

Color video compression using color mapping into textured grayscale video frames

Sambuddha Kumar · Sumana Gupta

Received: 22 June 2012 / Accepted: 2 December 2013 / Published online: 28 December 2013
© Springer-Verlag London 2013

Abstract Color video compression has become a necessity in the advent of newer applications of color video and its increasing size for higher definitions. New approaches are used to achieve more compression over the traditional and standard compression techniques. Considerable progress has been made of color transfer-based compression techniques that do not modify standard encoders but add some pre-processing and post-processing steps. We present a novel, simple yet robust, color transfer-based compression technique that can be integrated with the standard encoders like MPEG-2, H.264/AVC, etc. We achieve the compression by discarding the color components from all frames in a video sequence except those of the Intra (I) frames. The colored intra frames are converted to gray-textured images. The method is developed based on embedding the colors of the intra frames to low visibility high frequency textures of its gray luminance image. This is achieved by decomposing the luminance of the intra frame using the non-decimated DWT or discrete wavelet frame transform and replacing the bandpass subbands by the chrominance signals. The lowpass band is same as that of the luminance signal. At the output of the decoder, the colored intra frames are recovered using the wavelet transform. Instead of calculating motion vectors, we identify and reuse the motion vectors present in the decoder. The remaining luminance frames of the video are colored with the help of identified motion vectors and the color information of the preceding frames. The proposed codec

considerably improves the compression ratio achievable by the standard codec, at the cost of slight increase in computation time at the decoder end.

Keywords Color transfer · Discrete wavelet transform (DWT) · Discrete wavelet frame transform (DWFT) · H.264 · Intra (I) frame

1 Introduction

In recent times, digital video application has grown exponentially from the conventional to several newer ones. As the size of raw video is increasing due to increase in quality, the transmission of these visual media in digital form alone will require far more bandwidth than what is available for the Internet, TV, or wireless networks. On the other hand, to maintain the quality of service, transmit time of the videos cannot be allowed to increase without limit. The limitation on available bandwidth demands that video compression be pushed to its maximum limit. In most cases, encoder side of the digital video transmission can afford a large storage and processing power. But the decoders are generally attached to the end user, who can supply neither a large storage nor large power for process. As a result, additional complexity that may be required to achieve higher compression rate is not acceptable in the most applications.

Over the last few decades, various compression paradigms for color video compression have been suggested. With the success of the various video coding standards based on hybrid motion compensated methods, the research community has been investigating new approaches that can address next-generation video services besides providing more advantage in terms of achieved compression. The

S. Kumar (✉) · S. Gupta
Department of Electrical Engineering, Indian Institute of Technology, Kanpur 208016, India
e-mail: sambuddhakumar@gmail.com

S. Gupta
e-mail: sumana@iitk.ac.in

new functionalities of these approaches provide higher robustness, interactivity and scalability with lower complexity. Most of the recent advances in video coding schemes approach to the methods of distributive video coding, texture-based video coding, scalable video coding or multi-view video coding, etc., as mentioned and discussed in [1].

In the recent past, color transfer technique has been used for video compression. Color is selectively removed from the input video before feeding it to the standard codec. This process increases the compression capability of the codec by reducing most of the chrominance components to zero. The coded video is then decoded by the standard decoder to produce partially colored video. The color transfer techniques are then used to colorize the video.

Color transfer is a recently proposed method for colorizing grayscale images. The task of color transfer involves transferring one image's color characteristics into another. Landmark works in the area of color transfer has been reported in [2] and by Levin et al. [3], in which they proposed an optimization of colorization scheme. These traditional schemes cannot be used for the video compression since these methods for colorization are too slow and require considerable manual intervention. Work by Kumar et al. [4] describes the application of color transfer in video compression. They proposed a motion-based color transfer algorithm in a hybrid video compression scheme using standard MPEG-1 codec. In some other reports [5], the decoder turns out to be very complex. A major reason for this is that motion vectors that are required for effective transfer of color, are calculated separately for color transfer at the decoder. As calculation of motion vectors is computationally intensive, it adversely affects the application prospects of the algorithm. In [6], authors used motion vector information which were tapped from standard codec. Semi automatic process for colorization stated in [7] is computationally simple and forms the basis of the color transfer algorithm used in this work. We prefer this approach to the technique [2] that uses manual input as a form of color scribble.

Queiroz et al. [8] described a reversible method to convert color graphics and pictures to gray images. The technique was used to print color images with black and white printers. We have applied this concept to develop a new video compression technique. The method is based on mapping colors to low-visibility high-frequency textures that are applied onto the gray scale frame. The transmitted

video has the size of a gray-scale equivalent with useful color information embedded in it, so that we can recover the complete color information on the decoder side. Instead of discrete wavelet transform (DWT), as used in the work [8], we have used discrete wavelet frame transform (DWFT) for mapping colors to grayscale image. DWFT avoids down-sampling using an overcomplete wavelet decomposition [9]. Consequently, it is both aliasing free and translation invariant [10]. Thus, DWFT-based scheme gives better result (in terms of PSNR and visual perception) than the DWT-based technique. On decoder side, after recovering color in I frames, we have used an edge enhancement technique on the chrominance components. It is based on the work described in [11] to compensate the loss of edge information present in high-frequency texture of luminance component of I frames.

The paper is divided into seven sections. Details of the proposed encoder scheme are discussed in Sect. 2. Details of the proposed decoder schemes are given in Sect. 3. Results regarding achieved compression ratio and decoded video quality are discussed in Sect. 4. In Sect. 5, rate-distortion analysis of proposed codec is presented. Section 6 discusses the performance of proposed codec with scalable coding scheme. Section 7 contains the conclusion.

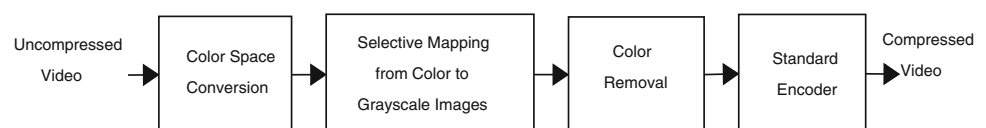
2 Proposed encoder

The proposed encoder (block diagram shown in Fig. 1) primarily consists of a standard encoder with some pre-processing blocks. Pre-processing includes color space selection and conversion, selective mapping from color to gray frame for I frame and color removal for other frames.

2.1 Color space selection and conversion

Choice of color space is important for the process of color embedding into textured gray frames and color transfer (discussed in next section). These methods require the color space which has uncorrelated chrominance and luminance channels. Traditionally, color transfer techniques [2, 3, 12] have used $l\alpha\beta$ color space for its property of uncorrelated channels. Most of the standard video codecs use $YCbCr$ color space to represent and store the video frames. The conversion to $YCbCr$ provides two benefits: first, it improves compressibility by providing decorrelation of the color signals; and second, it separates

Fig. 1 Proposed encoder block diagram



the luminance signal, which is perceptually much more important, from the chrominance signal, which is less perceptually important and which can be represented at lower resolution to achieve more efficient data compression [4].

We use YC_bC_r color space because, firstly, it is uncorrelated enough to not create artifacts while color transfer and, secondly, as it is commonly used in compression schemes, using it saves a lot of overhead which is involved in converting to and from other color space like $l\alpha\beta$ or YUV. Normally, raw videos come in RGB format. So as the first step of our proposed work, the color space of the input video is converted into YC_bC_r format.

2.2 Selective mapping from color to grayscale images

The second unit of the pre-processing step is mapping colors to low-visibility high-frequency textures that are applied onto the luminance component of the Intra frames. This method maps colors to texture. However, instead of having a dictionary (or palette) of textures and colors, it produces a continuum of textures that naturally switch between patterns without causing visual artifacts. In this stage, following functions are applied on the I frames.

