different kinds of prediction systems. The terms INTRA 4×4 and INTRA 16×16 can be used to refer to these two schemes [31]. A macroblock of size 16×16 samples is split into sixteen 4×4 subblocks in INTRA 4×4. These 4×4 subblocks are subjected to different intra prediction schemes. Nine alternative prediction modes are provided, as follows:

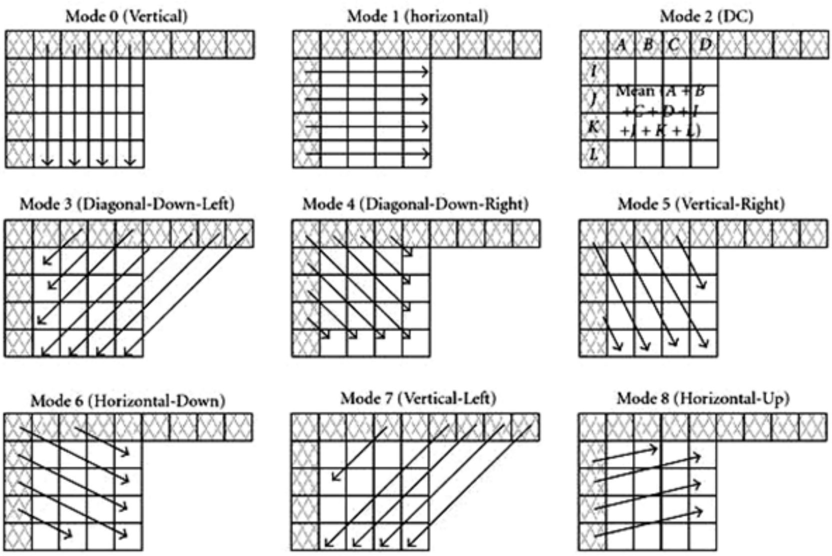

Figure 2

Nine modes of intra prediction

The macroblock samples are projected in mode 0 based on the samples above them that are nearby. The macroblock samples in mode 1 are anticipated from the nearby samples on the left. For prediction in mode 2, the mean of all adjacent samples is employed. The third mode is diagonally down-left. The fourth mode is diagonally down-right. The fifth mode is to the right and vertically. Mode 6 is horizontally downward. Mode 7 moves vertically and to the left. In mode 8 is horizontally upward. Four modes are employed for the 16×16 intra prediction of the luminance components.

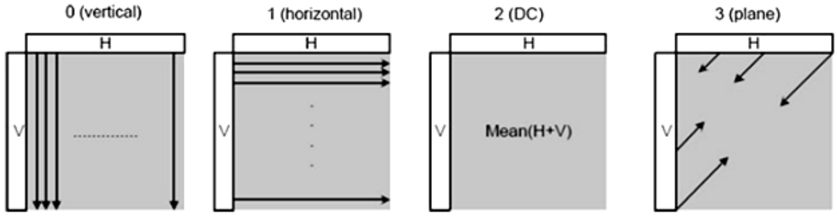

Figure 3
The division of a macroblock and a sub-macroblock for inter prediction

Similar to the prediction modes for the 4×4 block, it has three modes, mode 0 (vertical), mode 1 (horizontal), and mode 2 (DC). The fourth mode involves fitting the nearby samples with the linear plane function. Since the selected vector quantization is also 4×4 blocks, we have based our algorithm on 4×4 block prediction modes. Our proposal here, is to use the filtered image, with the median filter. This process is going to help us guess the original block, from the blurred image using the median filter with block size 10×10 as shown below.

Figure 4
Blurred image used in the first step of correction

2.3 CRC Code

Our aim during this study is to check the efficacy of the restoration technique, so we decided to use an error detection code based on the cyclic redundancy check (CRC). The CRC algorithms were developed and published by W. Wesley Peterson in 1961 [32]. The error-control coding method of CRC coding is used to find faults that happen during message transmission [33]. In CRC coding, the transmitter adds extra bits, termed the checksum, or syndrome, to each message word by applying a rule, and the checksum is then appended to the message word [34]. The receiver then applies the same criterion to the received word once it has been transmitted. The checksum should be zero, but if it's not, there was a mistake. The fundamental concept behind CRC algorithms is to simply interpret the message as a massive binary number, divide it by a different fixed binary number, and use the result as the checksum. The recipient can divide the message in the same way and compare the result to the "checksum" after receiving it (transmitted remainder).

