Color Image Compression Based on Contrast Sensitivity with Quality Factor

A. Christoper Tamilmathi[1](✉) and P. L. Chithra[2]

[1] Sri Ramachandra Faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research (Deemed to Be University), Chennai, Tamil Nadu 600 116, India
actamilmathi@gmail.com

[2] Department of Computer Science, University of Madras, Chennai, Tamil Nadu 600 025, India

Abstract. This paper presents an innovative lossy color image compression based on contrast sensitivity of human visual perception, in which the image contrast sensitivity value (CSV) of each block is determined from Symmetric Gaussian function (SG). Then the outcome of SG is grouped using K-Medoid (KM) cluster to define the adaptive block quantization table (ABQT) for each block in order to provide better quality reconstructed image with good compression ratio (CR). Our experimental result shows that the observed CR is averagely increased by 3 times and visual quality is averagely increased by 9.9% than the existing algorithms namely traditional JPEG and JPEG with contrast sensitivity technique. The tabulated results indicate that proposed methodology out performs very well with the structural content of natural color test images.

Keywords: Adaptive block quantization table · Cluster · Contrast sensitivity · JPEG · Lossy color image compression · Quality factor

1 Introduction

Images make a major portion of modern day digital information processing, storage, and transmission. Compression is essentially required to manage this high data-rate of images without degrading the quality to acceptable level. New compression method based on high degree of correlation between the RGB planes of a color image is reduced by transforming them to O1; O2; O3 planes. Each O plane is then encoded using BTC-PF method proposed [1]. In spatial domain, the JPEG-LS coder in the near-lossless compression mode is modified to make coding errors part of the perceptual redundancy in compressing color images in the RGB space. In the wavelet domain JPEG 2000 coder is refined by minimizing the perceptible distortion involved in the rate control of the compressed image in the YCbCr space stated in [2]. Different formulas were introduced for measuring the contrast sensitivity of an image such as Weber contrast; Michelson contrast and RMS contrast defined [3]. The extended technique of constructing membership functions of fuzzy set proposed in [4]. The color image compression is based on different correlation method in which the existing inter-color correlated is employed to approximate two of the components as a parametric function of the third one, called

R. Manza et al. (Eds.): ACVAIT 2022, AISR 176, pp. 382–399, 2023.
https://doi.org/10.2991/978-94-6463-196-8_29

the base component expressed in [5]. To get the best compression ratio the next step of proposed adaptive scanning providing for each (n, n) DCT block a corresponding (n $\times$ n) vector containing the maximum possible run of zeros at its end proposed in [6]. Available statistical quality metrics on compressed data analysis were discussed in [8].

Modified sigmoid function provides better enhancement of low contrast image than the fuzzy ruled based method [11]. JPEG compression algorithm works well when it is combined with K-Means cluster method explained in [12]. Joint Chroma subsampling and distortion-minimization based on Luma modification for RGB color images explained in [13]. The JPEG compression algorithm is improved by modified the luminance quantization table for color images that improved the CR, MSE and PSNR value expressed in [14]. A new lossy compression method PE-VQ is proposed which exploits prediction error and vector quantization concepts. An optimum codebook is generated by using a combination of artificial bee colony and genetic algorithms [15]. Quality of the compressed image using JPEG algorithm mainly depends on the quality factor [16]. In fast and efficient color image enhancement, only V component is stretched under the control of the parameters, namely average intensity value and contrast intensification proposed in [17]. Multiplier less efficient and low complexity 8- point approximate DCT were introduced in which only 17 additions required for both forward and backward transformation discussed in [18]. A new histogram equalization with automated estimate of number of clusters were produced depends on image brightness level displayed in [19] and image contrast enhancement based on intensity expansion-compression. By expanding the intensity, according to the polarity of local edges, an intermediate image of continuous intensity spectrum is obtained in [20].

In this article a novel symmetric Gaussian K-Medoid contrast sensitive (SGKMCS) lossy compression algorithm has been proposed and proved that this method performs well with compression ratio which has been averagely increased by 3 times and visual quality averagely increased by 9.9%. The rest of the paper is organized as follows. Section 2 explains the proposed work. Section 3 discusses the experimental results. Conclusion of the proposed work is given in Sect. 4.

2 Proposed Method

The seed of the proposed compression method is based on the JPEG lossy compression work. The general architecture of the proposed image encoding process is shown in Fig. 1. Since the human eye has different sensitivity to color and brightness, raw RGB image has taken from natural color image data base and it transformed to the YCbCr image [10]. Resultant YCbCr image has been decomposed into luminance and color components.

During the reconstruction process, this decomposition process can be reversible to RGB. Each component of an image is further divided into non-overlapping 8 $\times$ 8 image blocks. This compression method has been concentrated on the quantization process, which depends on the contrast value of each block in the component. Implementation of the proposed method has been explained below by the following sections.

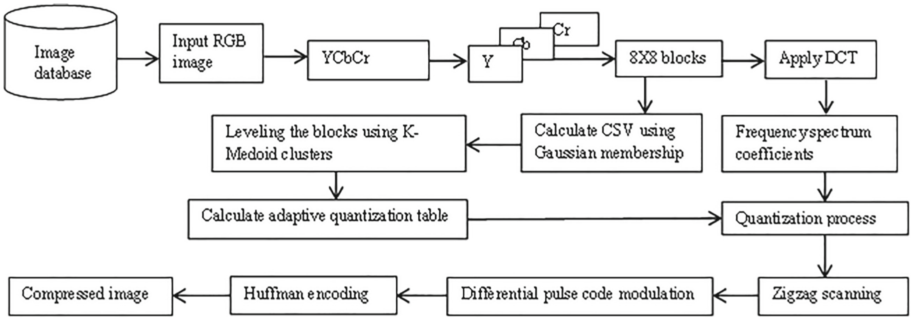

Fig. 1. The general architecture of proposed encoding process.