- Using one level of the discrete wavelet frame transform (DWFT), the luminance part of each I-frame is divided into four subbands: $Y \rightarrow (S_l, S_h, S_v, S_d)$, corresponding to the low-pass, horizontal, vertical and diagonal (high-pass in both directions) subbands, respectively. Use of DWFT saves the subbands to go through decimation filter banks. So the subbands are having same dimension as Y band. As there is neither down-sampling in pre-processing before encoder nor up-sampling in post-processing after decoder on the luminance or chrominance planes, this method ensures better result than a method based on DWT, as proposed in [8]. Corresponding results are shown in table.
- The horizontal subband (S_h) is replaced by C_b and vertical subband (S_v) is replaced by C_r . The idea behind our work is color embedding into gray components. So we have to replace some luminance components with chrominance channel data and those luminance component data will be lost. The low-pass band (S_l) is the utmost important as human visual system is most sensitive to this band information. On the other hand, both the high-pass band and the chrominance signals adapt well to scene object contours. Hence, the texture

pattern changes appear to be natural and are mostly invisible. The high-pass S_d band also keeps the information about diagonal edges which are important for overall edge information. So we are left the bandpass components, namely the horizontal and vertical subbands to be replaced by the chrominance bands. Also loss of these two luminance subbands has the least effect on the visual perception.

- An Inverse discrete wavelet frame transform (IDWFT) is carried to recompose the monochrome frame as $(S_l, C_b, C_r, S_d) \rightarrow Y'$.

2.3 Color removal

Once the I frames are processed, selective de-colorization of the video is performed. Chrominance components (C_b, C_r) of all the frames except the I frames are set to zero. All the Intra coded (I) frames are already in grayscale format.

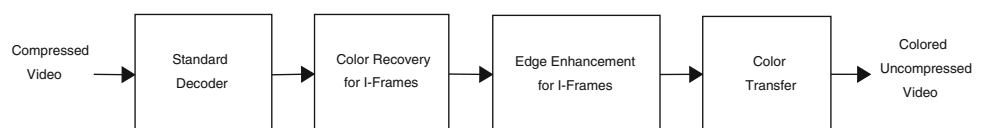
2.4 Standard encoder

Any standard encoder that uses motion compensation can be used for compressing the partially colored video. In our work, to prove the concept of effective color video compression using color transfer, the baseline profile of H.264 (i.e. with I and P slices only, no B slices) is used [13]. There are various ‘levels’ offered in H.264/AVC standard. In our work, we have used default level as ‘2’ which supports resolution up to 352×288 at highest frame rate as 30 fps. Default Quantization parameter (QP) is opted as 16. The period of I frames is kept to be 15 (I frames repeat after every 15 frames). The main profile (with I, P and B slices) of H.264/AVC is used afterwards to verify the results. Performance of the proposed encoder using H.264/AVC main profile and baseline profile emerges to be same for achieved additional compression ratio and encoding time, so primarily the result for proposed encoder consisting of H.264/AVC baseline profile is discussed.

3 Proposed decoder

Proposed decoder consists of a standard decoder along with a post-processing block (block diagram shown in Fig. 2). This Post-Processing block first recovers color to the Intra

Fig. 2 Proposed decoder block diagram



(I) frames, improves its edge details and subsequently transfers color to the uncolored frames.

3.1 Standard decoder

The compressed video sequence generated at the encoder is fed as an input to the standard decoder, which decodes the compressed sequence and produced the grayscale video sequence which we initially fed to the standard encoder. This uncolored video is to be processed so that we can get the original colored sequence.

The baseline profile of H.264 (i.e. with I and P slices only, no B slices) is used for the verification of the proposed scheme. We have used default level as ‘2’ which supports resolution up to 352×288 at highest frame rate as 30 fps. Default quantization parameter (QP) is opted as 16. The period of I frames is kept to be 15.

3.2 Recovering color in I frames

As the first stage of post-processing color recovery in the I frames is done. Colored I frames will be considered as reference for the initiation of color transfer block of post-processing scheme. Following steps are carried out for color recovery in I frames:

- One level of the discrete wavelet frame transform (DWFT) converts the gray scale form of Intra frames into subbands: $Y' \rightarrow (S_l, S'_h, S'_v, S_d)$
- The high-pass horizontal and vertical subbands (S'_h and S'_v , respectively) are interpreted as the C_b and C_r components.
- The high-pass subbands are set to be zero. The information originally contained in these bands was lost in the encoding process, which leads to a slight loss of detail in the output.
- One level of Inverse discrete wavelet frame transform (IDWFT) is applied, producing the Y component of the frame. Resultant Y , C_b and C_r are having same dimensions.
- The Y , C_b and C_r components are gathered and used to reconstruct the frames in YC_bC_r format. Colored I frames are obtained and are utilized for further processing steps.

3.3 Edge enhancement for I frames

As we are losing high-pass horizontal and vertical subbands of Y component of Intra coded frames, edge detailing in those frame will be affected. Corresponding C_b and C_r band information are intact, though. The edge details in the chrominance and luminance bands are more or less same as the edges come under pixels from same area throughout the

color and grey components. So we can use an edge enhancement technique on those chrominance planes to improve the edge details. In the work [11], an edge enhancement method by gray polynomial interpolation (GPI) is proposed. We used the same method on the chrominance planes and achieved improvement in edge description. Basically, the edge enhancement consists of two stages: to intensify edges and to smooth neighborhood pixels of edges. In the first stage, the Laplacian filter is applied to intensify edges. In the second stage, the edges in the edge enhanced image are found through Canny edge detector. Then, pixels around the zigzag edges are smoothed by averaging.

The following steps are done on chrominance bands for edge enhancement:

- Edges in the chrominance planes are intensified (In this part, we are denoting both chrominance band by C for convenience) by Laplacian filter where the mask in (Eq. 1) is used. And the output image is denoted as C_L .

$$L_m = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (1)$$

- \tilde{C}_L is obtained by scaling C_L with a positive factor β , i.e., $\tilde{C}_L = \beta \times C_L$. Then add \tilde{C}_L to image C_L whose result is $C_E = \tilde{C}_L + C_L$. For our work, we used β as 0.25.
- The Canny edge detection is applied to locate edges of image C_E .
- For edge pixel $f(x, y)$, horizontal and vertical components, $H(x, y)$ and $V(x, y)$, respectively, are computed as:

$$H(x, y) = [f(x-1, y-1) - f(x-1, y+1)]/4 + [f(x, y-1) - f(x, y+1)]/2 + [f(x+1, y-1) - f(x+1, y+1)]/4 \quad (2)$$

$$V(x, y) = [f(x-1, y-1) - f(x+1, y+1)]/4 + [f(x-1, y) - f(x+1, y)]/2 + [f(x-1, y+1) - f(x+1, y+1)]/4 \quad (3)$$

- When $H(x, y) \geq V(x, y)$, the edge direction is horizontal. Then, neighborhood pixels are modified as

$$f(x, y-1) = [f(x, y-1) + f(x, y-2)]/2 \quad (4)$$

$$f(x, y+1) = [f(x, y+1) + f(x, y+2)]/2$$

On the other hand, the edge direction is vertical and the neighborhood pixels are modified as

$$f(x-1, y) = [f(x-1, y) + f(x-2, y)]/2 \quad (5)$$

$$f(x+1, y) = [f(x+1, y) + f(x+2, y)]/2$$

3.4 Color transfer

3.4.1 Motion vector acquisition

Motion vectors are required for proper transfer of color. The simplest method of acquiring motion vectors is by calculating them. Although simple, this method has many limitations because of the high computational complexity of motion vector estimation algorithms.