Table 1
Certain CRC code generator polynomials

Common Name	Number of bits	Generator polynomial
CRC-12	12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$
CRC-16	16	$x^{16} + x^{15} + x^2 + 1$
CRC-32	32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

The polynomial used in this paper described as below, it adds 3 bits to each packet:

$$p(x) = x^2 + x + 1 \tag{1}$$

2.4 Binary Symmetric Channel Model

The channel diagram shown in Figure 5. It may be used to define the binary symmetric channel (BSC), which can be thought of as a Gaussian channel with hard decision. The channel matrix equation for the BSC is as follows:

$$[P(y/x)] = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix} \tag{2}$$

Two inputs (0, 1) and two outputs (1, 0, 1) make up the channel. The capacity to receive a 1 if a 0 is transmitted is equivalent to the chance of receiving a 0 if a 1 is transmitted, indicating that the channel is symmetric. P indicates that there is a chance of a mutual transition. Also, independent of the data bits are the error existences [35]. Although this is a simpler representation of a noisy channel, it nevertheless adequately depicts the intricacy of the overall issue. According to Equation 3, the capacity in this situation is:

$$C = 1 - H(p)\ in\ \frac{bit}{channel\ use} \tag{3}$$

Equation 4 provides the binary entropy function:

$$H(p) = -p\ log_2(p) - (1-p)log_2(1-p) \tag{4}$$

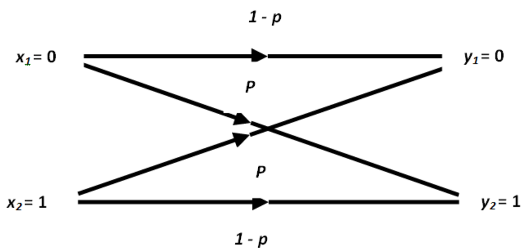

Figure 5
Binary symmetric channel

2.5 Intra Prediction Restoration

In this section we will explain how to generate a fixed length encoding compression using the H.264 and transform used in the VQ encoder as shown below. The concept of our algorithm is to use the discrete cosine transform (DCT) and vector quantization to compress our image. After that, we add CRC bits and send it over a binary symmetric channel. At the receiver, we detect the noisy packets and locate the noise. We reconstruct our image using a vector quantization decoder and the inverse of the DCT, and we obtain our noisy image. As a correction technique, we take the noisy image and blur it; if noise is detected, we apply the prediction mode in the same location as the noisy image, but in the blurred one. After getting the mode, we get the nearest block of the original one; if no noise detected, we keep the same block with no change. The steps of the system we recommend is shown in Figure 6. The pseudo code of our algorithm is given as follows:

Algorithm 1 Proposed Algorithm Steps

Step 1. Input: Read the original image I

Step 2. Encoder: Divide the original image into blocks size 4×4.

For each block:

1. Apply DCT.
2. Form a vector from the DCT coefficients.
3. Search for the closest matching code vector in the encoder using codebook.
4. Output the index of the matching code vector.
5. Add error detection code, CRC to each packet

Step 3. Transmission channel: Transmit the encoded data through a BSC which represent the channel noise.

Step 4. Decoder: Receive the indices from the channel.

For each index:

1. Look up for the corresponding code vector in the decoder based on the codebook.
2. Form a vector from the code vector.
3. Apply Inverse Discrete Cosine Transform to the vector.
4. Form the block form the transformed vector and construct the image.

Step 5. Error Detection: Detect the existence of errors for each received packet using the CRC code.
If errors are detected, mark the corresponding blocks as erroneous in the received image.
Step 6. Error Correction: If an error is detected:

1. Replace the erroneous blocks in the received image with the corresponding blocks from a high blurred version of received image.
2. Find the prediction mode from free noise blocks or restored one
3. Correct the blocks using the obtained mode and replace it with the predicted one.

If no error is detected no change is made to the received image.
Step 7. Output: Restored image I′ which is the corrected version of the received image.