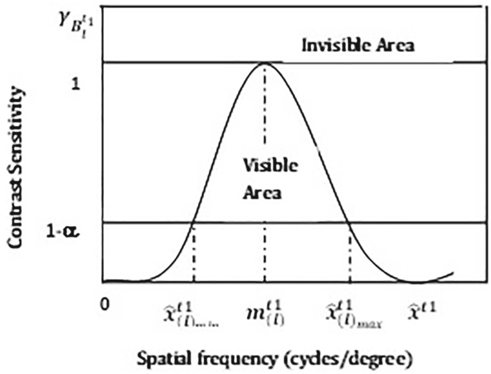

Fig. 2. Contrast Sensitivity function graph.

2.1 Contrast Sensitive Value Identification

Contrast value is the deviated value of brighter part from the darker part of an image [28]. This proposed method describes the new function to determine the contrast sensitive value (CSV) of a block, based on fuzzy activation function. CSV identified by the deviated value which is calculated from the symmetric Gaussian membership function denoted in Eq. (2.3). This method produces bell shaped graph for a block which denotes the contrast sensitivity of each pixel in the corresponding block as shown in Fig. 2. In a graph the contrast sensitivity function has the maximum peak value is 1. Visibility of the human eye perception decreases when increases the contrast sensitivity value from bottom to top with spatial frequency. Inner part of the contrast sensitivity function graph specifies the visibility area and the outer part of the graph as invisibility area. The difference between the visibility appearance and visibility disappearance measured as contrast sensitivity value (deviation). From the contrast sensitivity graph, a CSV value has been calculated using Eq. (2.4). This fuzzy based CS function mainly depends on two parameters, namely spatial frequency value for the corresponding peak contrast sensitive value $\left(m_{(l)}^{t_1}\right)$ and the deviated frequency from the peak value $\left(\sigma_{(l)}^{t_1}\right)$ of each block [4], have been determined by the following Eqs. (2.1) and (2.2).

Spatial frequency of a peak contrast sensitive value $\left(m^{t_1}_{(l)}\right)$ calculated by

$$m^{t_1}_{(l)} = \frac{\left(\hat{x}^{t_1}_{(l)min} + \hat{x}^{t_1}_{(l)min}\right)}{2} \tag{2.1}$$

where $t_1 = 1, 2,\ldots, m_l$ and $l = 1, 2,\ldots,$ c. Where m_l is dimensional input vector, l is a band width between $\hat{x}^{t_1}_{(l)min}$ and $\hat{x}^{t_1}_{(l)max}$, $\hat{x}^{t_1}$ is an input vector and c is dimensional output vector. Parameter $\hat{x}^{t_1}_{(l)min}$ referred as a visibility appearance frequency and $\hat{x}^{t_1}_{(l)max}$ referred as a visibility disappearance frequency.

Frequency deviation from $m^{t_1}_{(l)}$ at left side denoted as $\sigma^{t_1}_{L(l)}$ and frequency deviation from $\mathrm{m}^{\mathrm{t}_1}_{(\mathrm{l})}$ $\mathrm{m}^{\mathrm{t}_1}_{(\mathrm{l})}$ at right side denoted as $\sigma^{t_1}_{R(l)}$. From these two deviated frequency, $\sigma^{t_1}_{(l)}$ (frequency deviation from this CSV graph) selects the maximum deviated frequency value which has been given in Eq. (2.2).

$$\sigma^{t_1}_{L(l)} = \frac{\left(\hat{x}^{t_1}_{(l)min} + m^{t_1}_{(l)}\right)}{\sqrt{-2\ln(1-\alpha)}}$$

$$\sigma^{t_1}_{R(l)} = \frac{\left(\hat{x}^{t_1}_{(l)max} + m^{t_1}_{(l)}\right)}{\sqrt{-2\ln(1-\alpha)}}$$

where α is a visibility threshold between [0,1]. Using these two equations deviated frequency value has been calculated by the Eq. (2.2).

$$\sigma^{t_1}_{(l)} = \max\left\{\sigma^{t_1}_{L(l)}, \sigma^{t_1}_{R(l)}\right\} \tag{2.2}$$

Contrast sensitivity function $\gamma^{t_1}_{(Bl)}$ of a frequency set $B^{t_1}_{(l)}$ has obtained as follows

$$\gamma^{t_1}_{(Bl)} = exp\left[-\frac{1}{2}\left[\frac{\hat{x}^{t_1} - m^{t_1}_{(l)}}{\sigma^{t_1}_{(l)}}\right]\right] - \infty < \hat{x}^{t_1} < \infty \tag{2.3}$$

After calculating the contrast sensitivity $\gamma^{t_1}_{(Bl)}$ of a block, the contrast sensitivity value has been calculated by using the range of frequency between the lower range(visible part) and the higher range (invisible part) which has been given below in Eq. (2.4).

$$\mathrm{CSV} = \max\left\{\gamma^{\mathrm{t}_1}_{(\mathrm{Bl})}\right\} - \min\left\{\gamma^{\mathrm{t}_1}_{(\mathrm{Bl})}\right\} \tag{2.4}$$

Symmetric Gaussian membership based contrast sensitivity function $\gamma^{\mathrm{t}_1}_{(\mathrm{Bl})}$ of the spatial frequency set $B^{t_1}_{(l)} = \left(m^{t_1}_{(l)}, \sigma^{t_1}_{(l)}\right)$ is representing in Fig. 2. CSV calculation explained in Algorithm 2.1.

