Scholars Journal of Engineering and Technology (SJET) **ISSN 2321-435X (Online)**
Sch. J. Eng. Tech., 2014; 2(4C):613-620 **ISSN 2347-9523 (Print)**
©Scholars Academic and Scientific Publisher
(An International Publisher for Academic and Scientific Resources)
www.saspublisher.com

Research Article

A Video Compression Technique Based On Active Learning Approach

Shireen Fathima*[1], Mohammed Azharuddin Ahmed[2]
[1]PG-Student, HKBK College of engineering, Bangalore, India
[2]Technical trainer, RIIIT, Mysore, India

***Corresponding author**
Shireen Fathima
Email: shireen.fathima6@gmail.com

Abstract: Many Video compression algorithms manipulate video frames to dramatically reduce the storage requirements and bandwidth required for transmission while maximizing perceived video quality. Typical video compression methods first transform the video frames from its spatial domain representation to frequency domain representation using some transform technique such as Discrete Cosine Transform vector quantization, fractal compression, and Discrete Wavelet Transform and then code the transformed values. Recently, instead of performing a frequency transformation, machine learning based approach has been proposed which has two fundamental steps: selecting the most representative pixels and colorization. Our proposed method converts the color video frames to gray scale frames and the color information for only a few representative pixels is stored. At the decoder side is all the color values for the gray scale pixels across frames is predicted. Selecting the most representative pixels is essentially an active learning problem, while colorization is a semi-supervised learning problem. In this paper, we propose a novel active learning method for automatically extracting the RP is proposed for video compression. In this paper the active learning problem is formulated into an RP minimization problem resulting in the optimal RP set in the sense that it minimizes the error between the original and the reconstructed color frame.

Keywords: video compression; active learning; semi-supervised learning; representative pixels.

INTRODUCTION

With the evolution of internet and WWW, there was a need to transmit images, videos and other multimedia objects over the network and for this various compression techniques were proposed to achieve better throughput. Some of these techniques focused on high compression ratio while others on better quality and appreciable compression ratio. Video compression is a crucial technique for reducing the bandwidth required to transmit videos. Video data contains spatial and temporal redundancy. Similarities can thus be encoded by merely registering differences within a frame (spatial), and between frames (temporal).Recently, machine learning based approach has been proposed for video compression [1-2]. Instead of performing a frequency transformation, Cheng et al. proposed to convert the color video to a gray scale video. A few representative pixels are selected whose color information is stored. The gray scale video and the selected color pixels are used to learn a statistical model to predict the color values for the rest of the pixels[3-5]. Their empirical result has shown that good compression ratio can be achieved while the image quality is reasonably good according to Peak Signal to Noise Ratio (PSNR) score. From a machine learning perspective, there are two fundamental problems. First, how to select the most representative pixels, which is essentially an active learning problem. The selected pixels, together with the gray scale video, are stored as the encoding process. Second, how to combine color and gray scale information of the pixels to learn a model, which is essentially a semi-supervised learning problem[6]. The learned model is used to recover the color video as the decoding process. In this paper, we propose to new active and semi-supervised learning for video compression which does not requires iteration.

RELATED WORK

Cheng et al. developed a straightforward active learning algorithm which iteratively selects the pixels on which the prediction errors are the highest. The major disadvantage of Cheng's approach is that there is no theoretical guarantee that the predication error can actually be reduced by using the selected pixels[1].

LapRLS algorithm:

The use of LapRLS for video compression is based on the assumption that if two pixels have similar intensity values and are spatially close to each other then it is very likely that they have similar color values[7]. Consider **z** to denote the labeled point, and **x** to denote any point (either labeled or unlabeled).

Consider a linear regression model

$$y = w^T \mathbf{x} + \varepsilon \text{ …… (1)}$$

Where y is the *observation*, $\mathbf{x} \in R^n$ is the *independent variable*, **w** is the *weight vector* and ε is an unknown error with zero mean. Different observations have errors that are independent, but with equal variancesσ^2. We define $f(\mathbf{x}) = w^T\mathbf{x}$ to be the learner's output given input **x** and the weight vector **w**. The LapRLS algorithm makes use of both labeled and unlabeled points to learn a regression model *f*. It assumes that if two points x_i , x_j are sufficiently close to each other, then their measurements (f (x_i) , f(x_j)) are close as well. Suppose there are totally *m* points out of which *k* points are labeled. Let *S* be a similarity matrix. Thus, the LapRLS algorithm solves the following optimization problem:

$$JLapRLS(\mathbf{w}) = \sum_{i=1}^{k} (f(z_i) - y_i)^2 + \frac{\lambda}{2} \sum_{i,j=1}^{m} \left(f(x_i) - f(x_j)\right)^2 S_{ij} + \lambda_2 \, ||W||^2 \text{….(2)}$$

