

Research Article

Quality Enhancement of Synthesized Video by Improvement of VLC Using Artificial Neural Network

Hassan Farsi and Pouriya Etezadifar

Department of Electronics and Communications Engineering, University of Birjand, Birjand, Iran

Abstract: Progress of technology in recent decade's causes that video transmission via communication channels has met high demands. Therefore, several methods have been proposed to improve the quality of video under channel errors. The aim of this study is to increase PSNR for synthesized video by increasing channel encoder rate but in constant transmission rate. This is achieved by using intelligent neural network and Huffman coding in VLC blocks used in the MPEG standard to compress transmitted data significantly. Then, depending to the amount of compression by the proposed method, the compressed data is coded again using secondary channel encoder. The proposed method is able to increase channel coding rate without increasing the amount of information for each frame. This method provides more robustness for video frames against channel errors. The proposed method is tested for different source coding rates and several SNRs for channel and the obtained results are compared with a new method named Farooq method.

Keywords: Artificial neural network, channel coding, Huffman coding, variable bit rate, Variable Length Coding (VLC), video coding

INTRODUCTION

Today's, increasing growth of using wireless channels and mobile telecommunication and tend to serve the achievements of modern telecommunications for real-time multimedia data transmission such as video involves advance research in this area (Flierl and Girod, 2004). In wireless and mobile telecommunication due to nature of wireless environment and channel errors the quality of video is affected at the receiver. Several methods have been proposed to reduce the errors caused by the channel (Moreira and Farrell, 2006). Channel coding is an efficient method which is generally used to detect and correct the errors. However, channel coding was performed without considering source coding rate. In recent years, different methods as combination of channel and source coding have been introduced. These methods are known Joint Source and Channel Coding (JSCC). JSCC is generally based on channel estimation (Bystrom and Modestino, 2000; Cheung and Zakhor, 2000; Kondi *et al.*, 2002; Zhai *et al.*, 2006). This means that in order to have higher PSNR, higher channel coding rate is required. In this study, we propose a new JSCC method which increases channel coding rate but total transmission rate is maintained still constant. In fact by increasing channel coding rate more robustness of video frames can be provided. One of the most important problems in general methods is that the

channel information is required for modification of source and channel coding rates. However, in the proposed method the channel information is not needed. In other word, the proposed method is able to be applied on every source coding rate, independently and to improve the quality of the reconstructed video frames at receiver. In this study, Peak Signal to Noise Ratio (PSNR) and also Bit Error Rate (BER) are used as metrics measures to indicate the quality of the reconstructed video.

VIDEO FRAMES CODING

In this section, a Group of Pictures (GOP) is shortly introduced. Then video frames coding and decoding are explained using MPEG-4. In MPEG standard, three types of pictures are introduced by I, P and B briefly. GOP is made by combination of these pictures. As an example, a GOP structure is shown in Fig. 1.

A picture which is coded by its information is called I-picture. Thus JPEG compression algorithm is used for coding I-pictures (Salomon, 2004). The picture which is estimated by the nearest previous I or P pictures is called P-picture. For P-picture, motion compensation predictor is also used. The picture which is estimated by the nearest previous and next, I or P pictures is called B-picture. For B-picture, motion compensation predictor is also used, similar to P-picture. We use two forms of pictures; P-pictures and I-

Corresponding Author: Hassan Farsi, Department of Electronics and Communications Engineering, University of Birjand, Birjand, Iran

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

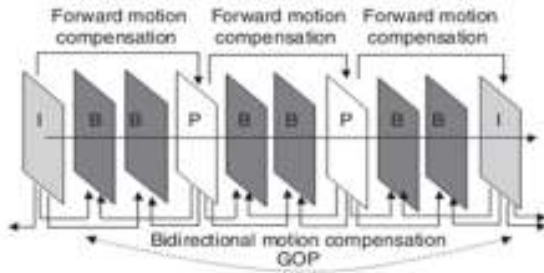


Fig. 1: An example of the GOP

pictures, through GOP for simulation, because using B-pictures (Bidirectional-pictures) leads to a delay in the system. The delay is created because prediction is performed on the next and previous frames. Thus, system is forced to wait until the next frames are entered and buffered such that the estimation of B-picture is possible to be performed. Thus we do not use B-picture in he proposed method.

Overview of the MPEG-4 encoding method: In this study, we use the MPEG-4 part 2. The structure of the encoder and decoder of I and P, are shown in Fig. 2 and 3, respectively. In following section some operations used in the encoder and decoder are described.

Reorder: After quantization of DCT coefficients, due to having many zero values, the obtained coefficients are arranged in zigzag form. Therefore the values related to different frequencies are grouped (Richardson, 2003). Zigzag arrangement is shown in Fig. 4.

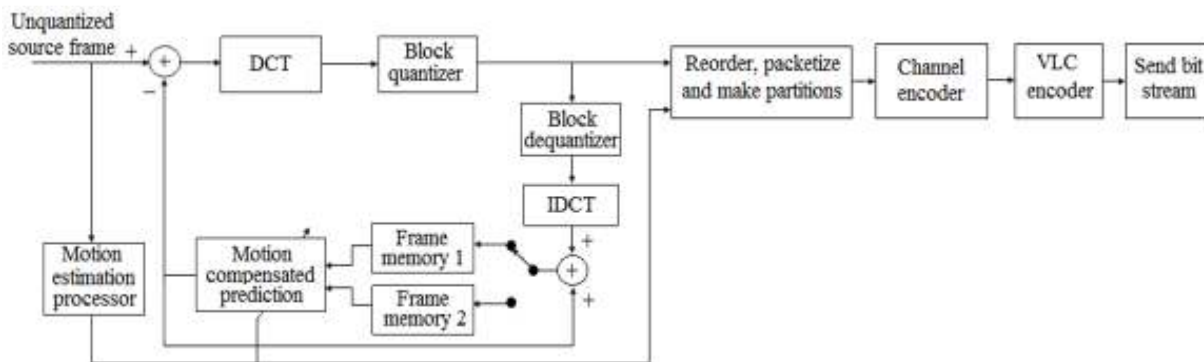


Fig. 2: MPEG-4 encoder for P-pictures

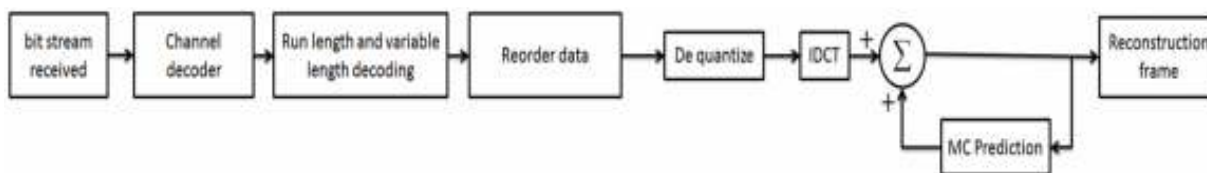


Fig. 3: MPEG-4 decoder for P-pictures

Data packetization: Occurring errors in data transmission may cause loss of synchronization between transmitted video frames. To overcome to this problem, a resynchronized mechanism is required. One method for synchronization is to use data Packetization. Thus, a number of markers are inserted in each packet for synchronization. Encoder divides each frame into packets and inserts markers related for Synchronization at the beginning of each packet. Encoder and decoder perform resynchronization when the markers located in the beginning of each packet are received. This procedure causes synchronization between video frames and avoids additive errors (Shi and Sun, 1999).