Algorithm 2.1 CSV calculation

Input: Blocks of a component.

Output: CSV of a component.

1. *for* each block in the component *do*
2. Calculate central value of the given block using equation (2.1)
3. Calculate probability density value using equation (2.2)
4. Draw a bell shaped graph using Symmetric -Gaussian member function using equation (2.3).
5. Find the deviated value from the result of -step 4 using equation (2.7)
6. Deviated value as the contrast sensitivity value
7. *end for*

2.2 CSV Based Block Leveling

CSV based block leveling has done by cluster of blocks using K-Medoid cluster. Procedure of block leveling technique is defined in Algorithm 2.2. This suggested method describes the cluster process which has been uniformly partitioned the whole component blocks into K number of clusters [32]. Blocks have been clustered based on the contrast value of each block and the reference value (number of blocks). Nearest CSV of blocks clustered into the same group. This clustering process continued until the cluster size reached the maximum size mentioned as reference value. During the cluster process, maximum CSV of the previous cluster is considered as the initial minimum CSV (medoid) of the current cluster for finding the minimum distance CSV. Minimum distance blocks identified from Eq. (2.5) and medoid value updated from Eq. (2.6). Level value is assigned to each cluster in the component. A higher level block requires more quantize than the lower level CSV blocks. Level1 is assigned to minimum CSV blocks and level 4 assigned to maximum CSV blocks. Other levels are assigned to the remaining CSV blocks. Block set B expanded as

$$B = B_1^{CSV}, B_2^{CSV}, \cdots, B_N^{CSV}$$

where N = 4096 blocks. K is the number of clusters. Medoids are represented as m1, m2, m3 and m4. Initially the clustering process starts with m1 = 1 and compare the each CSV to determine the minimum distance value. The obtained CSV cluster it into same group. Reference value of each cluster is 1024.

$$Reference\ value\ n\ = N/K = 4096/4 = 1024$$

$$Initially\ m_1 = 1$$

$$K_i = \|B_i^{csv} - m_i\| \leq \|B_i^{CSV} - m_i\| \text{ and blocks}(K_i) < n \tag{2.5}$$

$$j = 1, 2, 3, \cdots, N.i = 1, 2, 3, \cdots, N.$$

$$m_i = maximum\ distance\ CSV(K_{l-1}) \tag{2.6}$$

where N is the total number of blocks in a component, n is the number of blocks in each cluster, K as initialized as four that represents number of clusters in a component.

Algorithm 2.2 Blocks leveling

Input: CSV of a component.

Output: Blocks with leveling value.

1. Initialize the K value as 4 Assign first medoid m1=1
2. *for* each block in the component *do*
3. Determine the minimum distance block by
   -comparing the CSV and check the reference
   -value which is lesser than n using equation (2.5)
4. Cluster it as the same and assign the cluster level
5. Update the medoid value with maximum CSV
   - of previous cluster using equation (2.6)
6. *end for*
7. Assign minimum CSV cluster as the level1.
8. Assign maximum CSV cluster as the level4.
9. Assign the in-between CSV cluster as the level2 and level3.

2.3 Discrete Cosine Transformation

DCT has been separated the image into spectral sub-bands of differing importance with respect to the image visual quality [6]. This proposed method has transformed every 8 × 8 block image into high and low level of frequency spectrum to concentrate the compression on the high frequency spectrum [32].

2.4 Adaptive Block Quantization Table

Adaptive block quantization table (ABQT) formation depends on the JPEG standardized quantization table, cluster level and QF value explained in Algorithm 2.3. QF controls the quality of the compressed image with ranging from 1 to 100. Higher cluster level value improved the ABQT value for making better compression on the corresponding blocks. The ABQT calculation given in Eq. (2.8) below

$$q = \begin{cases} \left(\frac{QF}{50}\right) + 1, & \text{if } QF \leq 50 \\ \left(\frac{QF}{50}\right) - 1, & \text{if } QF > 50 \end{cases} \tag{2.7}$$

$$\text{ABQT}(i, j) = (\text{QT}(i, j) \times q) + \text{cluster level value} \tag{2.8}$$

QT (i, j) is the JPEG standard quantization table value [14] for corresponding *(i, j)* coordinate value of the luminance and chrominance component, *q* is a quality factor value from Eq. (2.7).

Algorithm 2.3 Adaptive block quantization table calculation

Input: Level value, JPEG quantization table (QT) and QF.

Output: Adaptive block quantization table.

1. *for* each block in a component *do*
2. *if* QF<=50 *then*
3. q=(QF/50)+1
4. *else*
5. q=(QF/100)-1
6. *end if*
7. ABQT= (QT × q) + cluster level value
8. *end for*

2.5 Quantization

The Quantization is an essential part in compression algorithm to reduce the number of bits per sample [33]. In this proposed method, quantization process concentrated on contrast sensitive value of each block. ABQT has been created for individual block depends on CSV not like JPEG quantization table which depends on component. This quantization process performed on the DCT spectrum coefficients by ABQT has been given in Eq. (2.9).

$$F(u, v) = round\left(\frac{f(u, v)}{ABQT}\right) \tag{2.9}$$

F(u,v) is an quantized image, *f(u, v)* is a DCT spectrum coefficient. During the quantization process, the coefficients have been separated into DC and AC coefficients, which have been reordered into 1-D format using a zigzag scanning in order to create long run of zero valued coefficients[29]. All DC coefficients are combined to form a separate bit stream. Encode the difference from the DC component of previous 8 × 8 blocks with the next block, i.e. Differential Pulse Code Modulation (DPCM) [30].