Where y_i is the measurement (or, label) of z_i . The loss function with our choice of symmetric weights S_{ij} ($S_{ij}=S_{ji}$) incurs a heavy penalty if neighboring points x_i and x_j are mapped far apart. Therefore, minimizing *JLapRLS*(**w**) is an attempt to ensure that if x_i and x_j are close then $f(x_i)$ and $f(x_j)$ are close as well. There are many choices of the similarity matrix *S*.

PROPOSED METHOD

From machine learning perspective, video compression can be considered as a semi-supervised learning problem [5]. Given a set of color pixels (labeled examples) and a set of grayscale pixels (unlabeled examples) of a frame, we have to learn a function which will predict color (labels) on the grayscale pixels of the current and next several frames. Figure 1 shows the overview of the learning method.

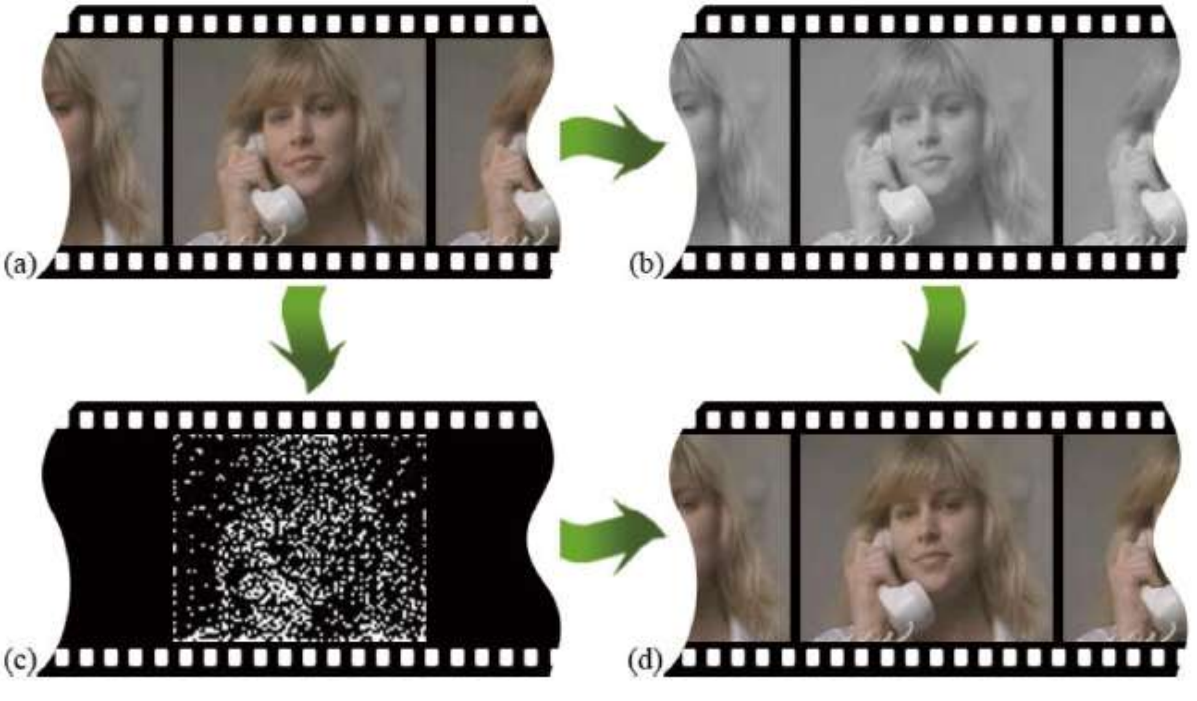


Fig-1: An illustrative example of learning based approach for video compression.

(a) Original color frames; (b) Grayscale frames; (c) Selected pixels with color information; (d) Recovered color frames.

Video is a sequence of still images called frames and hence the same image compression techniques can be used to compress the video frames. In this paper we extend our previous work [8] to video compression. The proposed method extracts minimal number of RPs using minimization problem which efficiently predicts the color information for other pixels in a frame resulting in a good quality compressed frame. Also the RP extraction is performed in a single step and does not require any iteration. Figure 2 shows the system diagram of our proposed(1) work.

