



# Color image quantization with peak-picking and color space

Taymaz Rahkar Farshi<sup>1</sup>

Received: 2 July 2020 / Accepted: 3 August 2020 / Published online: 12 August 2020  
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

## Abstract

Color image quantization is a significant procedure of reducing the huge range of color values of a digital color image into a limited range. In this paper, an automated clustering of pixels and color quantization algorithm is proposed. The ideal number of representative colors is unknown beforehand in most color quantization algorithms. This is an important handicap in most practical cases. The proposed color quantization approach (PPCS) is able to automatically estimate an appropriate number of colors in a quantized palette. Hence, PPCS requires no number of representative colors to be set in advance. This algorithm has two main steps to follow: color palette design and pixel mapping. The color palette is generated by the combination of the entire peaks of all color component histograms. Such that, all color component histogram was smoothed in order to remove unreliable peaks. Next, unreliable colors will be removed from the palette. Then, each pixel in the image will be assigned to the cluster (unit color in the palette) which has the least Euclidean distance. To evaluate the ability of the PPCS, 22 images from Berkeley segmentation dataset have been randomly selected and tested with PPCS and also by two well-known quantization algorithms. The numerical evaluations have been carried out by using computation time, PSNR, MSE, and SSIM performance criteria. Both visual and numerical evaluations reveal that the proposed method presents promising quantization results. Such that, PPCS is ranked first, second, first and first according to PSNR, MSE, SSIM and computation time, respectively.

**Keywords** Color quantization · Image display · Peak detection · Clustering

## 1 Introduction

RGB is a simple color scheme image which consists of three primary red (R), green (G), and blue (B) color components. Moreover, there is a total of 24 bits per pixel (8 bits for each color component). It is evident that each color component values range lies in  $[0, 255]$ . Hence, there are  $2^{24} \approx 16.8$  million possible colors. This image type is commonly used for transmission, representation, and storage of color images on both analog and digital devices [1–4]. Color image quantization is a significant procedure of reducing the huge range of color values of a digital color image into a limited range. This scheme may be used in displaying devices with a limited color range, color image compression or reducing

the transfer time of the image in a limited network traffic. Furthermore, a desirable quantization approach considers not only low computational complexity but also low color distortion. In other words, a desired output of quantization algorithm must be gained instantly with a minimum range of color values in a color palette. Additionally, the output must be visually similar to the original image, as far as possible. Although many algorithms have been presented for color quantization, the number of colors in a color palette is a predetermined parameter. Also, there are two types of color palette generating methods: image-independent method and image-dependent method. The former generates the color palette without taking the particular image contents into consideration. In the latter, however, the color palette is generated in regards to the color distribution of images [5].

Color quantization can be divided into two main groups, scalar quantization and vector quantization. In scalar method each color component range is limited to smaller range independently. So, the quantization process is done irrespective of the original color space distribution. It is worth mentioning that a scalar scheme is only ideal if image colors

---

Communicated by Y. Zhang.

✉ Taymaz Rahkar Farshi  
taymaz.farshi@altinbas.edu.tr

<sup>1</sup> Software Engineering Department, Altinbas University, Istanbul, Turkey

are uniformly distributed in RGB color space. Nevertheless, the distribution of colors in generic color images is by no means uniform. On the other hand, vector quantization considers all color components together, contrary to scalar quantization. Therefore, each pixel is considered as a color vector  $p_i = (r_i, g_i, b_i)$  which is regarded as a single unit [1]. Consequently, vector quantization likely outperforms scalar quantization. However, due to the large search space, computational time needed for vector quantization is longer.

Vector quantization is categorized into two groups: splitting algorithm [6–14] and clustering-based algorithm [15–24]. In splitting approaches the color space of the given image is split into a set of disjointed regions. In other word, the color cube is divided into many prisms. On the other side, in clustering method the color palette is structured in regards to cluster centroids. Therefore, optimum predefined  $K$  cluster centroids is figured out by means of clustering algorithm such as  $k$ -means or  $C$ -means. Then each cluster centroid represents a unit color in the color palette.

Median-cut algorithm is a well-known splitting-based method proposed by Heckbert [6]. The RGB cube is repeatedly split into smaller prisms along one of the color components axes. In each step, the sub-prisms with highest pixel count is selected for splitting. The optimal cutting point is selected from a point where both sub prisms contain almost equal number of image pixel. This procedure is repeated until the predefined number of representative colors is touched. Later the mean vector of the contained colors in each prism is selected to represent the image.