2.6 Huffman Entropy Coding

Huffman coding is popular entropy coding for compressing the data with variable-length codes. This method constructs a set of variable-length code words with the shortest average length and assigns them to the symbols [31]. The symbol with highest probability has assigned the shortest code using symbol table and vice versa. The less sized bit stream has been produced from the Huffman encoding process, which has been considered as the compressed image [32]. Compressed image is the final output of encoding process. The above all explained methods have applied on the proposed SGKMCS compression method is explained in Algorithm 2.4.

Algorithm 2.4 Proposed work

Input: RGB image data base.

Output: Reconstructed RGB image database.

1. Get an input RGB image from the database.
2. Covert the RGB image into YCbCr image format
3. Isolate the individual component Y, Cb and Cr from the YCbCr image format.
4. *for* each Y, Cb and Cr component *do*
5. Decompose the individual component into non- overlapped 8×8 blocks image.
6. *for* each input 8×8 block image *do*
7. Call CSV calculation function (Algorithm 2.1)
8. Call block leveling function (Algorithm 2.2)
9. Call Adaptive block Quantization - table calculation function (Algorithm 2.3)
10. *end for*
11. Transform each input 8 × 8 image block using - DCT into frequency spectrum.
12. Quantization process on the frequency spectrum using adaptive quantization table which is calculated from step 9 using equation (2.9).
13. Make a single vector values using Zigzag scanning
14. Apply Differential Code Modulation.
15. Produce bit stream using Huffman -Entropy encoding.
16. Calculate CR of an image.
17. Do the inverse process of Encoding.
18. *end for*
19. Reconstruct the YCbCr image.
20. Reconstruct the RGB image.
21. Measure the Quality metrics of all images in the database.

3 Experimental Results

The proposed method has been implemented in Matlab 2017b on 67 input color images which have been taken from the color image database [29] with different QF ranging from 1 to 100. The performance of the proposed work has been compared with the existing color image compression methods, namely JPEG and JPEG with contrast sensitivity function. JPEG compression has been implemented the component based quantization in which the quantization elements are constants. JPEG with contrast sensitivity compression experimented based on the component based quantization in which quantization elements are contrast detection threshold, derived from [10]. Figure 3 shows the 65 input color image database which has 24 bits jpg images with different resolution like 768 × 576, 428 × 569, 300 × 168 and 576 × 768.

In the proposed SGKMCS method, one of the sample RGB input image #13 shown in Fig. 4(a). It has been converted to YCbCr image format, and then converted to an individual isolated Y, Cb and Cr components. Each individual component is further divided into non-overlapping 8 × 8 blocks. Contrast sensitivity function (CSF) applied to every block, the sample output graph shown in Fig. 4(b). From the CS graph the CSV has been calculated. CSV of Y component in image #13 is shown in Fig. 4(c). The blocks are clustered on basis of CSV using K-Mediod Cluster. In this proposed method K represents 4. The whole number of blocks was equally partitioned into 4 clusters in which each cluster has 1024 blocks. The partitioned clusters based on the CSV are shown in 4(d). Parallel work of 8 × 8 block transformed using DCT, as shown in Fig. 4(e).

Adaptive block quantization table generated based on JPEG standard quantization table, block cluster level and QF. Some sample ABQT of proposed SGKMCS method shown below. For example, in an experimental methodology when K = 1, QF = 40 and

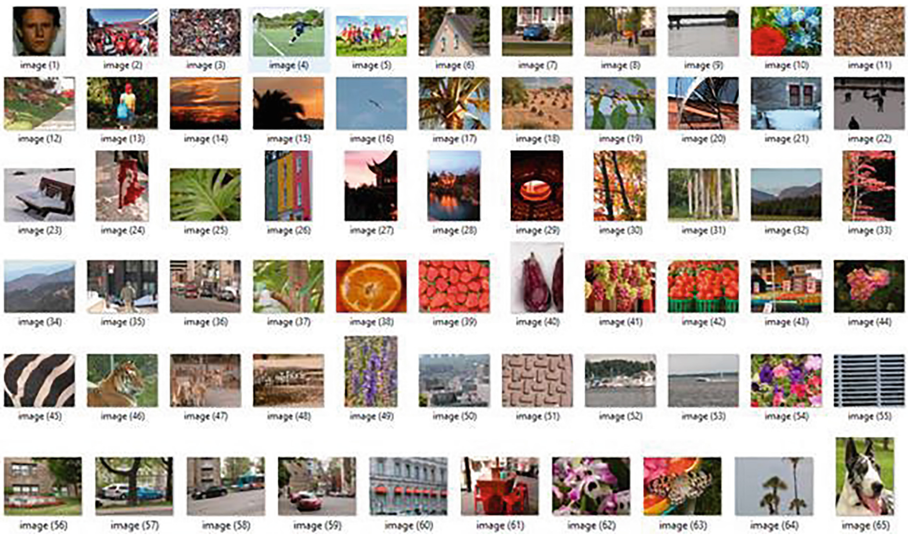

Fig. 3. Source RGB image database

first block in Cb component of an image #20 in Fig. 3 obtained the ABQT denoted as follows.