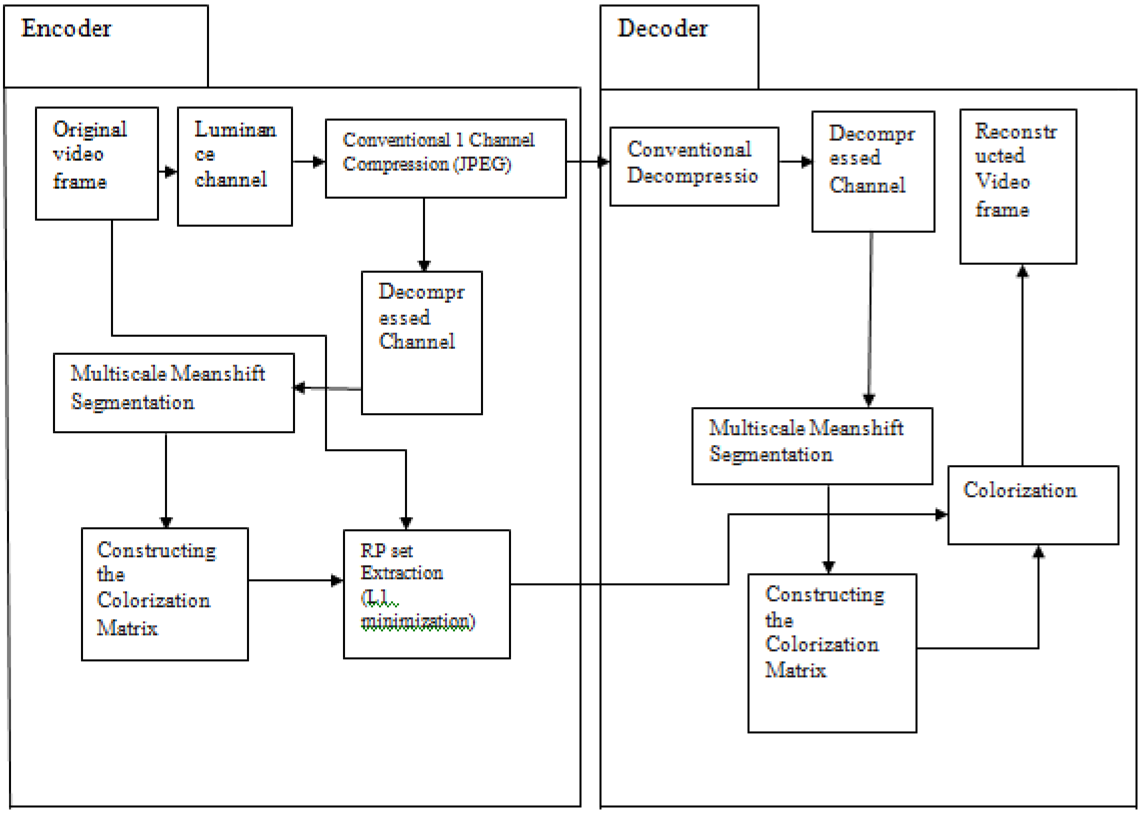

Encoder
Decoder
Original video frame
Luminance channel
Conventional 1 Channel Compression (JPEG)
Decompressed Channel
Multiscale Meanshift Segmentation
Constructing the Colorization Matrix
RP set Extraction (L1 minimization)
Conventional Decompressio
Decompressed Channel
Reconstructed Video frame
Multiscale Meanshift Segmentation
Colorization
Constructing the Colorization Matrix

Fig-2: Overview Of The Proposed(1) Method.

The proposed(1) method consists of the encoding and decoding process. In the encoder, the original color video frame is first decomposed into its luminance channel and its chrominance channels. The luminance channel is compressed using conventional one-channel compression techniques, e.g., JPEG standard, and its discrete Fourier or Wavelet coefficients are sent to the decoder. Then, in the encoder, the color matrix C is constructed by performing multi-scale meanshift segmentation on the decompressed luminance channel. The decompressed luminance channel is used to consist with that in the decoder. Using this matrix C and the original chrominance values obtained from the original color video frame, the RP set is extracted by solving a minimization problem. This RP set is sent to the decoder, where the colorization matrix C is also reconstructed from the decompressed luminance channel. Then, by performing a colorization using the matrix C and the RP set, the color video frame is reconstructed. The method is carried out for all the frames of the video. Figure 5.1 shows the overall system diagram of the proposed(1) method.