Data partitioning: In this model, the data packet is divided into two areas. The main idea is to separate data with higher values (DC coefficients of DCT matrix, coding model information and motion vectors) from data with lower values (AC coefficients of DCT matrix and residual errors). For I-picture transmission, the first area contains source coding information and DC coefficients. However the second area containing lower values has AC coefficients. For P-picture transmission, the first area contains source coding information and motion vectors whereas the second area contains the DCT data (texture, DC and AC coefficients). In data partitioning the markers (which are unique) are used for synchronization.

Motion compensation block: Motion compensation algorithm is based on the Minimum Mean Square matching Error (MMSE) between two successive frames using the following equations:

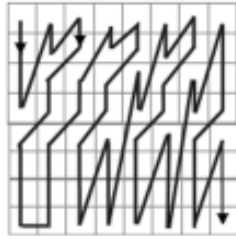


Fig. 4: Selection of DCT coefficients for the picking along the zigzag method

| | | | | | | | |
|----|----|----|----|----|----|----|---|
| 1 | -7 | -8 | 0 | 2 | 0 | 0 | 0 |
| 3 | 4 | -1 | -3 | -3 | -1 | 0 | 0 |
| -5 | 1 | 4 | 3 | 2 | -1 | -1 | 0 |
| -2 | -1 | 1 | 0 | 1 | 2 | 1 | 0 |
| -1 | 0 | -1 | -1 | -1 | -1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | -1 | 1 | 0 | -1 | 0 | 0 | 0 |
| 0 | -2 | 1 | 1 | -1 | 0 | 0 | 0 |

Fig. 5: DCT matrix coefficients quantized

$$MSE(d_x, d_y) = \frac{1}{M_1 N_1} \sum_{(m,n) \in W} (z[m, n, k] - b[m - d_x, n - d_y, k - 1])^2 \quad (1)$$

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \arg \min_{(d_x, d_y)} MSE(d_x, d_y) \quad (2)$$

In Eq. (1), the third parameter, (b), including (k) and (k-1), calls the pixels of two consecutive frames and two variables (m) and (n) show the pixel location in two dimensional spaces for each frame. The variable, (W), represents the search window and motion prediction vector is obtained by searching at this range. W = 15 is considered in this study. This means that the search window size is 16*16. Also prediction window size is considered 8*8. In other words, two consecutive frames within the search window are compared with each other and the least value is stored as a member of motion prediction matrix. After prediction of a frame completely and construction of the motion prediction matrix, the predicted value is subtracted from the previous frame. Next, the obtained value is applied to DCT block. After passing through the DCT block, the obtained value is quantized by 64 levels. At the receiver, as shown in Fig. 4, the received data is firstly dequantized and then the dequantized information is passed through IDCT block. Finally, using motion prediction matrix, the input frame is added to the previous frame and therefore the transmitted frame is reconstructed. In this study, we propose a new method to estimate and send Probability Density Function (PDF) based on the Huffman code and we use neural network to memorize transmitted bit stream for increasing the compression after passing through the VLC block. Next, we apply a secondary channel encoder which its rate depends on the difference between the bits made by VLC block encoder and the bits made using compression method proposed in this

study. In fact, by using memorize able data stream, we can implement Huffman algorithm on data stream in several times and therefore we can compress data and reconstruct bit stream at receiver.

Variable Length Coding (VLC): Variable length coding is final step before transmission of information on each frame. The VLC tries to reduce the transmission rate. This type of coding is performed in two different procedures:

In first procedure, due to existing many one values after quantization of DCT matrix as shown in Fig. 5, for instance, a few bits are considered for transmission. For example, only 3 bits are assigned to send the number of one value which represents as 11s. This code, 11s, is determined using specified table for the MPEG standard (Kondi *et al.*, 2002). The value of s is determined by the sign of one value. If the sign of the one value is positive (+1) then s = 0 and otherwise s = 1. On the hand, since the occurrence probability of the sample values except ±1 is low, the number of allocated bits for the these samples are higher than those bits allocated for ±1. For example, for sending the sample value of 7, 11 bits are required (Kondi *et al.*, 2002).

The second procedure in VLC uses this reality that the numbers of zero values in quantized DCT matrix is quite high, as shown in Fig. 5, for instance. After arrangement of DCT matrix elements in zigzag form, this procedure searches the next element is whether or not zero. If this element is zero, the next element value is checked. In each stage, the number of zeros is counted until reaching to a non-zero element or the entire DCT matrix elements are ended. After determining the number of repetition of zero elements between two non-zero elements, a set of binary code is given by the established table for standard MPEG (Kondi *et al.*, 2002). For example, to send the number of one which has 5 zeros look ahead, instead of allocation of 8 bits to each element (which leads to 48-bits) only 7 bits are sent which is represented by 000111s. The variable 's' is described as before. This procedure is based on two parameters used in the MPEG standard table (Kondi *et al.*, 2002) named "Run" and "Level". The value of Level is equal to the non-zero DCT matrix elements and the value of Run indicates the number of zeros after the selected element before reaching to the other non-zero element (MPEG Video Group, 1998).