For any standard codec, motion vectors are also required for proper decoding of the video at the decoder. The encoder calculates the required motion vectors for compression/decompression of video. This motion vector information is embedded in the bit-stream generated from the encoder and is then transmitted to the decoder. The decoder on the other hand extracts these motion vectors from the incoming bit-stream and uses them for decompressing/decoding the video. Since the calculation of motion vectors for color transfer, as discussed earlier, is not a viable option, extraction of motion vector from the standard decoder itself is a feasible choice. The decoder on extracting the motion vector information stores it into variables/matrices. These variables/matrices are identified and their values are tapped. Now, as and when a frame is decoded, the motion vectors from these variables/matrices are stored outside the decoder (within the color transfer block). These motion vectors are then synchronized with the frame output of the decoder. After synchronization, these vectors are used for transferring the color to the video. This process saves a lot of complexity with only a slight increase in computational delay at the decoder end, compared to a standard decoder due to the color transfer algorithm. The color transfer algorithm being a simple one, this increase in complexity and computational delay is negligible.

3.4.2 Color transfer algorithm

In this section, we describe in detail the color transfer technique used to colorize video sequence. A simple color transfer scheme, based on the work by Jacob et al. [7], has been used in our work. Proposed scheme makes use of the motion vector information for proper transfer of color. At a time, color is transferred to a group of $n \times n$ pixels. This group is referred to as a ‘pixel group’. A neighborhood N of the pixel group is defined as a group of $m \times m$ surrounding pixel groups. A search space is required for finding the best match of the pixel group. For this purpose, a group of $s \times s$ pixels, in the source frame, is defined as the search space S . The default search space for a pixel group is assigned to be the corresponding (same row and column) $s \times s$ pixel group in the color source frame. For computational purposes, values of n , m and s were taken to

be 2, 3, 3, respectively. So the ‘pixel group’ size we use is 2×2 . The value of m is taken as 3 in our work, i.e., each 2×2 pixel group will be having eight 2×2 pixel groups in its neighborhood. This is similar to the 8-connectivity neighborhood where instead of a pixel we consider a 2×2 pixel group.

To transfer the colors, we define an error distance E using the L_2 matching criteria between the color transfer metric of the target pixel group and the pixel groups present in the neighborhood of the displaced search space in the source image. The color transfer metric used is an equally weighted sum of average pixel group luminance and average neighborhood luminance variance. An error distance between the target neighborhood N_t and the source neighborhood N_s was defined using the following equation:

$$E(N_t, N_s) = \sum_{x \in N_t, N_s} [G(x) - C(x)]^2, \tag{6}$$

where G is the color transfer metric of the gray scale target frame (we have considered luminance channel of the target frame as G) while C is the luminance channel of the source frame and x are the pixels in these neighborhoods. In the summation term, N represents the respective neighborhood; x in $G(x)$ belongs to the neighborhood N_t while x in $C(x)$ belongs to the neighborhood N_s .

For each pixel group in the target image, a best match in the source image is searched using this error distance. For proper transfer of color, the default search space of a pixel group is displaced in accordance with the motion vector for that pixel group. For H.264, motion vector is generally calculated for each macro block (which can be of 16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 , 4×4 pixels) and not a pixel group. Thus, the motion vector for a pixel group is assumed to be the same as the motion vector of the macro block in which it falls. Once the search space is displaced, a best match is looked for by minimizing the error distance (calculated in Eq. 6). The ‘best match’ is the group for which this error distance is minimum. When the ‘best match’ is found, chrominance values for the selected source pixel group are transferred to the target pixel group. The luminance values of the target pixel group are kept unchanged. This process is repeated until the whole frame is colorized. Once a frame is colorized, it is used as a reference frame for colorizing the next uncolored frame.

The process is repeated until all the frames are colorized. The entire algorithm of colorization can be summarized as:

1. If the frame encountered is colored, store the frame in the buffer and move to the next frame. Repeat until an uncolored frame is encountered.
2. Acquire the motion vectors corresponding to the uncolored frame from the decoder.

3. For each pixel group in the target image displace, the default search space in the source frame according to motion vector information acquired in step 2.
4. Find the pixel group in the displaced search space of the previous colored (source) frame which gives the minimum error distance for the closes texture match. The error distance is calculated using Eq. 6.
5. Transfer the chromatic values of this best match pixel group to the target pixel group, retaining the luminance values of the target pixel group.
6. Repeat step 2 to step 5 until all the pixel groups have been processed.
7. Repeat step 1 to step 6 until all the frames have been processed.

4 Results

The proposed encoder design was tested for various standard videos ranging from videos with slow motion (example: ‘Akiyo’, ‘Claire’, etc.) to videos with high motion content (example: ‘Football’, ‘Coastguard’, etc.). These videos are available here: <http://media.xiph.org/video/derf/>. Besides standard videos, video from motion picture (example: ‘Jurassic Park’) and video having multiple shots plus high color content (‘Wildlife’) were also used to prove the viability of the design in such situations. ‘Jurassic Park’ and ‘Wildlife’ sequences are higher resolution clips and other videos are either CIF sequences or QCIF sequences. Frame dimensions and total number of frames in individual sequences are mentioned in Table 1. Simulations were performed using MATLAB 7.6 and Visual C++ (version 9.0.2) under Microsoft Visual Studio 2008 on a 3.2 GHz quad core computer system with Intel Core i5 processor with 3 GB RAM and Windows 7 as operating system.

Our proposed encoder adds a pre-processing step before using standard codec and this method provides additional

compression over standard codecs. For the compression through standard codecs, if we employ the grayscale video, pre-processed by proposed algorithm, instead of the original color video sequences, coded video would be more compressed. Our simulation results also support this fact. Table 1 presents the extra compression achieved by our work over H.264/AVC baseline profile (H.264/AVC codec used in our work: <http://iphone.hhi.de/suehring/tml/download/>). We get a significant amount of extra compression over standard codec and our scheme outperforms over the schemes described in [4] and [6]. Substantial amount of compression is yielded even for videos with higher resolution (‘Jurassic Park’ and ‘Wildlife’). Also in the [14], a more complex texture-based video coding yielded compression around 15 %. Proposed method certainly provides more compression than the method described in [14].

Figures 3, 4, 5 show the frames of different video sequences colored using the proposed color transfer algorithm (Football, Wildlife, Jurassic park video sequences, respectively). Ample examples of corresponding original frames and decoded frames are shown for visual comparison. We have used videos having high motion content (Football), high color content (Wildlife), multiple shots (Wildlife, Jurassic park) and high resolution motion picture sequences (Jurassic park). Visually, the results are satisfactory. Original and decoded frames are very similar. The color positioning at or near the boundary and edges are likewise in original and decoded frames. So it is evident that proper color transfer occurs.

We have measured the quality of overall color recovery and decoding of the frames using two metrics. The first one is the standard peak signal to noise ratio (PSNR). We use an overall PSNR which considers PSNR of all the three channels of $YCbCr$ color space. The PSNR is a standard method of measuring the degradation in quality. However, sometimes it fails to capture the perceptual aspects of an image because it does not take account of human visual system (HVS). Since in any codec, video quality from a users’ visual perception is of prime importance, a quality estimation method that considered HVS is very crucial. For this purpose, an objective image quality measure named as WPSNR (Weber-based peak signal to noise ratio), defined by Ameer et al. [15], is used. Table 2 shows PSNR and WPSNR achieved by standard codecs after compression and decoding original color video sequences and PSNR, WPSNR attained by proposed codec processing on the video sequences using pre and post-processing over the corresponding standard H.264/AVC codec. We can see obtained PSNR and WPSNR values for standard codec and our proposed method are in same range. Naturally, to achieve extra compression over standard codec, we lose quality of video, but that loss is mostly negligible. The drop in PSNR and WPNSR is also not very significant. Also if we compare

Table 1 Advantages in terms of compression ratio using proposed encoder over H.264/AVC