The proposed(1) is modified little in that instead of compressing the luminance channel by jpeg, the luminance channel of each video frame is compressed by jpeg-2000 standard at the encoder. This modified work is called proposed(2).

RESULTS AND DISCUSSION

The proposed technique has been tested with several different video sequences. A video is taken and it is compressed by proposed method. The peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) value are used as an objective evaluation of image quality. PSNR is defined as

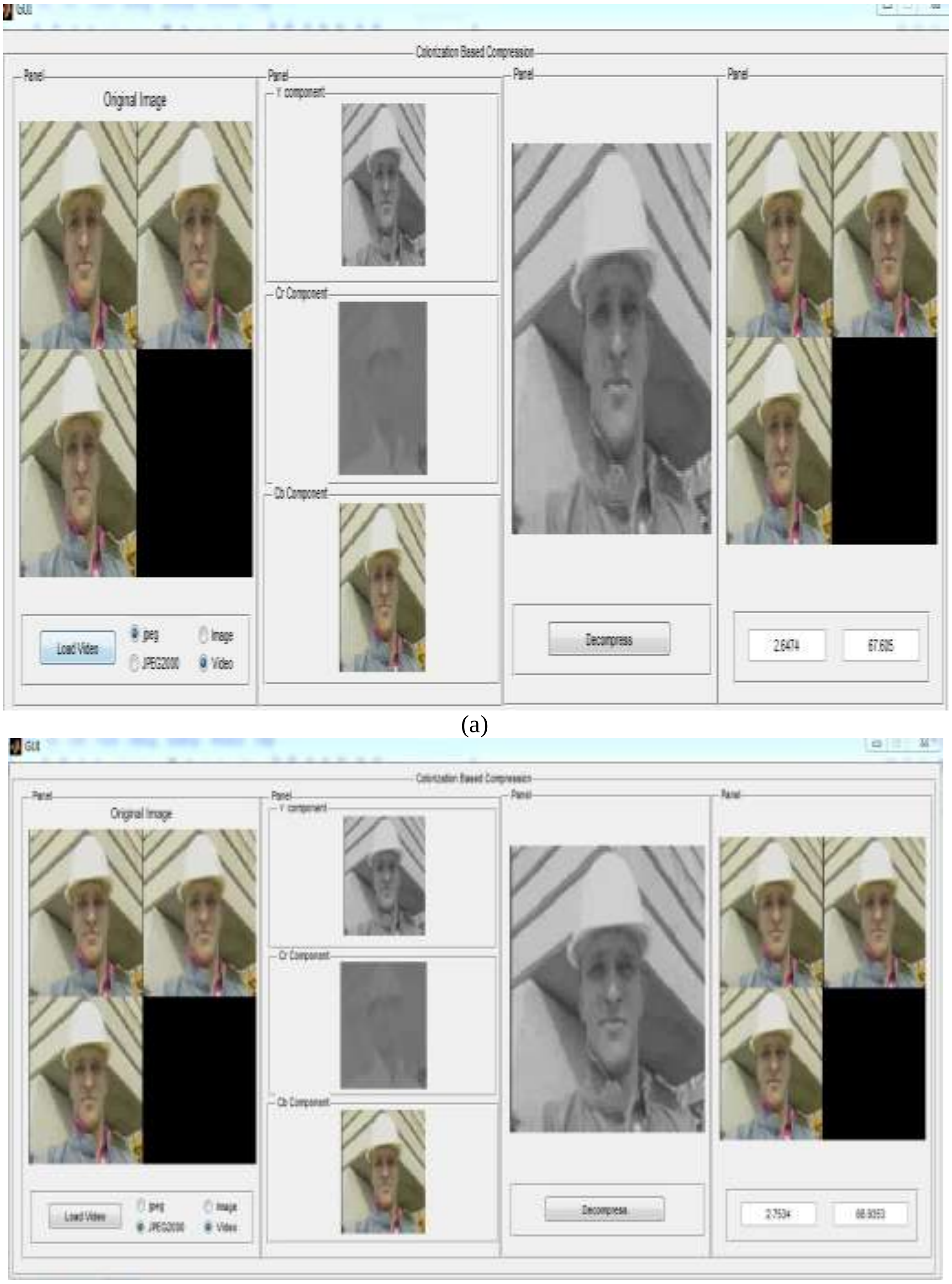
$$\text{PSNR[dB]} = 10\ \log_{10}\left[\frac{255^2}{MSE}\right]$$