$$ABQT_{40}^{1}(1) = \begin{bmatrix} 32 & 33 & 44 & 86 & 179 & 179 & 179 & 179 \\ 33 & 39 & 48 & 120 & 179 & 179 & 179 & 179 \\ 44 & 48 & 102 & 120 & 179 & 179 & 179 & 179 \\ 86 & 120 & 179 & 179 & 179 & 179 & 179 & 179 \\ 179 & 179 & 179 & 179 & 179 & 179 & 179 & 179 \\ 179 & 179 & 179 & 179 & 179 & 179 & 179 & 179 \\ 179 & 179 & 179 & 179 & 179 & 179 & 179 & 179 \\ 179 & 179 & 179 & 179 & 179 & 179 & 179 & 179 \end{bmatrix}$$

Quantization process is performed on each block depending on their contrast value of the transformed image using ABQT. After that zigzag scanning process performed on the quantized blocks convert the matrices into the vectors for Huffman encoding. Then the Huffman encoding process encodes the image into stream of bits in the form of compressed image. Finally compressed image with less size can be transformed from one place to another through the communication channel. Decompression process is the inverse process of encoding method.

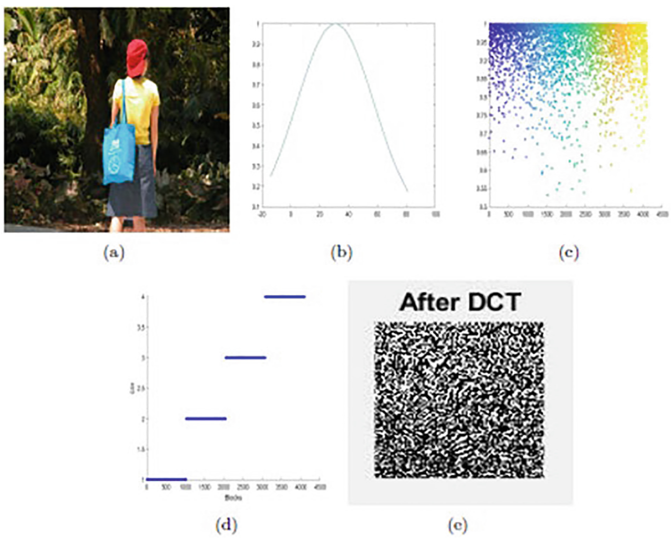

Fig. 4. Process of proposed method on one of the sample of Y component of an image #13 (a) Original RGB image #13, (b) Sample CSF graph of first block of Y component, (c) CSV of all blocks in Y component, (d) Block level cluster of Y component and (e) DCT coefficient of image #13.

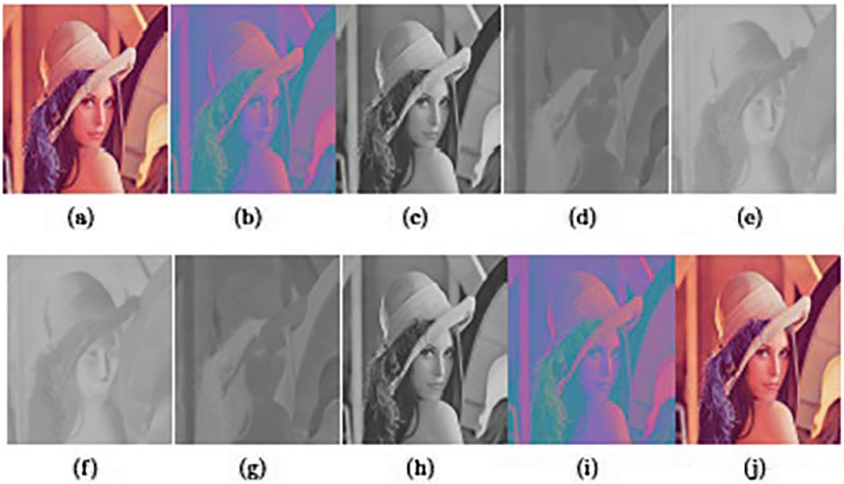

Fig. 5. Output of the proposed method using Lena image (a) Original Lena image (b) YCbCr image (c) Luminance image (d) Cb image (e) Cr image (f) Reconstructed Cr image (g) Reconstructed Cb image (h) Reconstructed Luminance image (i) Reconstructed YCbCr image (j) Reconstructed RGB image.

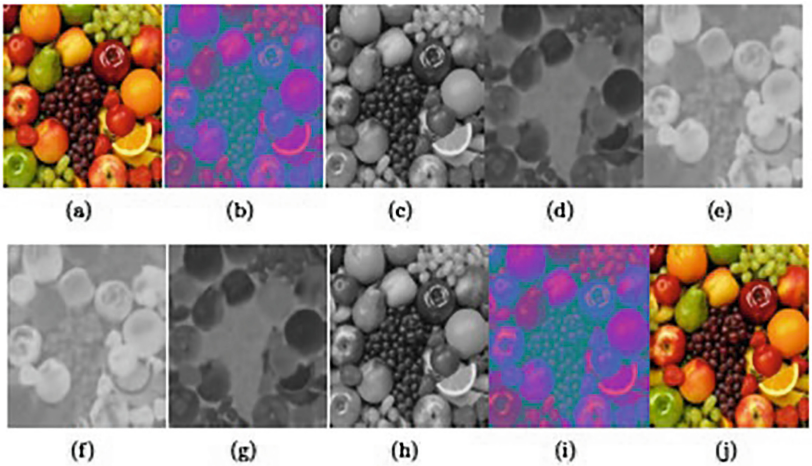

Fig. 6. Output of the proposed method using Fruits image (a) Original Fruits image (b) YCbCr image (c) Luminance image (d) Cb image (e) Cr image (f) Reconstructed Cr image (g) Reconstructed Cb image (h) Reconstructed Luminance image (i) Reconstructed YCbCr image (j) Reconstructed RGB image.