Video sequence	No of frames	Increase of compression ratio over H-264 (%)
Akiyo (352 × 288)	300	22.56
Flower (352 × 288)	250	20.37
Foreman (352 × 288)	300	20.00
Football (352 × 288)	260	16.69
Claire (176 × 144)	494	21.60
Coastguard (352 × 288)	300	20.16
Wildlife (1280720)	920	19.22
Jurassic Park (1280720)	800	20.23

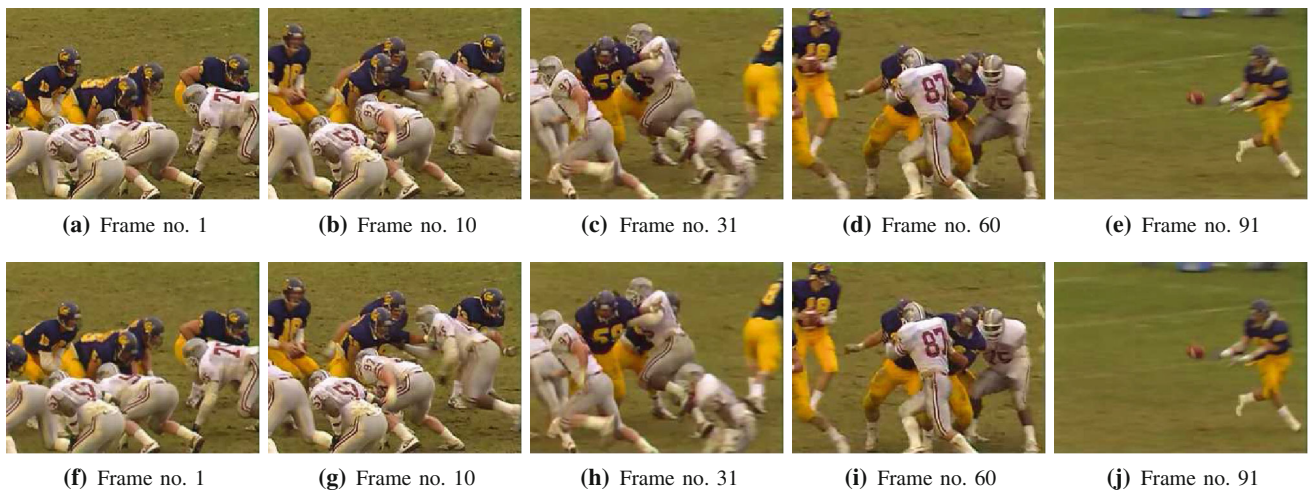


Fig. 3 Football video: *first row a–e* shows frames of original sequence; *second row f–j* shows corresponding decoded/colored frames

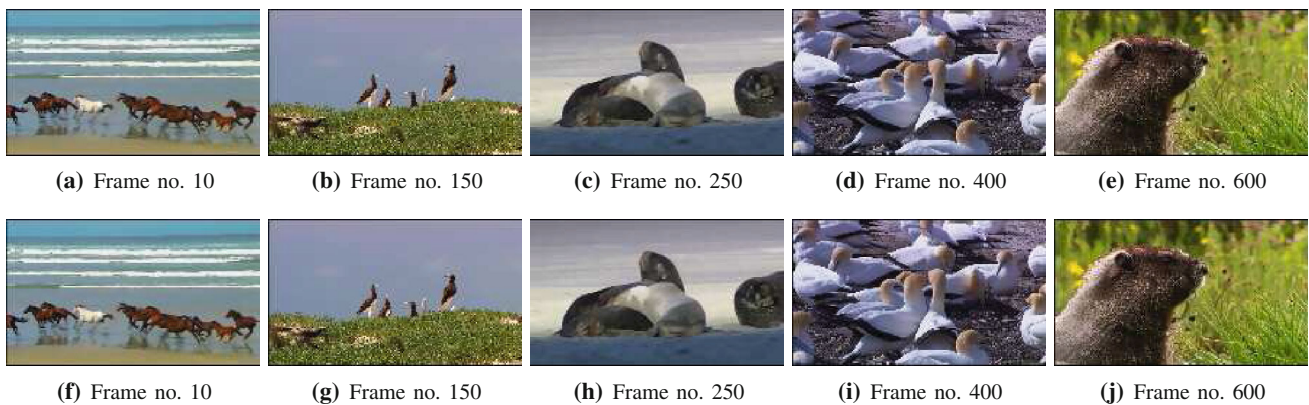


Fig. 4 Wildlife video: *first row a–e* shows frames of original sequence; *second row f–j* shows corresponding decoded/colored frames

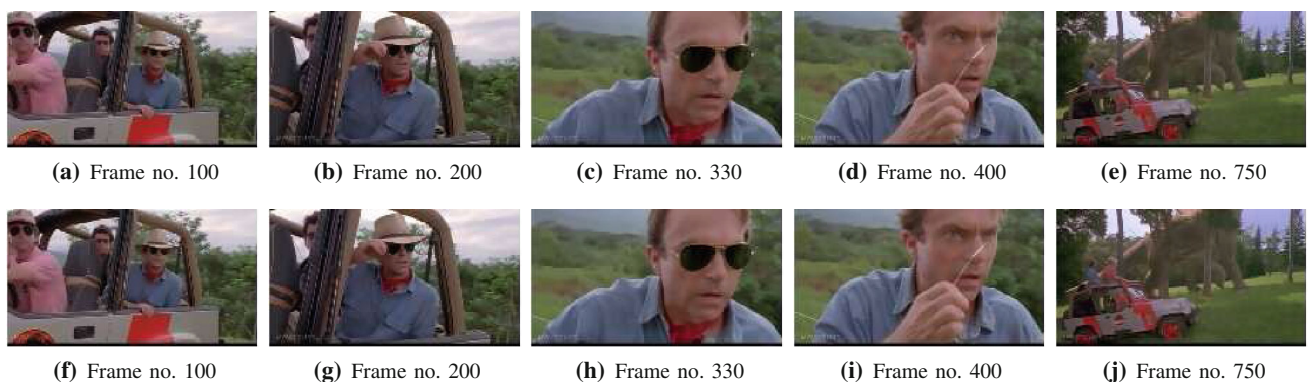


Fig. 5 Jurassic Park video: *first row a–e* shows frames of original sequence; *second row f–j* shows corresponding decoded/colored frames

PSNR and WPSNR values, our work shows certain advantage over the work proposed in [6]. Overall the results in terms of PSNR and WPSNR measurement are satisfactory.

We have used a color mapping to grayscale I frames scheme in encoder side. We have used discrete wavelet

frame transform (DWFT) instead of discrete wavelet transform(DWT). In the work [8], DWT was used for same kind of color mapping. DWT uses down-sampling and up-sampling which degrades the color of recovered image. On the other hand, DWFT does not use any down-sampling

Table 2 Performance analysis of proposed codec scheme over H.264/AVC baseline profile (QP =16) in terms of PSNR and WPSNR

Video sequence	H.264 (baseline profile)		Proposed decoder	
	PSNR (dB)	WPSNR (dB)	PSNR (dB)	WPSNR (dB)
Akiyo	45.22	57.15	43.39	53.77
Flower	44.53	57.87	42.65	55.00
Foreman	46.14	58.35	44.00	56.01
Football	44.58	55.41	42.48	53.16
Claire	48.25	58.51	46.00	55.88
Coastguard	47.97	58.00	45.88	55.55
Wildlife	45.65	56.67	43.97	54.25
Jurassic Park	46.66	57.31	44.87	55.20

Table 3 Improvement with DWFT-based color mapping scheme over DWT-based scheme: proposed method using H.264/AVC codec

Video sequence	Proposed Scheme using H.264	
	PSNR (dB) for DWT-based color mapping	PSNR (dB) for DWFT-based color mapping
Akiyo	41.70	43.39
Flower	40.00	42.65
Foreman	41.52	44.00
Football	40.17	42.48
Claire	43.57	46.00
Coastguard	43.21	45.88
Wildlife	41.49	43.97
Jurassic Park	41.73	44.87