Where MSE is a mean squared error. SSIM is the image quality assessment based on the degradation of structural information, better for the human visual estimation than traditional image quality assessments such as PSNR. SSIM between images X and Y is defined as

$$\text{SSIM} = \frac{(2\mu_x\mu_y+C_1)+(2\,\sigma_{xy}+C_2)}{(\mu_x{}^2+\mu_y{}^2+C_1)(\sigma_x{}^2+\sigma_y{}^2+C_2)}$$

Where μ_x is the average of X and μ_y is the average of Y. σ_{xy} is the covariance of X and Y. σ_x is the variance of X and σ_y is the variance of Y. $C1$ and $C2$ are constants. Result numbers used for PSNR and SSIM are averages of PSNR and SSIM of the three RGB components.

Video is a sequence of images and the same still image compression technique can be applied to each frame. The video compression is implemented in two ways first by compressing luminance channel of each frame by jpeg compression standard. Second by compressing it with Jpeg-2000.Figure 3(a) shows a foreman video compression for 3 frames using proposed1 and 3(b) shows the compression of the same video using proposed2.



● jpeg ○ Image
○ JPEG2000 ● Video

2.6474 67.605

(a)

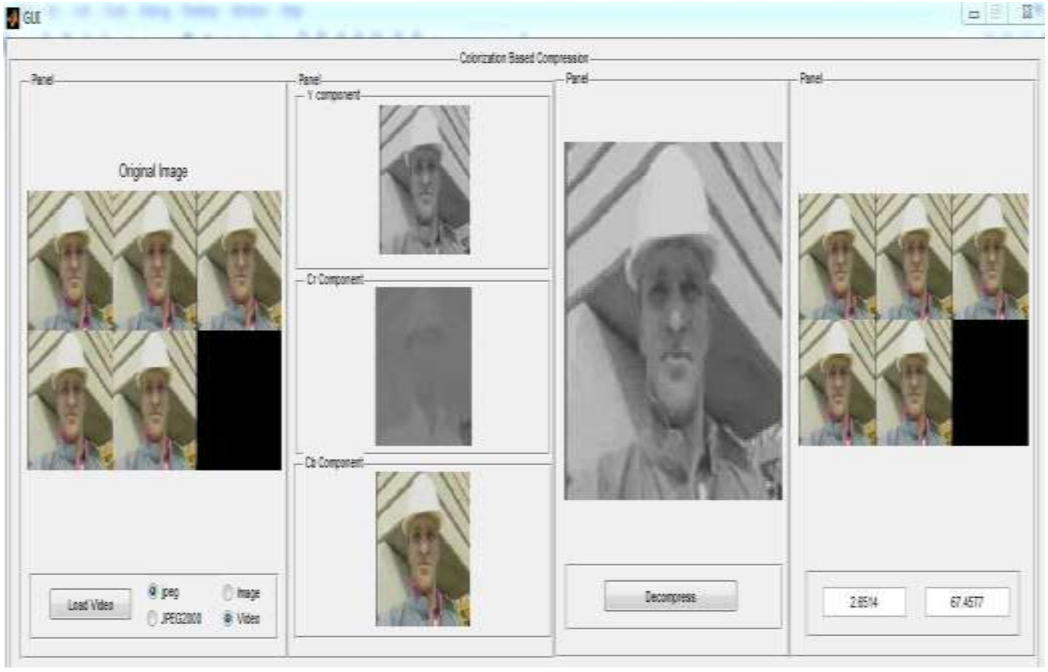
● jpeg ○ Image
● JPEG2000 ● Video

2.7534 68.9353

(b)