CHANNEL CODING METHOD

In this study, Reed-solomon coding is used in which is a class of cyclic BCH code (Shannon, 1986; Carlson, 1986), linear and non-binary and it is made by GF (q) field (Shannon, 1948). The reason for using this type of coding is non-binary characteristics. Therefore, the information of each block can be directly coded without converting to binary. Thus, speed of video frame coding increases. In this method, assuming that

the encoded message 'C (X)' is transmitted and the noisy message 'r (X)' is received, then:

$$r(X) = r_0 + r_1X + \dots + r_{n-1}X^{n-1} \quad (3)$$

Also, the error polynomial 'e (X)' and its relation with transmitted and received messages are given as:

$$e(X) = e_0 + e_1X + \dots + e_{n-1}X^{n-1}$$

$$r(X) = c(X) + e(X) \quad (4)$$

Thus, according to Eq. (4), it can be shown that $r(\alpha^i) = c(\alpha^i) + e(\alpha^i) = e(\alpha^i)$. In this type of coding, both error and value location error are calculated. Due to having variable of 't' and $\beta_i = \alpha^{ji}$, the location and error value are obtained using Eq. (5). This type of coding is able to correct any pattern of errors with length of 't' or less than 't':

$$S_1 = r(\alpha) = e(\alpha) = e_{j_1}\beta_1 + e_{j_2}\beta_2 + \dots + e_{j_\tau}\beta_\tau$$

$$S_1 = r(\alpha^2) = e(\alpha^2) = e_{j_1}\beta_1^2 + e_{j_2}\beta_2^2 + \dots + e_{j_\tau}\beta_\tau^2 \quad (5)$$

$$\vdots$$

$$S_{2t} = r(\alpha^{2t}) = e(\alpha^{2t}) = e_{j_1}\beta_1^{2t} + e_{j_2}\beta_2^{2t} + \dots + e_{j_\tau}\beta_\tau^{2t}$$

In the simplest case, in this coding, assuming $t = 1$, the value and location error are calculated using Eq. (6):

$$\alpha^{j_1} = \frac{S_2}{S_1} \Rightarrow \text{Error Place} \quad (6)$$

$$e_{j_1} = \frac{S_1^2}{S_2} \Rightarrow \text{Error Value}$$

Channel coding rate is given by:

$$R_c = \frac{k}{n} \quad (7)$$

In Eq. (7), 'k' is the number of coded bits related to frame information, n is length of stream which contains the frame information and added information by channel encoder. The channel encoder is able to detect and correct $(n - k) / 2$ errors. Therefore, more n-k bits (the number of parity bits) results in more correction of message bits by channel encoder.

PDF FOR A STREAM WITH ARBITRARY LENGTH

In order to obtain and to present PDF, many bits are needed. For this aim, we first separate arbitrary length of bits from information. Next, we put every 8 bits of arbitrary stream into another buffer (Fig. 6). Therefore, buffer includes different numbers between 0-255 in decimal base instead of 0 and 1. In next stage, the number of repetition of the values at bit stream is calculated. We next create a histogram according to the

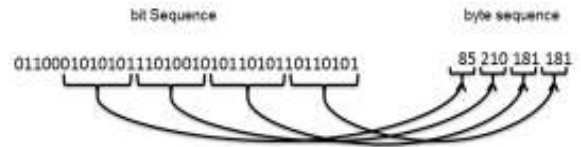


Fig. 6: Separating the bits from information stream for making stream as bytes

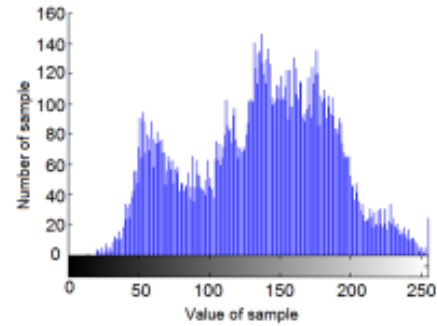


Fig. 7: Histogram of repetition numbers of samples in an information stream

number of repetition which are based on our PDF. For instance, Fig. 7 shows an example of histogram for a bit stream after conversion to the bytes stream.

After this stage, we should obtain PDF using created histogram (Fig. 7), where it is needed the following conditions to be satisfied:

$$f_X(X) \geq 0 \quad (8)$$

$$f_X(X) \leq 1 \quad (9)$$

$$\sum_{-\infty}^{+\infty} f_X(X) = 1 \quad (10)$$

It could be shown that the condition 8 is satisfied by investigation of histogram. Because the numbers of repetition of value always is positive and in case of no repetition, its value at histogram curve (Fig. 7) is zero. In order to satisfy the conditions 2 and 3 the numbers of repetitions for each point should be divided to the all existed repetitions Eq. (11):

$$f_X(X_i) = \frac{f_X(X_i)}{\sum_{k=0}^{255} f_X(X_k)} \quad (11)$$

In this situation, the conditions 2 and 3 are satisfied and therefore we could obtain PDF curve (Fig. 8) using histogram (Fig. 7). So far we have achieved the PDF related to the numbers in a stream. Since the PDF has different shape related to the stream values in order to code a stream, using Huffman algorithm, it is needed

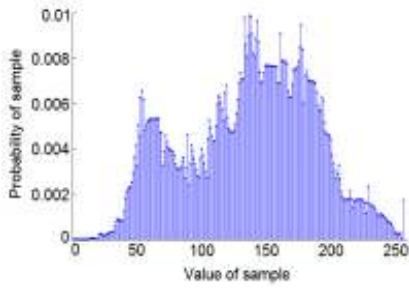


Fig. 8: Probability density function curve

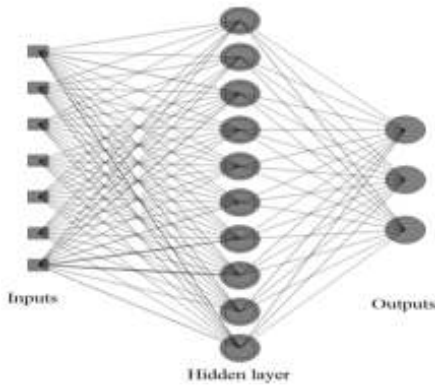


Fig. 9: Representation of a multilayers neural network structure

the information representing the PDF related to stream to be sent. In order to send the PDF, we fit a curve on the relative PDF and then the information representing the curve is sent with the related stream. Since the PDF changes are rapid and nonlinear (Fig. 8) a question arises. Which kind of curve could be adapted to result in minimum error? In order to answer to this question we use artificial neural network detailed in following section.

ARTIFICIAL NEURAL NETWORK

Artificial neural network contains a series of layers and nodes. Each network has input and output layers

necessarily (Vemuri, 1988; Gardner and Dorling, 1998). We could add several derivative layers arbitrarily. Every layer consists of arbitrary nodes which are known as neurons. The relation between layers is constructed by the links connected to the neurons. These links called network weights. In Fig. 9, a structure of e neural network with 7 entrances and 1 derivative layers and 3 output layer is observed. Note that in addition to weights, each neuron is under effect of another independent quantity which called bias. The way of impressibility of weights and biases at neurons of each layer to the subsequent layer is specified by a function named transfer function. This function can be linear or nonlinear in each layer and is given by user. Therefore, the important factors in network are weights and biases related to each neuron, whose value would affect the outputs value of neural network. Thus, in various methods used to improve the accuracy of neural network, it is attended to choose suitable values of weights and biases such that they correspond to least errors in neural network.