Table 4 Comparison of decoding time between H.264/AVC and proposed codec scheme (For 100 frames)

Video sequence	Decoding time H.264 (in s)	Decoding time proposed codec (in s)	Increase of decoding time using proposed codec (%)
Akiyo	143	168	17.56
Flower	139	167	20.16
Foreman	129	152	16.19
Football	172	190	18.39
Claire	101	119	17.76
Coastguard	150	180	20.03
Wildlife	220	254	15.65
Jurassic Park	209	246	17.63

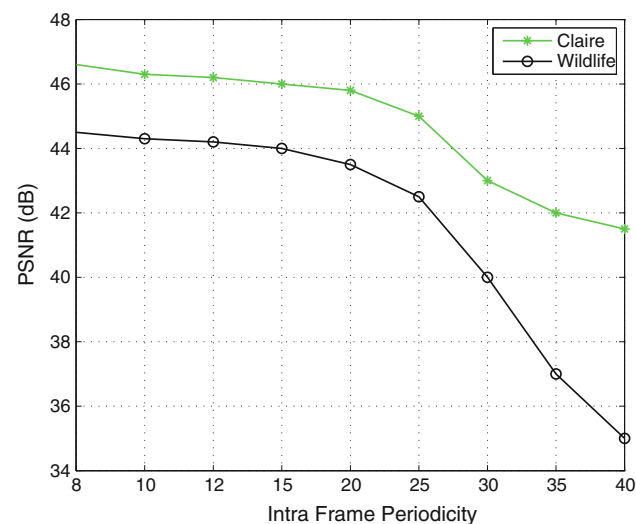


Fig. 6 Effect of Intra frame periodicity on PSNR (for proposed codec using H.264)

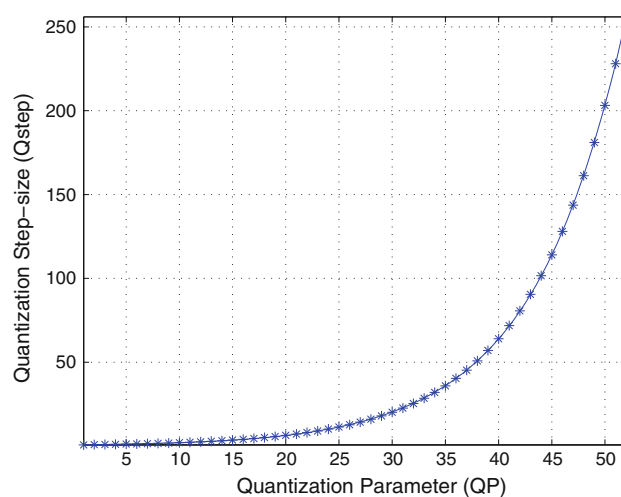


Fig. 7 Relation between QP and Qstep in H.264/AVC

and up-sampling. So, definitely DWFT-based color mapping scheme provides better color recovery. Table 3 also proves better performance of DWFT over DWT as a transform technique used in color embedding/mapping.

Colored I frames are used as the reference or source frames at the beginning of color transfer process. So their periodicity of occurrence has effects on the quality of color transfer. Figure 6 shows the changes of PSNR for proposed codec using H.264/AVC with I frame periodicity. We have considered two sequences: video with slow motion content (Claire) and video with moderate motion content and multiple shots (Wildlife). We can see with increasing I frame periodicity, PSNR value decreases. So longer gaps between I frame degrade the color transfer. For I frame periodicity <20, change of PSNR with varying periodicity of I frame is not severe. But periodicity more than 20 the variation becomes more and PSNR drops more in the case

Table 5 Relation between QP and Qstep in H.264/AVC

QP	0	1	2	3	4	5
Qstep	0.625	0.6875	0.8125	0.875	1	1.125
QP	6	7	8	9	10	12
Qstep	1.25	1.375	1.625	1.75	2	2.5
QP	16	18	22	24	28	30
Qstep	4	5	8	10	16	20
QP	34	36	40	42	48	52
Qstep	32	40	64	80	160	224

Table 6 Effect of different QP on compression ratio of proposed codec using H.264/AVC

Video sequence	Increase of compression ratio over H.264 using proposed encoder for different QP (%)				
	QP = 10	QP = 16	QP = 22	QP = 28	QP = 34
Akiyo	23.84	22.56	20.01	16.50	12.23
Flower	21.55	20.37	18.00	14.80	11.34
Foreman	21.04	20.00	17.56	14.03	11.01
Football	18.01	16.69	14.03	12.10	10.10
Claire	22.98	21.60	18.93	15.00	12.32
Wildlife	20.19	19.04	17.03	14.00	12.00
Jurassic Park	20.87	19.46	16.48	13.59	11.93

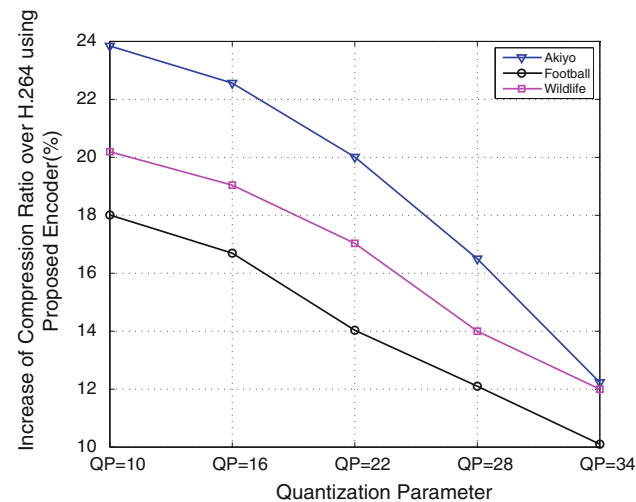


Fig. 8 Effect of QP on compression ratio of proposed codec using H.264/AVC

of video having moderate motion content and multiple shots (Wildlife).

Decoding time required in the post-processing is an important issue for codec performance analysis. A longer delay in post-processing limits the scope of possible application of codec. In Table 4, we have presented the

Table 7 Rate-distortion analysis on football (CIF) sequence. Effect of changing QP on bitrate and PSNR of proposed codec using H.264/AVC codec

QP	H.264 codec		Proposed codec	
	Bitrate (kbps)	PSNR (dB)	Bitrate (kbps)	PSNR (dB)
4	315.67	55.00	250.20	53.45
10	160.35	50.11	128.25	47.87
16	83.50	44.58	69.51	42.48
22	44.00	39.28	35.03	37.00
28	23.35	34.15	17.80	32.01
34	12.45	29.62	9.06	27.53
40	7.12	26.15	5.10	24.35

Table 8 Rate-distortion analysis on Akiyo (CIF) sequence. Effect of changing QP on bitrate and PSNR of proposed codec using H.264/AVC codec

QP	H.264 codec		Proposed codec	
	Bitrate (kbps)	PSNR (dB)	Bitrate (kbps)	PSNR (dB)
4	370.00	54.50	295.35	52.80
10	190.30	49.60	147.70	47.70
16	96.40	44.67	75.00	42.71
22	50.00	40.20	38.30	37.80
28	26.70	35.24	21.30	33.01
34	15.10	30.30	13.50	28.10
40	10.50	25.90	9.10	23.80