Fig-3: video compression for 3 frames. (a):Proposed1 (b) :Proposed2

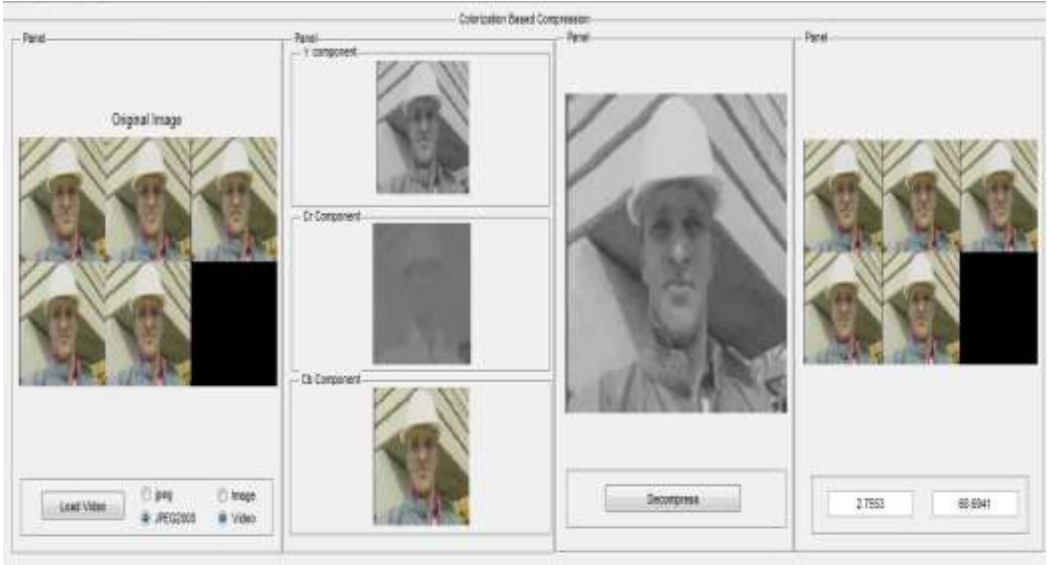
Figure 4 and 5 shows the video compression for 5 and 15 frames respectively using both the proposed methods. Figure 6 shows the results of another video compression for 30 frames using both proposed methods.



● jpeg ○ Image
○ JPEG2000 ● Video

2.8514 67.4577

(a) Proposed(1)



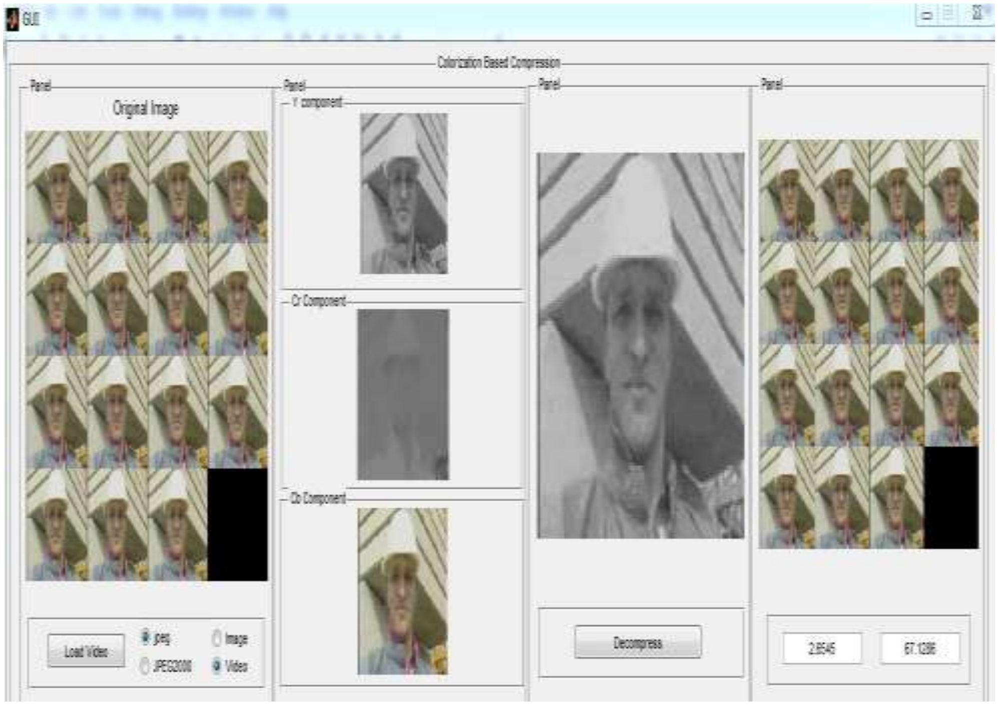
○ jpeg ○ Image
● JPEG2000 ● Video

2.7553 66.6941

(b) Proposed(2)