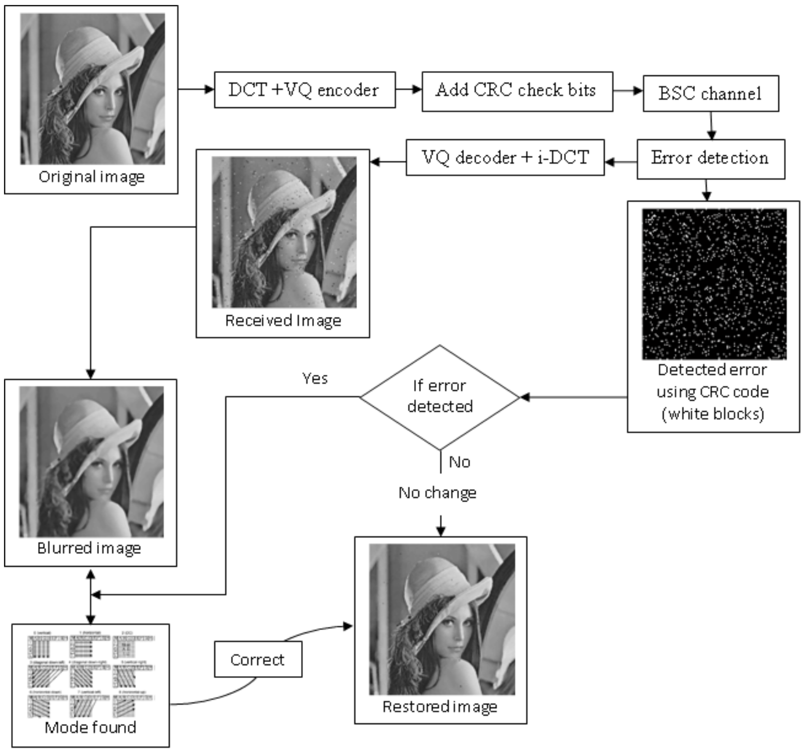

Figure 6

Main steps of image transmission and restoration

The idea of using blurred image gives us a corrected image but also details without noise. So, it is the best way to find the best prediction mode for the original one. Finally, the mod will be applied to the image and replace the wrong one.

3 Experimental Studies

For the simulation results, we used a gray level image encoded in 8 bits. The size of the images was 512x512 for the compression and the size of the universal codebook was 256 generated by the Linde, Buzo and Gray (LBG) algorithm [36]. The size of the block was 4x4 for error detection and we used 3 bits for every 8-bit packet. The compression ratio was 16 without considering error detection, while it was 10.67 with error detection.

To evaluate our results, we used the PSNR as evaluation parameter which represents the peak signal-to-noise ratio and its equal to:

$$PSNR = 10.\,log_{10}\left(\frac{d^2}{MSE}\right) \tag{5}$$

where d is the signal dynamic range. In the standard case of an image where pixel components are encoded in 8 bits, $d=2^8-1=255$.

Table 2

Results for different binary error rate (BER) using standard images

	PSNR of decomp.	**PSNR (received/restored) dB**		
BER		**10^{-1}**	**10^{-2}**	**10^{-3}**
Images				
lena	30.9539	17.4177/24.6285	25.9483/30.2037	29.7976/30.8498
boat	29.5384	16.9918/22.5302	25.1356/28.6145	28.9761/29.4548
goldhill	29.7015	17.1847/24.4932	25.4062/28.9672	29.0982/29.6413
barbara	26.9420	16.9162/22.5071	24.1527/26.3301	26.5459/26.8816
baboon	23.6522	16.2835/20.6164	21.9401/23.2550	23.4855/23.6202
cameraman	31.6319	16.4743/23.1898	25.3008/30.1969	30.4467/31.4995

The results in Table 2 shows a high restoration and correction ratio based on the PSNR as evaluation parameter to evaluate our results. For a high noise ratio BER= 10^{-1}, we achieved a restoration average about 6 dB depending on the nature of the image also the quality of noise free restored image. For BER=10^{-2}, we achieved a restoration average about 3.2 dB which enough to get an image similar to the original one. For BER=10^{-3}, we achieved a restoration average about 0.6 dB which produce an image having approximately the same PSNR of the original one. Average PSNR results of the proposed method for different BER are given in Figure 7. Visual results for different BER are shown in Figures 8, 9, and 10.

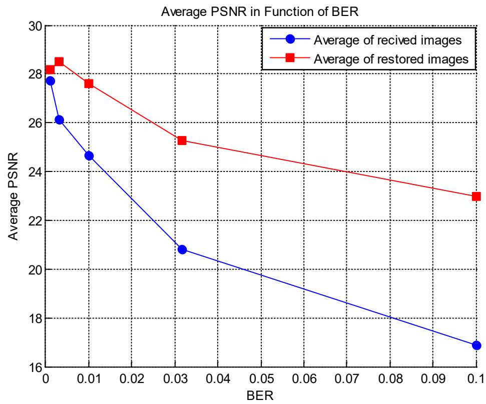

Figure 7
Average PSNR results of the proposed method for different BER values

Figure 8
Results of the proposed method for BER = 10^{-1}

Figure 9
Results of the proposed method for BER = 10^{-2}

Figure 10
Results of the proposed method for BER = 10^{-3}

Because it is difficult to restore a whole block of 16 pixels, the idea of using intra prediction modes from H.264 to correct the image shows high efficiency. However, in cases of high noise ratio, which is a critical case (BER=10^{-1}, Figure 7), the noise will not be removed even after blurring the image, causing the noise to propagate. The second issue is in the cyclic redundancy check, because it uses a mathematical logic operation, XOR (exclusive or). It could be a false detection or miss detection because the mathematical operation is correct. So, it is not about correction because if it's not detected, it won't be corrected. In this study, the issue of correction is solved, but there are still challenges with detection. Noise average versus restoration average in percentage is shown in Figure 11.