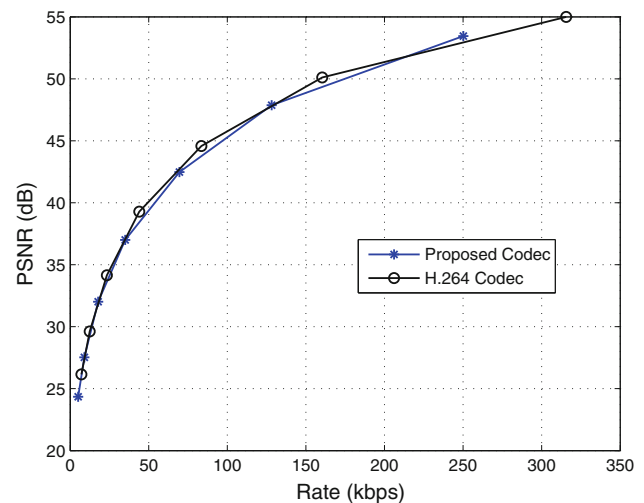


Fig. 9 Football (CIF) video: rate vs. PSNR plots

additional time required for the post-processing. We have used 100 frames of the video sequences in all the cases. Among the video sequences, Wildlife and Jurassic park are high resolution sequence (frame size: 1280720) while rest are CIF sequence (352 × 288). For all the sequences, additional time required for post-processing is around or

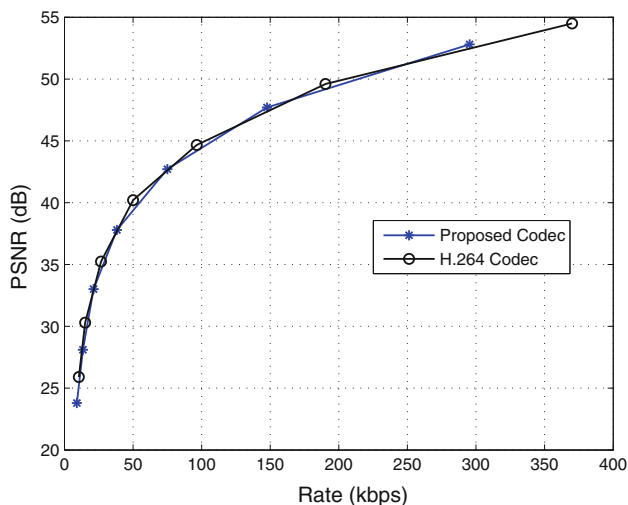


Fig. 10 Akiyo (CIF) video: rate vs PSNR plots

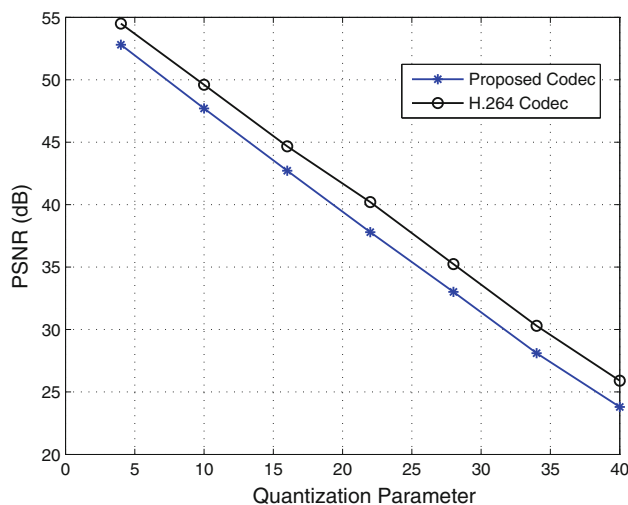


Fig. 12 Akiyo (CIF) video: quantization parameter vs. PSNR

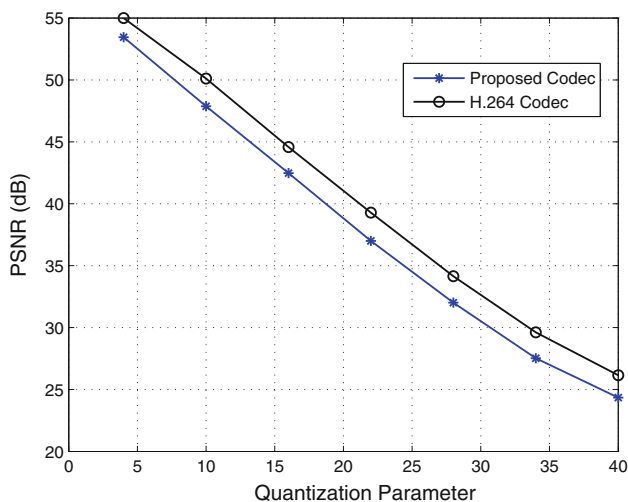


Fig. 11 Football (CIF) video: quantization parameter vs. PSNR

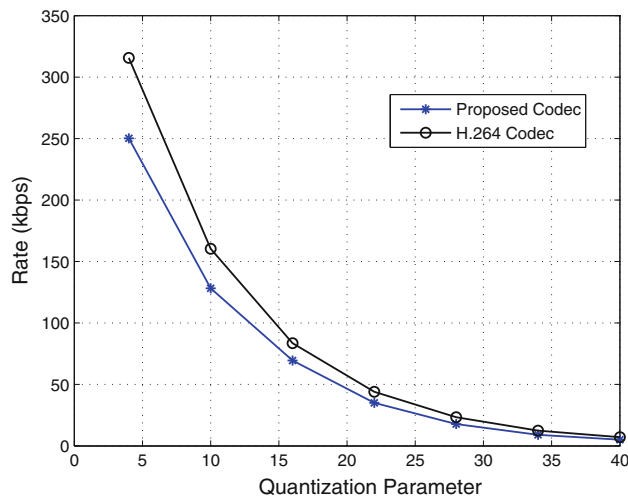


Fig. 13 Football (CIF) video: quantization parameter vs. Bitrate

below 20 % of the total time required for decoding the original color video by standard codecs. Obtained encoding time is far less than that obtained by the work described in [4, 6]. However, we have used mainly visual C++ and MATLAB code for decoder algorithm design. Performance can be bettered using dedicated software.

Quantization in an H.264/AVC encoder is controlled by a quantization parameter, QP. Quantization parameter denotes the quantization scale indirectly, whereas the quantization step-size is the true value used in quantization [16]. Quantization step-size (Qstep) or quantization parameter(QP) has direct effect on compression, quality of encoded video as well as transmission rate in case of video communication. QP gives the measure of rate-control and rate-distortion analysis [17]. Increase of quantization parameter (QP) or quantization step (Qstep) means loss in video quality, more compression and less required bit-rate.

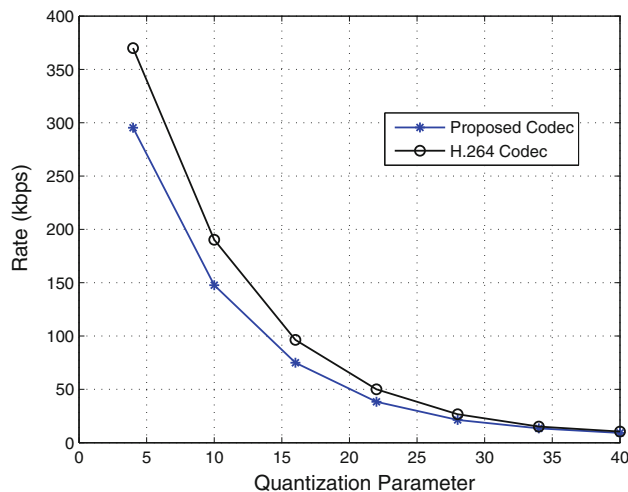
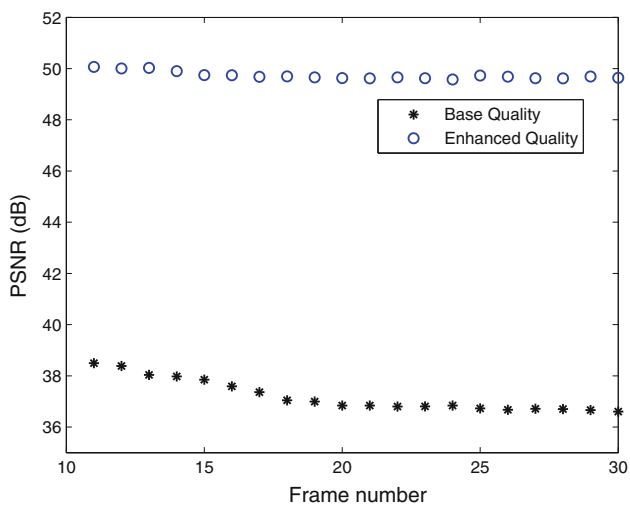
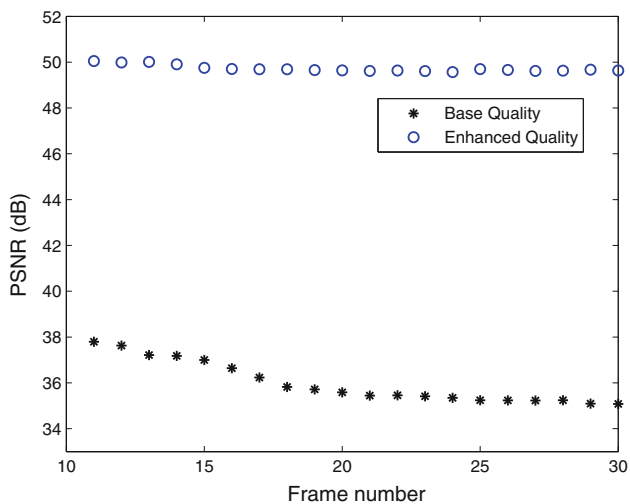