Neural network for description of PDF: In this part, we fit a function on probability density curve (Fig. 8) using neural network (Cobourn *et al.*, 2000). For this aim, first, a neural network with an untrained and hidden layer is created. Then, all weights and biases are introduced as optimum variables. Also, target function is introduced as Mean Square Error (MSE) between input matrix (containing integer numbers between 0-255) and output matrix (probability of samples occurrence). Therefore, in every step of optimization, weight values and net biases are updated such that the network errors decrease. Then, the resulting biases are used for construction of network. We have used perceptron neural network with a hidden layer containing 60 neurons. Transfer function of hidden neurons layers is “tansig” and output function is purelin. Also, input and output layers are mono neuron and all neurons in each layer have biases. For instance, Fig. 10 indicates the input and output of the neural

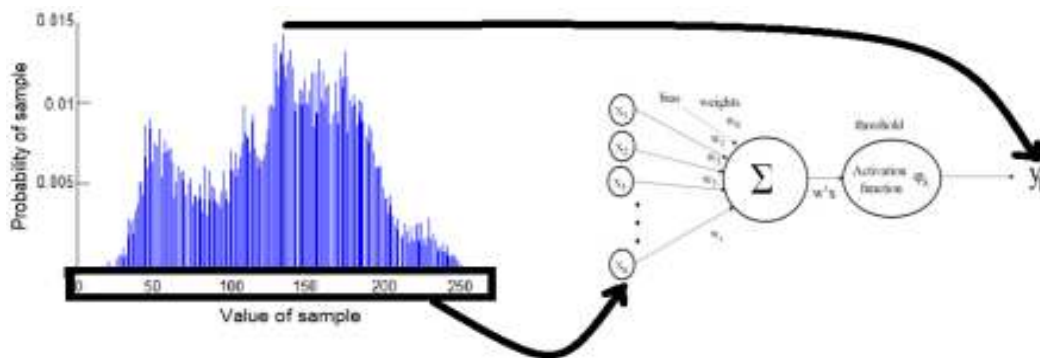


Fig. 10: Display the inputs and outputs used in the neural network

network, for instance. Neural network inputs are the values of horizontal axis and the objective functions of neural network are vertical values corresponding to horizontal values. Network weights are separated until the error is lower than 10^{-10} . The transfer function is selected by leave-one-out method (Fukunaga and Hummels, 1989; Stone, 1947) after testing on several PDFs. In this method, we are intended to teach neural network in spite of usual teaching network because in usual teachings, some information is discarded from test data. Test data is then given to network as input at end of each teaching and then weights and biases are obtained. Then, the output is calculated and compared with the correct values. If resulting error is less than specified extent, then calculated weights and biases are assigned as network's weights and biases. If error is more than specified extent, then teaching is repeated again. By performing this procedure, we can prevent over learning or inordinate teaching. However, in this study we try to retain the transfer function and permit over learning to be happened. This means that all points are given to the net as trained points and therefore we have the best fitting at the end.

Implementation of neural network for PDF: Since PDF is time variable because of up and down points in probability density curves, it is impossible to achieve a function for representation of these curves by using classic methods. Thus, we have used neural network for this purpose, which is performed for memorize able PDFs. For example, if we teach a neural network which is made by 60 neurons, 120 weights and 61 biases (Number of 60 biases for hidden layer and a single bias for the output layer) for an information stream, it possess PDF as shown in Fig. 8, outputs in the best case with supposing average errors less than 10^{-10} are presented in Fig. 11 and 12. In Fig. 11, target function and output values are drawn by using neural network. As observed, the output values have been fitted (dispersion of the target values from the output values) on target values as shown in Fig. 12, which is a proof on accuracy of inconsiderable errors on output data. However, it should be mentioned when the curve descends with sharp declivity and value tends to zero, it is possible the estimated function provides values less than these points and the condition 8 is violated. In this situation, negative values are replaced by the least positive values of estimated function. Because the estimated points with negative values are inconsiderable values and therefore we can replace them with the least positive values of estimated function.

THE CONCEPT OF MEMORIZE ABLE INFORMATION STREAM

According to the above, the PDF can be represented by using artificial neural network and in some cases where most values of PDF are restricted to finite samples; the PDF can be described using these

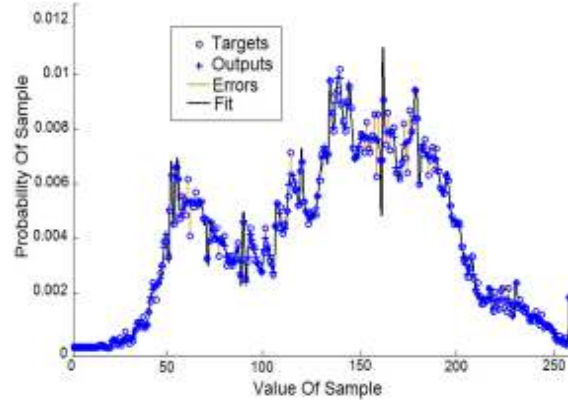


Fig. 11: Target function values with achieved output values using neural network

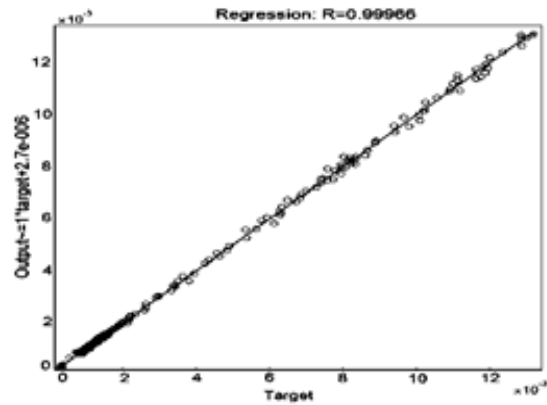


Fig. 12: Dispersion rate of target values from output values

samples. Regarding to previous section, it is obvious that we can describe PDF by using an artificial neural network or finite samples. Thus, it is possible to describe PDF using weight and biases obtained from intelligence teaching. Also, we can save the PDF behavior by using a series of weights and biases of the neural networks or finite samples. The required memory to save each obtained weigh, bias or sample is only 2 bytes. We put PDF information during each iteration of compression at the end of resulted bit stream and therefore this information is saved and then transmitted. In fact, we memorize PDF information related to bit stream in each iteration which is required during the reconstruction process. By memorizing data stream, we can implement Huffman algorithm on data string in several times. In this case, the PDF is approximated in every time and Huffman coding is then applied by the calculated PDF. By repetition of this method, we can reduce information string bulk. Information related to the PDF is added to the beginning of compressed bit stream. By repetition of this method, it is possible to bring down the amount of information for transmission.