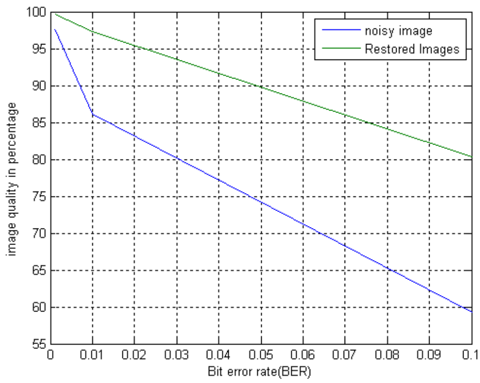

Figure 11
Noise average vs restoration average in percentage

Conclusions

In this study, we used prediction modes for internal predictions, to correct the compressed image using vector quantization. This type of compression produces block-shaped errors, making the restoration process very difficult, since information is removed from the original block.

Based on obtained results, our future work will be to develop a new detection technique, since the problem was not found in the correction technique, due to imperfections in the CRC code, which are based on mathematical operations that can produce erroneous results and will effect on the level of restoration fallibility, because our algorithm considers it as correct information, however, it’s not that which propagated the errors sometimes spread across the image.

This technique is extremely efficient, but it requires some calculation time, because we must extract the modes required for the correction operation; however, it is efficient enough and we intend to develop a method that will make this operation even faster.

Correspondingly, the restoration process can be more efficient by using the new intra prediction modes in the H.264 standard.

References

[1] Ungureanu, V. I., Negîrla, P., & Korodi, A. (2024) Image-Compression Techniques: Classical and “Region-of-Interest-Based” Approaches Presented in Recent Papers. Sensors (Basel, Switzerland), 24

[2] Garg, G., & Kumar, R. (2022, February) Analysis of different image compression techniques: a review. In Proceedings of the International Conference on Innovative Computing & Communication (ICICC)

[3] Satone, K. N., Deshmukh, A. S., & Ulhe, P. B. (2017, April) A review of image compression techniques. In 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA) (Vol. 1, pp. 97-101)

[4] Dimopoulou, M., San Antonio, E. G., & Antonini, M. (2021, August) A JPEG-based image coding solution for data storage on DNA. In 2021 29th European Signal Processing Conference (EUSIPCO) (pp. 786-790) IEEE

[5] Fuentes-Alventosa, A., Gómez-Luna, J., González-Linares, J. M., Guil, N., & Medina-Carnicer, R. (2022) CAVLCU: an efficient GPU-based implementation of CAVLC. The Journal of Supercomputing, 78(6), 7556-7590

[6] Bhambay, S., Poojary, S., & Parag, P. (2018) Fixed length differential encoding for real-time status updates. IEEE Transactions on Communications, 67(3), 2381-2392

[7] Devi Kotha, H., Tummanapally, M., & Upadhyay, V. K. (2019, May) Review on lossless compression techniques. In Journal of physics: conference series (Vol. 1228, No. 1, p. 012007) IOP Publishing

[8] Lam, D. K. (2024) Real-time lossless image compression by dynamic Huffman coding hardware implementation. J. Real Time Image Process., 21, 84

[9] Hu, Y., Yang, W., Ma, Z., & Liu, J. (2021) Learning end-to-end lossy image compression: A benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(8), 4194-4211

[10] Tao, D., Di, S., Chen, Z., & Cappello, F. (2017, May) Significantly improving lossy compression for scientific data sets based on multidimensional prediction and error-controlled quantization. In 2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS) (pp. 1129-1139) IEEE

[11] Miaou, S. G., & Chao, S. N. (2005) Wavelet-based lossy-to-lossless ECG compression in a unified vector quantization framework. IEEE Transactions on Biomedical Engineering, 52(3) 539-543

[12] Chang, S. Y., & Wu, H. C. (2022) Tensor quantization: High-dimensional data compression. IEEE Transactions on Circuits and Systems for Video Technology, 32(8) 5566-5580

[13] Gray, R. (1984) Vector quantization. IEEE Assp Magazine, 1(2) 4-29

[14] Chou, P. H., & Meng, T. H. (2002) Vertex data compression through vector quantization. IEEE Transactions on Visualization and Computer Graphics, 8(4), 373-382