The proposed method has been applied on Lena and Fruits images with 512×512 sizes. Encoding process of Lena and Fruits images shown in Fig. 5(a) to Fig. 5(e) and Fig. 6(a) to Fig. 6(e). Decoding process of Lena and Fruits images shown in Fig. 5(f) to Fig. 5(j) and Fig. 6(f) to Fig. 6(j).

Fig. 7. Comparison of the proposed SGKMCS reconstructed images with the existing methods image (a) Original Lena image, (b) Reconstructed image from JPEG, (c) Reconstructed image from JPEG with contrast sensitivity method (d) Reconstructed image from proposed SGKMCS method (e) Original Fruits image, (f) Reconstructed Fruits image from JPEG (g) Reconstructed image from JPEG with contrast sensitivity method (h) Reconstructed image from proposed SGKMCS method.

Proposed reconstructed Lena and Fruits images have been compared with the existing JPEG and JPEG with contrast sensitive compression methods shown in Fig. 7. We observed that visual quality of proposed reconstructed images as shown in Fig. 7(d) and Fig. 7(h) are more contrast than the other two images. From this figure, this work concluded that the original color image is perfectly reconstructed from the compressed image using the proposed SGKMCS compression work. There is some minimum difference in color information due to the work based on lossy compression method. The performance efficiency of the proposed method compared with the existing methods [10] on the basis of CR and SSIM are tabulated in the Table 1.

Result from the table shows that compression ratio averagely increased by 3 times and visual quality averagely increased by 9.9% than the existing methods JPEG and the JPEG with contrast sensitivity value [10]. According to the Table 1, the proposed work produced better compression ratio and lesser SSIM when QF is 50. Maximum QF (100) has been produced better SSIM of reconstructed image and lesser CR. From the comparison table, this work concludes that the proposed quantization process based on contrast sensitivity value of each block in the component produce better compression ratio than the component based existing method. New SGKMCS compression method applied on all source images in the database with QF is 100 and produced an effective compression ratio that has been displayed using histogram in Fig. 8.

In Fig. 8 image #1(single human image) has the higher CR of chrominance red component than the other images. Image #34(landscape) has the second higher CR of chrominance component value than the remaining 63 images.

Table 1. Performance Comparison of Proposed SGKMCS with Existing PEG and JPEG- CSF

Parameters		Compression Ratio				SSIM			
		Y-Components	CB Components	CR Components	RGB image	SSIM_Y	SSIM_CB	SSIM_CR	SSIM_RGB
Fruits	**JPEG-CSF**	15.3723	42.7188	39.696	26.3963	0.9923	09706	09854	0.9861
	JPEG	16.9959	34.611	34.5472	25.7122	0.9717	0.9694	0.9703	0.9422
	SGKMCS (QF=1)	52.4690	68.0457	67.9853	62.9853	0.9864	0.9778	0.9672	0.9623
	SGKMCS (QF=50)	57.7	83.508	81.4305	74.24	0.9679	0.9523	0.9453	0.9412
	SGKMCS (QF=100)	31.7222	45.8771	46.8839	41.4944	0.9993	0.9985	0.9765	0.9753
Lena	**JPEG**	19.6588	46.2308	41.3889	31.0367	0.9591	0.9464	0.9467	0.9048
	JPEG-CSF	19.7133	58.4927	37.9017	31.8446	0.9889	0.9585	0.9876	0.9507
	SGKMCS (QF=1)	55.7279	88.6366	76.6034	73.6559	0.9768	0.9774	0.9517	0.9500
	SGKMCS (QF=50)	64.9343	119.6279	106.9321	97.1648	0.9563	0.9564	0.9341	0.9312
	SGKMCS (QF=100)	28.6303	59.4415	48.3659	42.1459	0.9989	0.9980	0.9979	0.9954

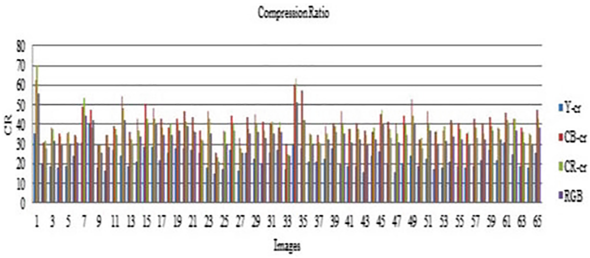

Fig. 8. Compression Ratio of all source images with QF = 100.

These observed results from our experimental method, implied that, this proposed method maximum of images having the higher CR of chrominance red component than the chrominance blue components.

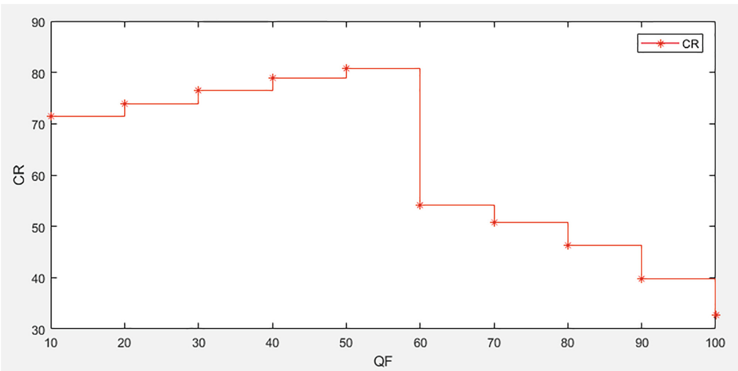

Fig. 9. Stair graph of CR with QF.