Fig-4 : Video compression for 5 frames

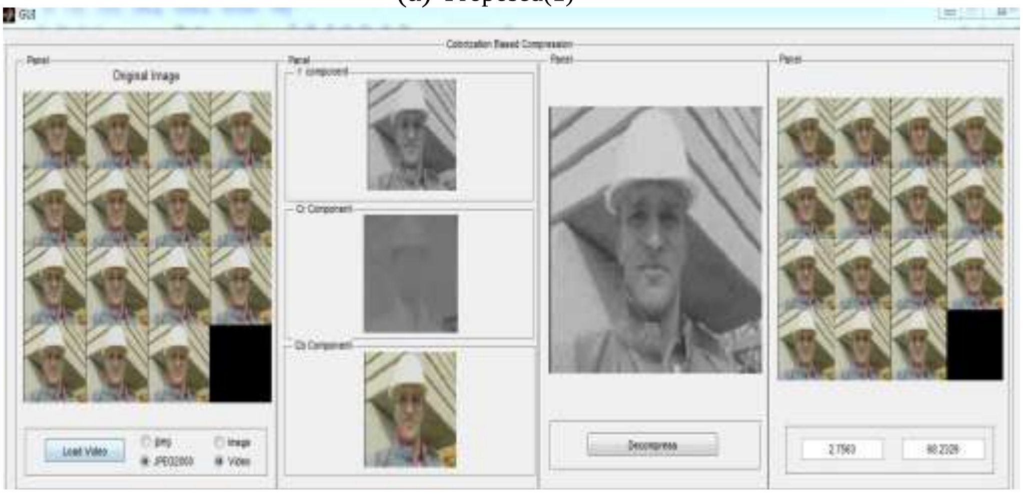
The Left side of the GUI shows the video frames and the individual frame decomposed into YCbCr components. The GUI is designed such that each frame decomposition is shown in a sequence. The right side of the GUI shows the reconstructed Y component and the final reconstructed frames along with the SSIM and PSNR values.



● jpeg ○ JPEG2000 ○ Image ● Video

2.8545 67.1286

(a) Proposed(1)

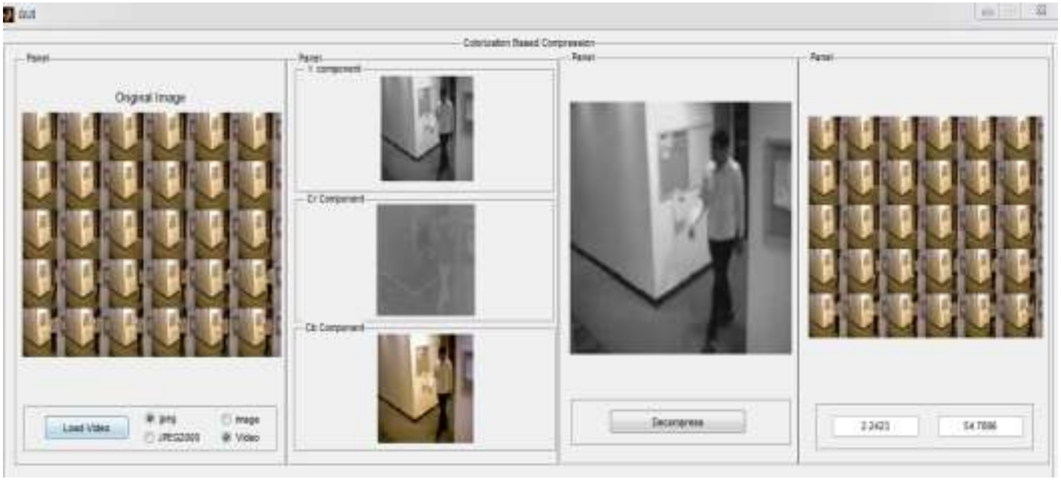


○ jpeg ● JPEG2000 ○ Image ● Video

2.7563 60.2329

(b) Proposed(2)