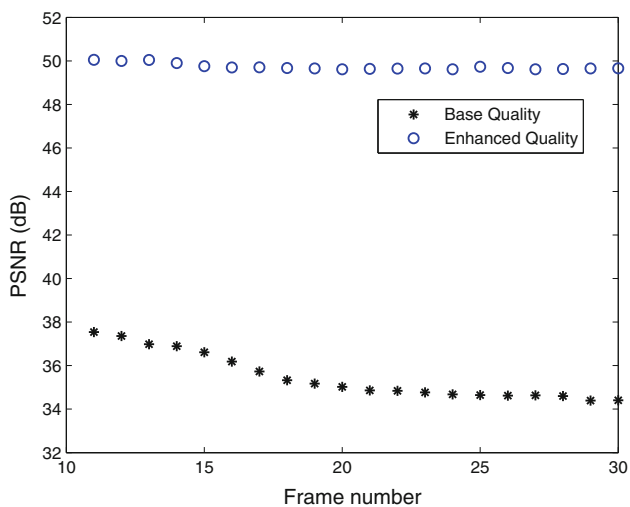
Fig. 14 Akiyo (CIF) video: quantization parameter vs. Bitrate



(a) Qstep = 2; QP = 10



(b) Qstep = 3; QP = 16



(c) Qstep = 4; QP = 22

◀ **Fig. 15** SNR/Quality scalability for different QP: Akiyo video sequence

In the previous video coding standards, the relation between quantization parameter and quantization step size is usually linear. However, in H.264/AVC, the relation between QP and Qstep is that $Qstep = 2^{(QP - 4)/6}$ [17]. The relation between Qstep and QP is shown in Fig. 7. A total 52 values of Qstep and QP is supported by the standard (shown in Table 5). Qstep doubles in size for every increment of 6 in QP.

Table 6 and Fig. 8 exhibit the effect of change of quantization parameter (QP) on achieved extra compression over standard H.264/AVC baseline profile codec. It can be seen with increasing QP, advantage obtained in our work in terms of archived additional compression is getting reduced. Because high QP means higher quantization steps and higher compression provided by standard codec itself. But this higher compression comes at the cost of degraded quality. For lower QP and higher quality output, our encoder offers more compression. Thus, the proposed scheme is capable of achieving high compression at high quality (low QP). At a lower quality (high QP), the compression achieved by the standard encoder is itself high.

5 Rate-distortion analysis of proposed codec

Rate-distortion analysis is another crucial aspect for a codec. In this part, we have discussed this analysis of our proposed codec which uses H.264/AVC baseline profile standard as a part of it. Rate distortion optimization is always a checkpoint for a codec for its performance requirement. For a high quality of coded output, distortion will be less so PSNR will be high but also compression would be less which means a higher required bitrate for video transmission over network. For a lower available bitrate, a higher compression will be applied that will increase the distortion and hamper the quality of output video. So a trade-off must be maintained between desired quality and maximum bitrate required for transmission over networks. The most vital to control the quality and compression is the quantization parameter (QP). Unlike previous standard codecs, H.264 does not maintain a linear relationship between QP and Qstep. Details about this are discussed in the last section. Tables 7 and 8 present the statistics in details of change in PSNR and bitrate with varying QP for Football and Akiyo video sequences, respectively. Both the sequences are in CIF format. We are considering PSNR as the measure of distortion. With the raising PSNR, distortion will be decreased and vice versa.

From the tables, we can see bitrate decreases with increasing QP. Increasing QP means increasing quantization size and increase in compression size. So output will be more compressed. So bitrate will be lower. Higher compression comes with the cost of degraded quality. So PSNR will drop with increasing QP, which can be noticed from these tables. Figures 9 and 10 show PSNR versus rate plots for the mentioned sequences. We can see for same bitrate proposed codec and standard H.264/AVC baseline codec maintain almost same PSNR. Figures 11 and 12 indicate the almost liner relationship between QP and PSNR. For same QP proposed codec shows slight drop in PSNR in comparison with H.264/AVC baseline codec. This is because of little data loss to achieve extra compression over standard codec (discussed in last section). Figures 13 and 14 show the drop of bitrate for increasing QP. As proposed codec gains extra compression over standard codec, bitrate is less for proposed codec in comparison with H.264/AVC baseline codec for same quantization parameter. It is evident that rate-distortion and other related performances for proposed codec is similar to standard H.264/AVC codec. Proposed codec is advantageous in terms of bitrate but poses slight loss in quality. For same bitrate, quality or distortion measure is about same for both codec, proposed one and H.264/AVC.

6 Scalability of proposed codec

Scalability refers to the capability of recovering physically meaningful image or video information from decoding only partially compressed bitstreams [18]. Scalable video coding (SVC) has been added with the H.264/AVC standard recently. We have checked the compatibility and performance of our proposed codec as a scalable video codec. We have employed scalable baseline profile of H.264/AVC as the part of standard codec in proposed scheme. We have checked the spatial and SNR scalability performance of the proposed scalable codec. Under spatial scalability video is coded at multiple spatial resolutions. The data and decoded samples of lower resolutions can be used to predict data or samples of higher resolutions to reduce the bit rate to code the higher resolutions. For SNR/quality/fidelity scalability, video is coded at a single spatial resolution but at different qualities. The data and decoded samples of lower qualities can be used to predict data or samples of higher qualities to reduce the bit rate to code the higher qualities [19]. We have used Akiyo sequence for showing the results. Results of SNR and spatial scalability for different QP can be seen in Figs. 15 and 16.