The Stair graph in Fig. 9 represents the relationship between CR and QF of the reconstructed image #20. Compression ratio increases as the QF increases from 10 to 50. Once QF reaches 50, CR decreases from the maximum value.

The proposed work has been applied on all the source images with 3 different QF (10, 50 and 100) and calculated the CR which has been plotted in Fig. 10. From this graph, the work concluded that maximum CR has been produced when the QF is 50 and the minimum CR has been generated when the QF is 100. It means that higher QF preserves the quality of the original image.

Figure 11 shows the quality measurement of all test images such as Mean Squared Error (MSE) and Peak Signal Noise Ratio. From that graph, observed that all the test images achieved better PSNR (more than 45 Decibel) value which implies that this proposed work maintains the quality of the original image.

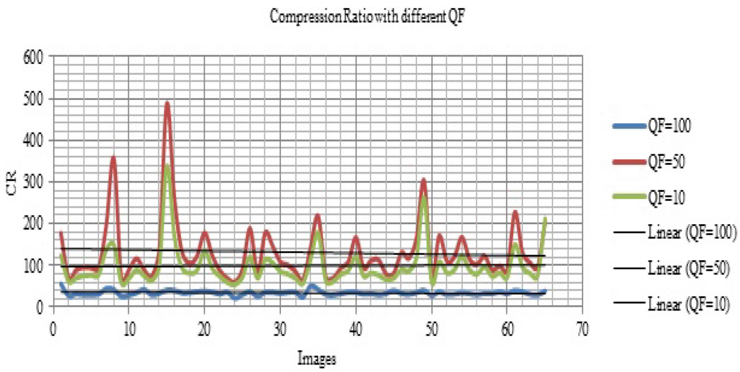

Fig. 10. Compression Ratio (CR) of all test images with three different Quality Factor (QF) values.

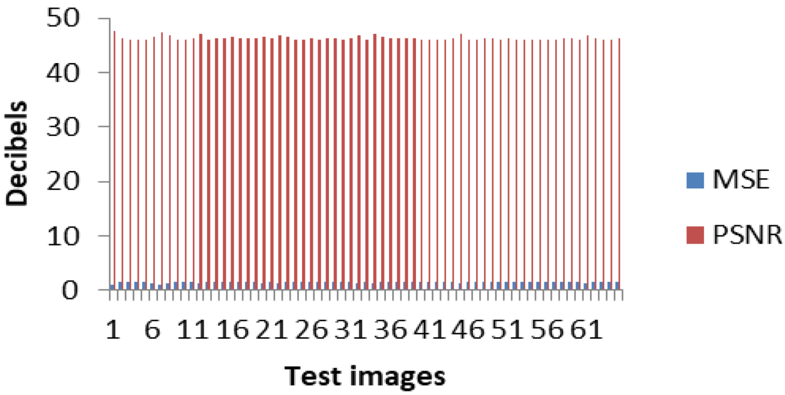

Fig. 11. Quality measurement of all test images.

4 Conclusion

This paper proposed a new methodology SGKMCS lossy compression to improve the compression ratio and visual perception of the decompressed color images. A novel SGKMCS compression algorithm is based on the contrast sensitivity value calculated by symmetric Gaussian member function on a block. ABQT has improved the quantization process than the existing methods namely JPEG and JPEG with contrast sensitivity function. The experimental result shows that proposed work averagely increased the CR by 3 times and SSIM improved by 9.9%. In Future work this method can be extended to 3D Compression with automatic analysis using Artificial Neural Network.

References

1. Bihas C.D., Bhubatosh, C.: Color image construction based on block truncation coding using pattern fitting principle. Journal of the Pattern Recognition Society, 40, 2408–2417, (2007).
2. Chou, C. H., Liu, K.-C.: Color image compression based on the measure of just noticeable color difference. IET image processing, 2 (6), 304–322, (2008). https://doi.org/10.1049/iet-ipr:20080034.
3. Denis, G. P., Peter, B.: Measuring contrast sensitivity. Vision Research, 90, 10 -14, (2014).
4. Dmitri, A. V., Reyhane, T., Aniaksander, D.: Designing Gaussian membership functions for fuzzy classifier generated by heuristic possibility clustering. Journal of information and organizational sciences 37 (2), 127–139, (2013).
5. Evgeny, G, Emilia, L., Moshe, P.: Correlation-based approach to color image compression. Signal processing: Image communication, V. 22, 719–733, (2007). https://doi.org/10.1016/j.image.2007.04.001
6. Fouzi, D., Redha, B, Nabil, B.: Color image compression algorithm based on the DCT transform combined to an adaptive block scanning. International Journal of electronics and communication 65, 16–26, (2011). https://doi.org/10.1016/j.aeue.2010.03.003
7. Haiqiang, W., Ioannis, K., Jiantong, Z., Jeonghoon, P., Shawin, L., Xin, Z., Man-On, P., Xin, J., Ronggang, W., Yun, Z., Jiwu, H., Sam, K., C.-C.Jay, K.: VideoSet: A large-scale compressed video quality dataset based on JND measurement, Journal of visual communication and image representation, 46, 292–302 (2017).
8. Ismail, A., Bulent, S.: Statistical analysis of image quality measures. IEEE, (2015)
9. Joe, Y. L., Lina, J., Sudeng, H., Ioannis, K., Zhi, L., Anne, A., Jay , K. C. -C.: Experimental design and analysis of JND test on coded image/video. Applications of Digital image processing, (2015).
10. Juncai, Y., Guizhong, L.: A novel color image compression algorithm using the human visual contrast sensitivity charecteristics.Photonic sensors, 7(1), 72–81, (2017).
11. Kannan, P, Deepa, S., Ramakrishnan, R.: Contrast enhancement of sports image using two comparative approaches. American Journal of Intelligent Systems, 2(6),141–147, (2012).
12. Karri, T. R., Ramesh, C.: Baseline JPEG compression with K-Means clustering based algorithm. International Journal of computer science and mobile computing, 4 (4), 379–386, (2015).
13. Ku-Liang, C., Tsu-Chun, H., Chi-Chao, H.: Joint chroma subsampling and distortion- minimization based luma modification for RGB color images with application. IEEE transaction on Image processing, 1–3, (2017). PubMed ID: 28650815. https://doi.org/10.1109/TIP.2017.2719945