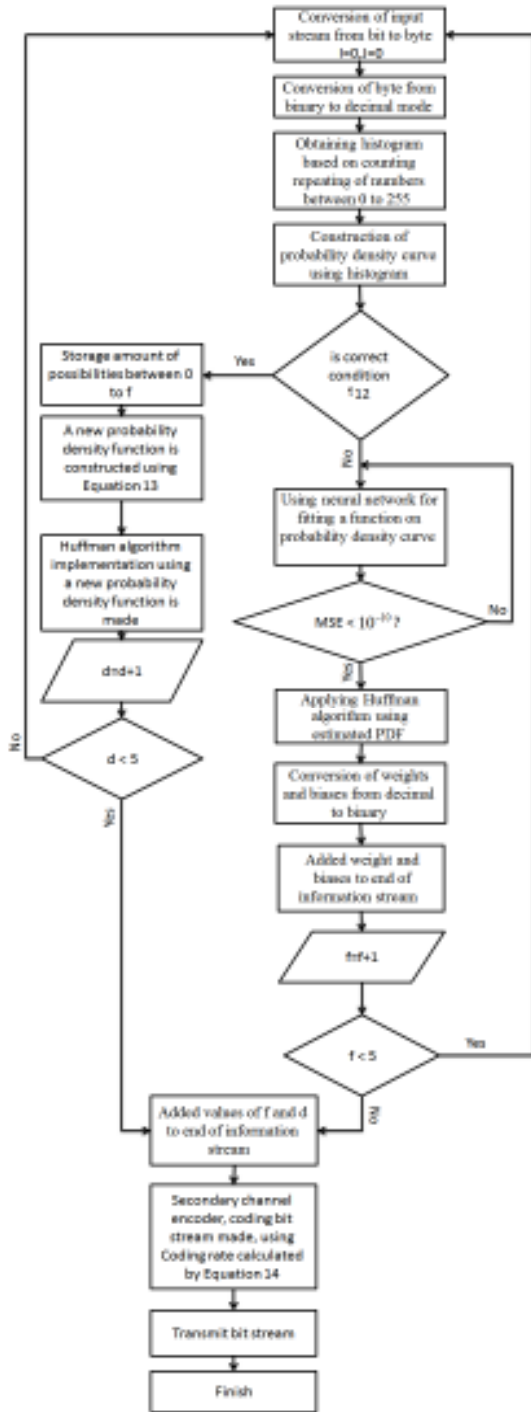


Fig. 13: Proposed method encoder

PROPOSED ALGORITHM

The proposed algorithm encoder: The Flowchart of the proposed encoder algorithm is shown in Fig. 13. The proposed method is detailed as follow. After preprocessing, the VLC output data is coded using Huffman code. Then, the secondary channel coding with the rate obtained from difference between bits made by VLC and bits made using Huffman coding is

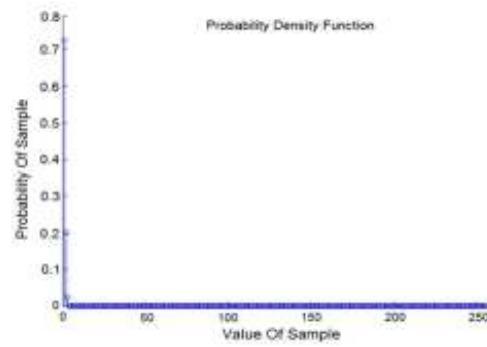


Fig. 14: A new probability density function curves obtained from the curves in Fig. 8 after the implementation of Huffman coding on strings data

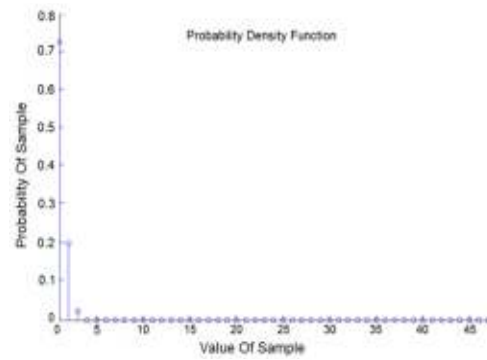


Fig. 15: Magnification of Fig. 14 in range $0 \leq \text{value of sample} \leq 47$

applied on the transmitted bit streams. Therefore, steps of the proposed algorithm are as follows:

- **Step 1:** The bit stream is converted to byte stream and therefore binary base is converted to decimal basic (PDF for a stream with arbitrary length).
- **Step 2:** Calculation of histogram regarding to the numbers of repetition between 0 to 255 in calculated byte stream in step 1 (PDF for a stream with arbitrary length).
- **Step 3:** Calculation of the probability density curve by using predetermined histogram to satisfy the conditions 8, 9 and 10 (PDF for a stream with arbitrary length).
- **Step 4:** If the number of iterations of the algorithm using neural network is more than 4 times the process is transferred to Step 16, otherwise neural network is used for fitting a function on probability density curve (Artificial neural network).
- **Step 5:** Applying of Huffman coding by using estimated PDF in step 4.
- **Step 6:** Conversion of weights and biases from decimal to binary and adding them to end of information stream.

Next Step is performed according to following explanations about the Huffman coding characteristics:

Huffman coding encodes the data stream such that the maximum number of zeros is provided (Huffman, 1952; Proakis, 1995). If the operation explained in step 1 is performed on bit stream made as output of Huffman coding, as shown in Fig. 14 and 15 (Fig. 15 is a magnification of Fig. 14), low value numbers have high repetition whereas larger value numbers have only a few repetitions. This issue has been obtained experimentally, therefore, it is needed a condition given in Eq. (12) to be checked and if this condition is satisfied and the next step is performed:

- **Step 7:** The bit stream created in step 6 converted to byte stream and then binary converted to decimal basis again (PDF for a stream with arbitrary length).
- **Step 8:** Calculation of histogram regarding to the numbers of repetition between 0 to 255 in calculated byte stream in step 7 (PDF for a stream with arbitrary length).
- **Step 9:** Calculation of the probability density curve by using predetermined histogram to satisfy the conditions 8, 9 and 10 (section 4).
- **Step 10:** If condition 12 is true, the process is transferred to next step, otherwise, it is returned to step 4:

$$\sum_{k=0}^j f_X(X_k) > 0.9 ; j = 1, 2, 3, 4 \quad (12)$$

- **Step 11:** The probability values 0 to j are saved.
- **Step 12:** In this stage, a new PDF is constructed according to the fact that the most PDF values are appeared in low value range. Since the other PDF values are negligible, it is assumed that all the samples have equal PDF value. This value is given by:

$$f_X(X_i) = \frac{1 - \sum_{k=0}^i f_X(X_k)}{256 - j} ; i = j + 1, j + 2, \dots, 255 \quad (13)$$

- **Step 13:** Performing of the Huffman coding using the created PDF in Step 12.
- **Step 14:** The number of 'j' first PDF values stored in step 11 and also, the value of 'j' is converted from decimal to binary basis and added to the end of compressed data stream for transmission.
- **Step 15:** If the number of repetition of the algorithm for the construction of PDF without using neural network is more than 4 times, the process is conducted to step 16, otherwise it returned to step 7 again.
- **Step 16:** The values 'f' and 'd' which indicate the number of repetition Huffman coding using neural network for PDF estimation and the sequence of

execution of 'f' or 'd' are added to the end of constructed bit stream. The sequence of execution means whether neural network has been used. Note that the value of 'd' or 'f' and the relative sequence are presented by 3 bits and 2 bits, respectively.