Figure 15 shows the result for SNR Scalability. In the figure, we have shown PSNR of 20 consecutive frames of

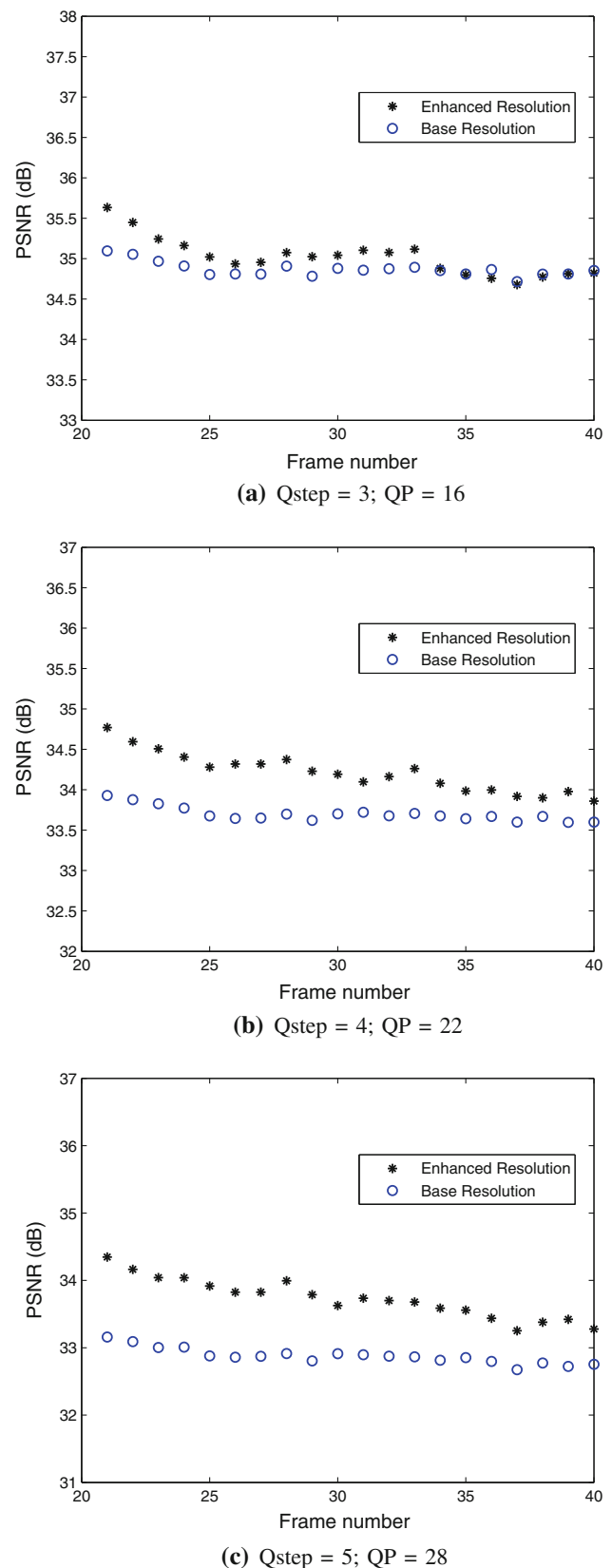


Fig. 16 Spatial scalability for different QP: Akiyo video sequence

Akiyo sequence. By decoding only the base layer, a lower quality image is obtained. By decoding the base layer and an additional enhancement layer, we get the same spatial resolution picture but with a better quality which poses a higher PSNR (in our results near 50 dB). For different quantization parameters, enhancement layer of same PSNR is produced.

Figure 16 shows the result for spatial scalability. Again we have shown PSNR of 20 consecutive frames of Akiyo sequence. The base layer carries a compressed lower spatial resolution video (In this work, 172×144) and the enhancement layers carry incremental data (frame size: 352×144). When only the base layer bit stream is decoded, one obtains a base resolution video. When the base layer and enhancement layers are together decoded, the highest resolution video is obtained. However, the quality of both the base and enhanced resolution videos may be about the same in lower quantization parameter because the intent is to distribute the same video material in a single bit stream which carries information about different resolutions of the videos. For higher quantization parameter, the quality of the originally decoded video degrades, so base layer PSNR also decreases.

7 Conclusion and future scope

In this paper, we have proposed a scheme for color video compression. The proposed scheme produces visually good results. The Proposed scheme is computationally very close to a standard decoder but provides much better compression than the standard codec. One of the most important features of this algorithm is that it transmits a video as a sequence of gray scale images, yet we obtain a colored output images of quality similar to the original video. The reuse of the motion vectors present in the standard decoder makes it simple in terms of computational burden. The degradation in quality, as verified by the standard PSNR as well as the perceptual measures, is also within acceptable limits. Thus, the proposed scheme can find applications in variety of arenas like Mobile video and other handheld devices, Internet TV, Video Streaming, etc. Work is now in progress to implement the proposed coding scheme using the most recent video coding standard HEVC. The new standard described in [20, 21, 22] has several new features and can achieve significantly more compression compared to standard H.264/AVC.

References

- Liu L, Zhu F, Bosch M, Delp E (2007) Recent advances in video compression: What's next?. In: Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on, pp 1–8
- Reinhard E, Adhikhmin M, Gooch B, Shirley P (2001) Color transfer between images. *Comp Graph Appl IEEE* 21(5):34–41
- Levin A, Lischinski D, Weiss Y (2004) Colorization using optimization. In: ACM SIGGRAPH 2004 Papers, SIGGRAPH '04, pp 689–694
- Kumar R, Mitra SK (2008) Motion estimation based color transfer and its application to color video compression. *Pattern Anal Appl* 11(2):131–139
- Rao KMS, Mitra SK (2004) A study on the application of color transfer technique for video compression. In: ICVGIP 2004, Proceedings of the Fourth Indian Conference on Computer Vision, Graphics and Image Processing, Dec 16–18, 2004, pp 628–631
- Agarwal R, Gupta S, Gude V (2011) A color video compression technique using key frames and a low complexity color transfer. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 7882 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series
- Jacob V, Gupta S (2009) Colorization of grayscale images and videos using a semiautomatic approach. In: Image Processing (ICIP), 2009 16th IEEE International Conference on, pp 1653–1656
- de Queiroz R, Braun K (2006) Color to gray and back: color embedding into textured gray images. *Image Proc IEEE Trans* 15:1464–1470
- Yang S, Lin H, Zhang A, Xu L (2011) A color image fusion method using discrete wavelet frames. In: Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on, vol. 6. pp 2807–281
- Li S, Kwok JT, Wang Y (2002) Using the discrete wavelet frame transform to merge landsat tm and spot panchromatic images. *Inform Fusion* 3(1):17–23
- Hsieh CH, Hung WL, Lan CW (2008) Grey image enlargement with edge enhancement. In: Machine Learning and Cybernetics, 2008 International Conference on, vol. 6. pp 3272–3276
- Welsh T, Ashikhmin M, Mueller K (2002) Transferring color to greyscale images. *ACM Trans Graph* 21:277–280
- H.264/AVC (2008) JM Reference Software, software version: JM 18.3
- Zhu F, Ng KK, Abdollahian G, Delp EJ (2007) Spatial and temporal models for texture-based video coding. In: Society of Photo-Optical instrumentation engineers (SPIE) Conference Series, vol. 6508 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series
- Ameer S, Basir O (2008) Objective image quality measure based on weber-weighted mean absolute error. In: Signal Processing, 2008. ICSP 2008. 9th International Conference on, pp 728–732
- Ma S, Gao W, Lu Y (2005) Rate-distortion analysis for h.264/avc video coding and its application to rate control. *Circuits Syst Video Technol IEEE Trans* 15:1533–1544
- Kwon DK, Shen MY, Kuo CC (2007) Rate control for h.264 video with enhanced rate and distortion models. *Circuits Syst Video Technol IEEE Trans* 17:517–529
- Wang Y, Ostermann J (2007) Video processing and communications. Pearson Education, London
- Schwarz H, Marpe D, Wiegand T (2007) Overview of the scalable video coding extension of the h.264/avc standard. *Circuits Syst Video Technol IEEE Trans* 17:1103–1120
- Zhang Y, Yan C, Dai F, Ma Y (2012) Efficient parallel framework for h.264/avc deblocking filter on many-core platform. *Multimed IEEE Trans* 14:510–524

21. Sullivan G, Ohm J, Han W-J, Wiegand T (2012) Overview of the high efficiency video coding (hevc) standard. *Circuits Syst Video Technol IEEE Trans* 22(12):1649–1668
22. Yan C, Zhang Y, Dai F, Li L (2013) Highly parallel framework for hevc motion estimation on many-core platform. In: *Data Compression Conference (DCC)*, 2013, pp 63–72