14. Lamia A., Pranab K. D., Mirza, A.F.M., Rashidul H.,Mohammed G. S. B.,Golam M. D.: An improved JPEG image compression algorithm by modifying luminance quantization table.International Journal of computer science and network security, 17,200–208, (2017).
15. Mohamed, U. A. A., Eswaran, C., Kannan,R.: Lossy image compression based on error and vector quantisation. Eurasip journal on image and video processing, (2017).
16. Rahul, P., Nindhiya, K., Gurjeet, S., Hiteshwari, S.: Image compression and quality factor in case of JPEG image format. International Journal of Advanced Research Computer and communication Engineering, 2(7), 2578–2581, (2013).
17. Raju, G, Madhu, S. N.: A fast and efficient color image enhancement method based on fuzzy-logic and histogram. International Journal of Electronic and communications, 68, 237-243, (2013).
18. Reem, T. H.,Wail, S. E., Hassan, H. R.: Fast approximate DCT with GPU implementation for image compression. Journal of visual Communication and image representation, 357–365, (2016). https://doi.org/10.1016/j.jvcir.2016.07.003
19. Shakeri, M., Dezfoulian, M. H., Khotanlou, H., Barati, A. H., Masoumi, Y.: Image contrast enhancement using fuzzy clustering with adaptive cluster parameter and sub-histogram equalization. Digital signal processing, (2016). https://doi.org/10.1016/j.dsp.2016.10.013
20. Shilong, L., Md, A. R., Ching-Feng, L., Chin, Y. W., Guannan, J., San, C. L., Ngaim-ing, K., Haiyan, S.: Image contrast enhancement based on intensity expansion-compression. Journal of visual communication image representation, 48, 169–181, (2017). https://doi.org/10.1016/j.jvcir.2017.05.011
21. Sreelekha, G., Sathidevi, P. S.: An HVS based adaptive quantization scheme for the compression of color images. Digital signal processing, 20, 1129–1149, (2010). https://doi.org/10.1016/j.dsp.2009.12.003
22. Stefen, W.: Analysis of public image and video databases for quality assessment. IEEE Journal - Signal processing, 6 (6), (2012). https://doi.org/10.1109/JSTSP.2012.2215007
23. Sudeng, H., Haiqiang, W., Jay, K. C.- C.: A GMM-based stair quality model for human perceived JPEG images. IEEE, (2016).
24. Walaa, M. A., Wajeb, G.: Color image compression algorithm based on the DCT blocks. Institution of Engineering and Technology, 52 (20), (2016).
25. Wang, X., Jiang, G. Y., Zhang, J. M., Shao. F., Peng, Z. J., Yu, M.: Visibility threshold of compressed stereoscopic image: effects of asymmetric coding. The Imaging Science Journal. 61 (128), (2013). https://doi.org/10.1179/1743131X11Y.0000000035
26. Yao, J., Liu, G.: An adaptive quantization method of image based on the contrast sensitivity characteristics of human visual system. Journal of Electronics and Information Technology, 38 (5), (2016).
27. http://psy.vanderbilt.edu/courses/hon185/spatialfrequency/spatialfrequency.html accessed on 10.01.2018.
28. Image samples are taken from https://www.cs.cmu.edu/afs/cs/project/cil/www/V-image.html accessed on 12. 11. 2017.
29. Chithra, PL., Thangavel, P.: A New Efficient Embedded Wavelet Image Codec Based On Multidirectional Traversal Algorithm Based On Multidirectional Traversal. Int. J. Signal and Imaging Systems Engineering, Vol. 5, No. 3, (2012). ISSN online: 1748–0701 ISSN print: 1748–0698
30. Chithra, PL., Christoper Tamilmathi, A.: 3-d wavelet codec (compression/decompression) for 3-d Medical Images. International Journal of Information Technology Convergence and Services (IJITCS),Vol.6, No.1, 45–54, (2016) ISSN 2231–153X: 2231–1939
31. Chithra, PL., Srividhya, K.:A Comparative Study of Wavwlet Coders for Image Compression. Springer International Publishing Switzerland (2013), pp. 260–269

32. PL Chithra, PL., Christoper TamilMathi, A.: Effective lossy and lossless color image compression with multi layer perceptron. International Journal of Engineering and Technology, 7 (2.22) (2018) 9–14, ISSN: 0975–4024.
33. Chithra, PL., Christoper Tamilmathi, A.: Image Preservation using Wavelet Based On Kronecker Mask, Birge-Massart And Parity Strategy International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278–3075, Volume-8 Issue-11, September (2019).