[15] Chen, T., Liu, H., Ma, Z., Shen, Q., Cao, X., & Wang, Y. (2021) End-to-end learnt image compression via non-local attention optimization and improved context modeling. IEEE Transactions on Image Processing, 30, 3179-3191

[16] Zha, S., Pappas, T. N., & Neuhoff, D. L. (2020) Hierarchical lossy bilevel image compression based on cutset sampling. IEEE Transactions on Image Processing, 30, 1527-1541

[17] Elawady, I., Lakhdar, A. M., & Mustapha, K. (2016) The Noise Reduction over Wireless Channel Using Vector Quantization Compression and Filtering. International Journal of Electrical & Computer Engineering (2088-8708), 6(1)

[18] Elawady, I., Lakhdar, A. M., Beladgham, M., Habchi, Y., & Bassou, A. (2016) The effect of error transmission on compressed image using vector quantization with different codebooks. Electrotehnica, Electronica, Automatica, 64(1), 143

[19] Cheng, K. H., & Kumar, A. (2020) Efficient and accurate 3D finger knuckle matching using surface key points. IEEE Transactions on Image Processing, 29, 8903-8915

[20] Jiang, K., Zhang, T., Zhang, Y., Wu, F., & Rui, Y. (2020) Self-supervised agent learning for unsupervised cross-domain person re-identification. IEEE Transactions on Image Processing, 29, 8549-8560

[21] Tian, Y., Lei, Y., Zhang, J., & Wang, J. Z. (2019) Padnet: Pan-density crowd counting. IEEE Transactions on Image Processing, 29, 2714-2727

[22] Hedelin, P., & Skoglund, J. (2000) Vector quantization based on Gaussian mixture models. IEEE transactions on speech and audio processing, 8(4), 385-401

[23] Lin, K., Jia, C., Zhang, X., Wang, S., Ma, S., & Gao, W. (2022) Dmvc: Decomposed motion modeling for learned video compression. IEEE Transactions on Circuits and Systems for Video Technology, 33(7), 3502-3515

[24] Wiedemann, S., Kirchhoffer, H., Matlage, S., Haase, P., Marban, A., Marinč, T., ... & Samek, W. (2020) DeepCABAC: A universal compression algorithm for deep neural networks. IEEE Journal of Selected Topics in Signal Processing, 14(4), 700-714

[25] Bross, B., Chen, J., Ohm, J. R., Sullivan, G. J., & Wang, Y. K. (2021) Developments in international video coding standardization after AVC,

with an overview of versatile video coding (VVC) Proceedings of the IEEE, 109(9), 1463-1493

[26] Tun, E. E., & Aramvith, S. (2022) Reducing Complexity on Coding Unit Partitioning in Video Coding: A Review. Engineering Journal

[27] Richardson, I. E. (2004) H. 264 and MPEG-4 video compression: video coding for next-generation multimedia. John Wiley & Sons

[28] Nasrabadi, N. M., & King, R. A. (1988) Image coding using vector quantization: A review. IEEE Transactions on communications, 36(8), 957-971

[29] Grinstead, C. M., & Snell, J. L. (2012) Introduction to probability. American Mathematical Soc.

[30] Acharya, T., & Tsai, P. S. (2004) JPEG2000 standard for image compression: concepts, algorithms and VLSI architectures. John Wiley & Sons

[31] Ostermann, J., Bormans, J., List, P., Marpe, D., Narroschke, M., Pereira, F., ... & Wedi, T. (2004) Video coding with H. 264/AVC: tools, performance, and complexity. IEEE Circuits and Systems magazine, 4(1), 7-28

[32] Peterson, W. W., & Brown, D. T. (1961) Cyclic codes for error detection. Proceedings of the IRE, 49(1), 228-235

[33] Kumar, N., Kedia, D., & Purohit, G. (2023) A review of channel coding schemes in the 5G standard. Telecommunication Systems, 83, 423-448

[34] Sharma, D. D., & Choudhary, S. (2023) Pipelined and Partitionable Forward Error Correction and Cyclic Redundancy Check Circuitry Implementation for PCI Express® 6.0. 2023 IEEE Symposium on High-Performance Interconnects (HOTI), 1-8

[35] Dabbabi, O., Salariseddigh, M. J., Deppe, C., & Boche, H. (2023) Deterministic K-Identification for Binary Symmetric Channel. GLOBECOM 2023 IEEE Global Communications Conference, 4381-4386

[36] Linde, Y., Buzo, A., & Gray, R. (1980) An algorithm for vector quantizer design. IEEE Transactions on communications, 28(1), 84-95