- **Step 17:** In this stage, the secondary channel encoder codes the bit stream using coding rate calculated by Eq. (14) and then coding rate is added to the end of stream:

$$R'_c = \frac{N'_b}{N_b} \quad (14)$$

In Eq. (14), R'_c is secondary channel coding rate, N_b is the number of bits generated by the VLC blocks (The number of final bits, should be equal N_b) and N'_c is the number of bits after compression using the Huffman coding.

- **Step 18:** The constructed bit stream is transmitted.

The proposed algorithm decoder: The Flowchart of proposed algorithm decoder is shown in Fig. 14. The process of the proposed method is described as follow:

- **Step 1:** Secondary channel coding rate is calculated from received bit stream and then the channel decoder decodes data using the coding rate.
- **Step 2:** The values of 'f' and 'd' are calculated from end of the decoded bit stream with the sequence of execution.
- **Step 3:** If the PDF is estimated using neural network (this means that the condition 12 has not been satisfied.) Q number of bits obtained using Eq. (15), separated from end of the data stream:

$$Q = (2 + 3 * N_n) * \frac{2 * \text{byte}}{\text{neuron}} * \frac{8 * \text{bit}}{\text{byte}} \quad (15)$$

In Eq. (15), N_n is the number of hidden layer neurons and because the output layer has a single neuron the number of weights in the hidden layer is equal to $2 * N_n$. Also, since each layer has a bias, we have N_n bias in hidden layer. Thus, for hidden layer weights and biases information, $3 * N_n$ Numbers should be sent. Since output layer is single neuron, only 2 numbers, one for the output weights and the other one for bias is sent. Also, for each weight and bias 2 bytes are considered:

- **Step 4:** Using weights and bias generated from step 3 and applying the neural network, the PDF is reconstructed.
- **Step 5:** According to the PDF obtained in step 4, the data stream is decoded using Huffman coding.
- **Step 6:** We check whether during the encoding process of the PDF has been estimated using neural

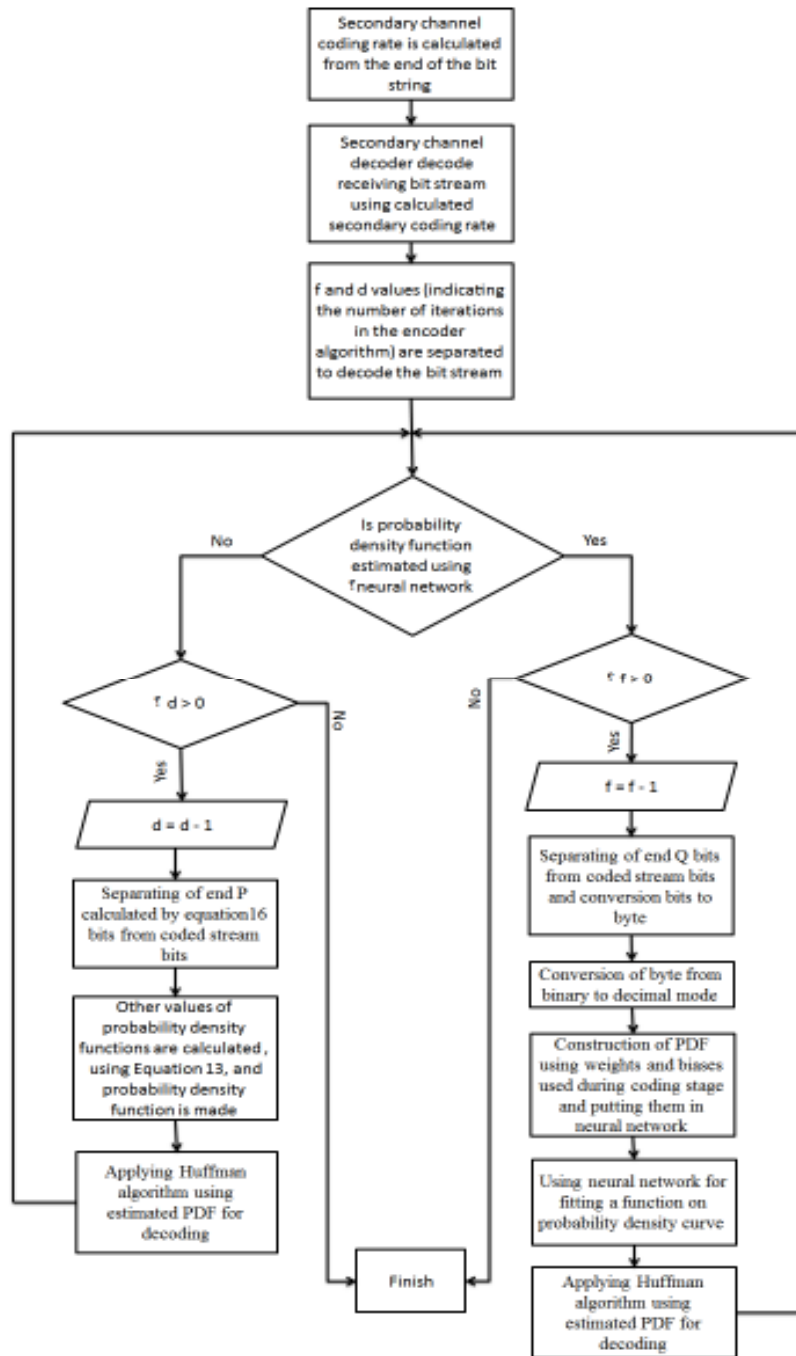


Fig. 16: Proposed method decoder

network. If neural network has been used, the decoding process jumps to step 3, otherwise, it goes to next step.