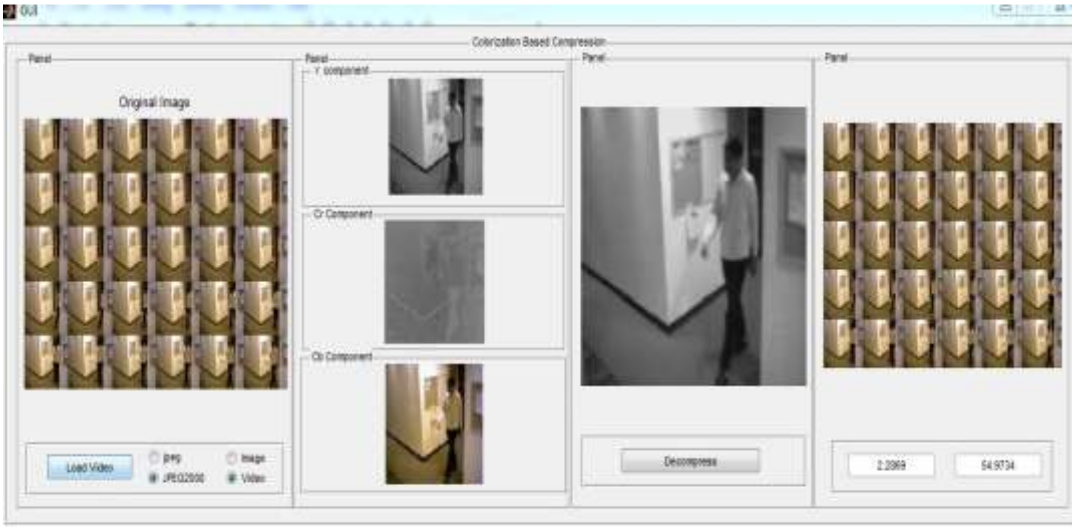
Fig-5: Video compression for 15 frames.



● jpeg ○ JPEG2000 ○ Image ● Video

2.2423 54.7886

(a) Proposed (1)



○ JPEG ● JPEG2000 ○ Image ● Video

2.2969 54.9734

(b) Proposed(2)

Fig-6: video compression for 30 frames

The PSNR and the SSIM values are also calculated for the video compression. The elapsed time for simulation is also calculated. The GUI is designed to include the average PSNR and average SSIM for the video. Table-1 shows the PSNR and SSIM for the foreman video using the proposed methods.

Table -1: PSNR and SSIM values for the Foreman video compression.

No. of frames	Proposed (1)		Proposed(2)	
	SSIM	PSNR	SSIM	PSNR
3	2.6474	67.605	2.7534	68.9353
5	2.6514	67.4577	2.7553	68.6941
15	2.6545	67.1286	2.7563	68.2329

The individual PSNR and SSIM values for each frame of the video are also calculated which are shown in the matlab window. Video compression is a complex technique involving complex motion estimation algorithms to extract and compress redundancy across frames. In this paper the similarities between frames is not considered for compression. This approach for video compression can be extended to extract the redundancy between frames to achieve better compression in future.

CONCLUSION AND FUTURE WORK

Colorization-based coding extracts redundant representative pixels from each frame and do not extract the pixels required for suppressing coding error. Our paper demonstrates automatic extracting of representative pixels in a single step and do not require iterations.

In future it would be more interesting to investigate how one can improve the proposed video compression technique by considering the similarities between the adjacent frames for compression and enhancing the compression rate and quality of the compressed video. The performance evaluation of the proposed method can be carried out by comparing it with other video compression standards in future.

REFERENCES

1. Cheng L, Vishwanathan SVN; Learning to compress images and videos. Proc. Int. Conf. Mach. Learn, 2007; 227:161–168.
2. He X, Ji M, Bao H; A unified active and semi-supervised learning framework for image compression. Proc. IEEE Comput. Vis. Pattern Recognit., 2009; 65–72.
3. Levin A, Lischinski D, Weiss Y; Colorization Using Optimization. ACM Transactions on Graphics, 2004; 23:689–694.
4. Sapiro G; Inpainting the colors, IMA Preprint Series 1979, Institute for Mathematics and Its Applications, University of Minnesota, May 2004.
5. Takahama T, Horiuchi T, Kotera H; Improvement on Colorization Accuracy by Partitioning Algorithm in CIELAB Color Space. Lecture Notes in Computer Science, 2004.
6. Chapelle O, Sch¨olkopf B, Zien A, eds; Semi-Supervised Learning. Cambridge, MA: MIT Press. 2006.
7. Belkin M, Niyogi P, Sindhwani V; Manifold regularization: A geometric framework for learning from examples. Journal of Machine Learning Research, 2006; 7: 2399

8. Fathima S, Kavitha E; An Image Compression Technique Based on the Novel Approach of Colorization Based Coding. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2014; 3(5):9719-9726.