- **Step 7:** P-bits (calculated by Eq. (16) is separated from end of bit stream constructed in step 6. In Eq. (16), 'j' is the number of values which satisfies the condition given by Eq. (12):

$$P = j * \frac{2 * \text{byte}}{\text{neuron}} * \frac{8 * \text{bit}}{\text{byte}} \quad (16)$$

- **Step 8:** In this stage, another PDF values are calculated using Eq. (13) and thus the PDF is constructed.
- **Step 9:** According to the obtained PDF in step 8, the data stream is decoded by using Huffman coding.
- **Step 10:** The steps 1 to 9 are repeated 'f+d' times. If the number of iterations is less than 'f+d', then the decoding process is transferred to step 6. Otherwise, the decoder algorithm is ends (Fig. 16).

Table 1: Channel encoder rate R_c and also R'_c secondary channel encoder rate for two video sequences foreman and walk with difference transmission rates

| Source transmission rate | Type of video sequence which transmit | R_c | R'_c |
|--------------------------|---------------------------------------|--------|--------|
| 384 Kbps | Foreman | 0.4660 | 0.107 |
| 768 Kbps | Foreman | 0.2000 | 0.054 |
| 2 Mbps | Foreman | 0.0645 | 0.012 |
| 256 Kbps | Walk | 0.6000 | 0.175 |
| 512 Kbps | Walk | 0.3300 | 0.084 |
| 1.5 Mbps | Walk | 0.1290 | 0.026 |

EVALUATION OF THE PROPOSED METHOD

First, we introduce several parameters used in our simulation and then we compare the experimental results of the proposed method with the results of a new method indicated by Farooq *et al.* (2009). In this study, the channel is Binary Symmetric Channel (BSC), the occurred error is Additive White Gaussian Noise (AWGN) and fading has Rayleigh PDF. We have used two video files named Foreman and Walk (Video Test Media, <https://media.xiph.org/video/derf>). The number of Foreman video frames is 123 that 3 frames are coded as I-picture and the rest frames are divided into three parts. Each part has 40 frames which are coded and transmitted as P-pictures. It should be noted that 3 frames coded by I-picture, at the beginning of each part, are transmitted. The numbers of Walk video frames are 105 in which 5 frames are coded as I-picture and the rest frames are divided into five parts. Each part contains 20 frames which are coded as P-picture. It should be noted that 5 frames coded by I-picture, at the beginning of each part are transmitted. Frame size is equal to 288*352. Also frame transmission rate is 25 frames/sec. This simulation has been performed for three source transmission rates. Since the number of transmission frames is constant for a fixed source coding rate, higher transmission rate corresponds to higher channel coding rate and therefore results in more robustness against noise for video frames. Channel encoder rate, R_c , also the secondary channel encoder rate, R'_c , for two video files with different rates are shown in Table 1.

As shown in Table 1, due to the high compression rate of bit sequences, the secondary channel coding rate is less than the original channel coding rate. This means that the number of bits used in the secondary channel encoder is more than these bits used in the original channel encoder. Therefore, the transmitted frames are significantly more robust against channel errors.

As shown in Table 1, by increasing transmission rate the channel encoding rate increases. This means that the energy of the received frames increases. After simulation of the proposed method, the obtained results have been compared with the results of Farooq *et al.* (2009). Figure 17 to 19, show the obtained PSNR by applying the proposed method in comparison with

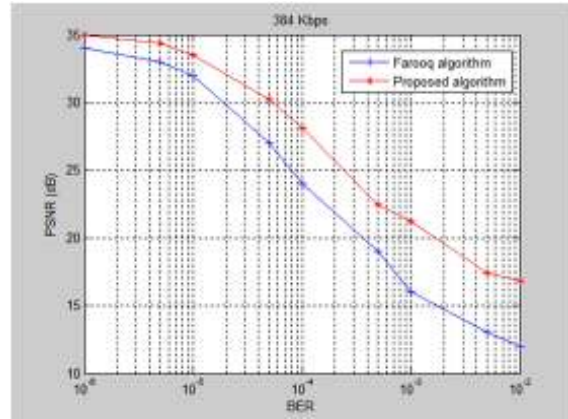


Fig. 17: Comparison of PSNR for 123 frames of Foreman between the proposed method and Farooq algorithm, with source coding rate = 384 Kbps

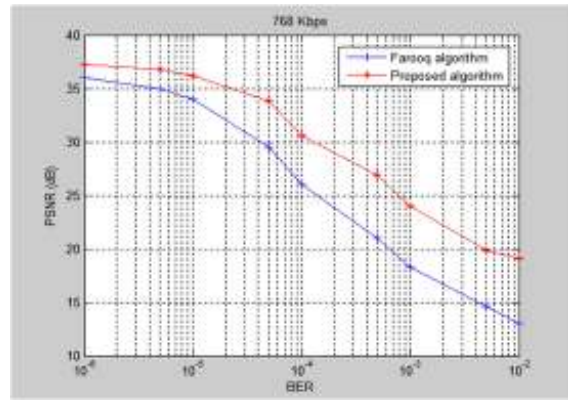


Fig. 18: Comparison of PSNR for 123 frames of Foreman between the proposed method and Farooq algorithm, with source coding rate = 768 Kbps

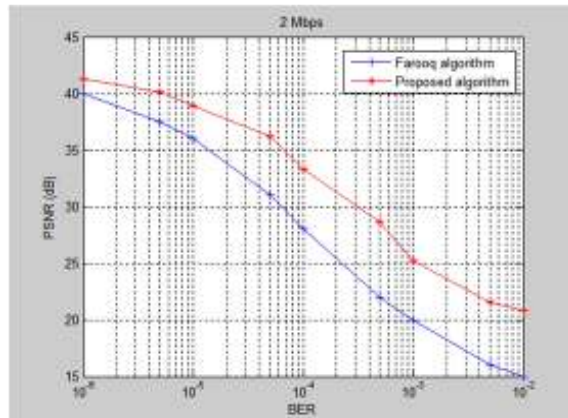


Fig. 19: Comparison of PSNR for 123 frames of Foreman between the proposed method and Farooq algorithm, with source coding rate = 2 Mbps

Farooq method for all 123 frames of Foreman. The only difference in Fig. 17 to 19 is source coding rate. This

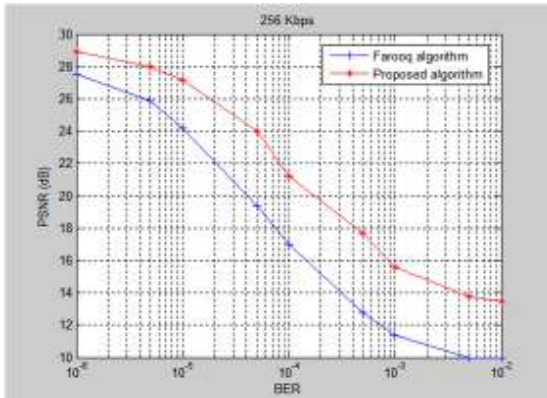


Fig. 20: Comparison of PSNR for 105 frames of walk between the proposed method and Farooq algorithm, with source coding rate = 256 Kbps

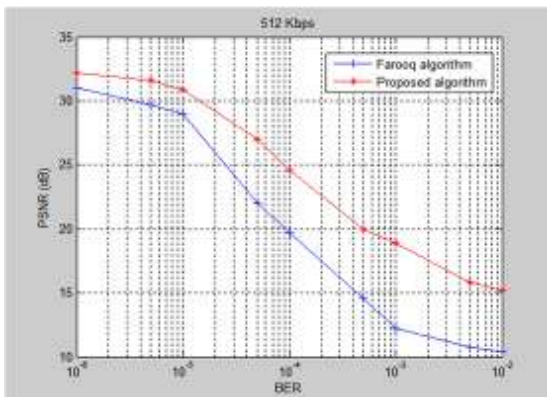


Fig. 21: Comparison of PSNR for 105 frames of walk between the proposed method and Farooq algorithm, with source coding rate = 512 Kbps

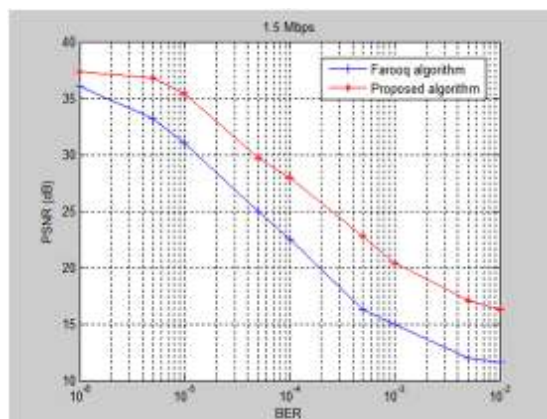


Fig. 22: Comparison of PSNR for 105 frames of walk between the proposed method and Farooq algorithm, with source coding rate = 1.5 Mbps

means that if source coding rate increases then the transmission of the frames is robust against channel noise and distortion. Also, Fig. 20 to 22, show the

obtained PSNR by applying the proposed method in comparison with Farooq method for all 105 frames of Walk. As shown the proposed method through embedding channel encoder bits in transmitted frames is able to increase channel encoding rate without increasing the transmission bit rate. Therefore the quality of the reconstructed frames has considerably increased.

CONCLUSION

In this study, we proposed a novel method in which the received video frames contain higher PSNR in comparison with the new method indicated by Farooq *et al.* (2009). Therefore, the reconstructed frames have higher quality. In the proposed method, due to compression of the obtained bits by VLC block (obtained from difference between bits made by VLC block encoder and bits made using Huffman method) and replacing the bits created by the secondary channel encoder, we could increase quality of the received video frames in comparison with the new methods. The proposed method through increasing the channel coding rate and therefore more protection of transmitted bits and maintaining overall transmission rate constant, could improve the quality of the reconstructed video frames.

REFERENCES

- Bystrom, M. and J.W. Modestino, 2000. Combined source-channel coding schemes for video transmission over an additive white Gaussian noise channel. *IEEE J. Sel. Areas Commun.*, 18(6): 880-890.
- Carlson, B., 1986. *Communication Systems. An Introduction to Signals and Noise in Electrical Communication*. 3rd Edn., McGraw-Hill, New York.
- Cheung, G. and A. Zakhor, 2000. Bit allocation for joint source/channel coding of scalable video. *IEEE T. Image Process.*, 9(3): 340-356.
- Cobourn, W.G., L. Dolcine, M. French and M.C. Hubbard, 2000. A comparison of nonlinear regression and neural network models for ground-level ozone forecasting. *J. Air Waste Manage.*, 50(11): 1999-2009.
- Farooq, M.S., R.W. Heath and B. Cornard, 2009. Joint source-channel distortion modeling for MPEG-4 video. *IEEE T. Image Process.*, 18(1): 90-105.
- Flierl, M. and B. Girod, 2004. *Video Coding with Superimposed Motion-Compensated Signals: Applications to H.264 and Beyond*. Springer, Boston, pp: 176, ISBN: 1402077599.
- Fukunaga, K. and D.M. Hummels, 1989. Leave-one-out procedure for nonparametric error estimate. *IEEE T. Pattern Anal.*, 11(4): 421-423.

- Gardner, M.W. and S.R. Dorling, 1998. Artificial neural networks (the multilayer perceptron), a review of applications in atmospheric sciences. *Atmos. Environ.*, 32(14-15): 2627-2636.
- Huffman, D.A., 1952. A method for the construction of minimum redundancy codes. *Proc. IRE*, 40: 1098-1101.
- Kondi, L.P., F. Ishtiaq and A.K. Katsaggelos, 2002. Joint source-channel coding for motion-compensated dct-based snr scalable video. *IEEE T. Image Process.*, 11(11): 1043-1052.
- Moreira, J.C. and P.G. Farrell, 2006. *Essentials of Error-control Coding*. John Wiley and Sons, Chichester.
- MPEG Video Group, 1998. MPEG-4 video verification model version 11.0. In ISO/IEC JTC1/SC29/WG11 MPEG98/N2172, Tokyo, Japan.
- Proakis, J.G., 1995. *Digital Communications*. Mcgraw Hill, Hardcover.
- Richardson, I.E.G., 2003. *H.264 and MPEG-4 Video Compression, Video Coding for Next-generation Multimedia*. Wiley, New York.
- Salomon, D., 2004. *Data Compression: The Complete Reference*. 3rd Edn., Springer, New York.
- Shannon, C.E., 1948. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27: 379-423.
- Shannon, C.E., 1986. Communications in the presence of noise. *Proc. IEEE*, 86(2): 447-458.
- Shi, Y.Q. and H. Sun, 1999. *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms and Standards*. CRC Press, ISBN: 0-8493-3491-8.
- Stone, M., 1947. Cross-validators choice and l assessment of statistical predictors. *J. Royal Stat. Soc.*, 36: 111-147.
- Vemuri, V., 1988. *Artificial Neural Networks, Theoretical Concepts*. IEEE Computer Soc., Washington, DC.
- Zhai, F., Y. Eisenberg, T. Pappas, R. Berry and A. Katsaggelos, 2006. Rate-distortion optimized hybrid error control for real-time packetized video transmission. *IEEE T. Image Process.*, 15(1): 40-53.