The popularity algorithm is also proposed by Heckbert [6]. This method chooses  $k$  number of colors to represent the image. For that reason,  $k$  regions of image color distribution with high density are selected for the color palette.

The center-cut algorithm [7] makes three efficient modifications on the median-cut scheme in order to achieve a more effective result. At the first step the sub-prism with the longest-dimension is selected for splitting instead of the sub-prism with the largest number of pixels. Next, the prism is cut from the center of the prism instead of cutting from a point where both sub-prisms contain almost equal number of image pixels. Finally, instead of cutting the three significant bits off each color component, three, two and four bits of the red, green and blue component are cut respectively.

On the other hand, there is another approach which is based on median-cut, called variance-based algorithm [10]. In this approach the prism with the highest weighted variances of color distribution is selected for splitting. Moreover, the prism is split along the axis which reduces the variance more than the other two axes. Lastly, the sum of squared errors from prisms are almost equal.

Another alternative splitting-based approach is Octree algorithm [11] which is also like median-cut algorithm. In this method the color cube can be divided into eight sub

prisms. Then, the further sub prisms are divided into eight smaller prisms. The dividing operation continues until the  $K$  number of predefined sub prisms are constructed. Later the average of color in sub prisms represents the colors in the color palette.

Moreover, clustering-based quantization algorithms divide pixels into predefined  $K$  clusters with respect to pixel similarity. The  $K$ -means clustering scheme is widely applied in color quantization [25–29]. The  $K$  number of cluster centroids is randomly initialized. Afterwards, the initialized centroids are improving iteratively in order to minimize the similarity error. In each iteration pixels are assigned to the closest centroid. Thus, pixels associated with the same centroid represent a cluster. Subsequently, cluster centroids are updated by calculating the mean of all pixel values associated with the same centroid. Next, each cluster centroid represents colors in the color palette. Furthermore, Fuzzy  $c$ -means (FCM) also has been well applied in color quantization [30–32]. FCM is the extended version of  $k$ -means such that each cluster member can belong to many clusters instead of exactly one cluster. There is a membership value between each pixel and cluster centroid and this value indicates the level of the association between that pixel and a cluster centroid [33, 34].

The rest of this paper is arranged as follows: Sect. 2 describes the proposed method; Sect. 3 includes the comprehensive experimental results and discussion. Finally, Sect. 4, provides the conclusions.

## 2 Proposed color quantization algorithm

The purpose of the color quantization algorithm is to create an image with almost the same quality as the original image with a limited number of colors. In the meantime, important data of the image should be preserved while reducing the color range. In the proposed method there is no predefined parameter for desired number of colors for presenting a color image. This paper presents a vector-based color quantization algorithm using peak-picking strategy from red, green, and blue channel histograms. Moreover, in the proposed approach the color palette is determined by an image-independent scheme.

### 2.1 Color palette/quantized palette design

Generally, the real world image histograms are multimodal and there are multiple peaks and valleys on image histogram [34]. The histogram curve shows that pixels near the peak are more frequent in occurrence than pixels far from the peak. On the other hand, there are many unreliable peaks and several too close dominating peaks upon the image histogram which makes it hard to find the dominating peaks. To

tackle this shortage, it is beneficial to eliminate unreliable peaks upon the image histogram by smoothing the histogram curve. Hence, red, green and blue component histogram curve has been smoothed independently by applying a Gaussian smoothing filter. The original histogram curve,  $f(x)$  is convoluted with Gaussian mask,  $g(x)$  as follows:

$$h(x) = f \times g(x) = \int_{-\infty}^{\infty} f(x - \tau)g(\tau)d\tau, \tag{1}$$

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{x^2}{2\sigma^2}}, \tag{2}$$

where  $\sigma$  is variance of mask and  $h(x)$  is filtered histogram. The size and variance of the Gaussian filter used to smoothen the histogram are empirically set to 3 and 11.

As it is clear from Fig. 1 all dominated peaks are easily visible. The peak-picking method can be defined as follows:

$$peak_s = ((i, h_s(i)) | h_s(i) > h_s(i-1) \wedge h_s(i) > h_s(i+1)), s = \{r, g, b\} \tag{3}$$

Let  $\vec{r}$ ,  $\vec{g}$  and  $\vec{b}$  be the peaks location for red, green and blue color component, respectively. Let's generate the color palette with the assumption that  $n$ ,  $m$  and  $k$  are number of existing peaks in red, green and blue color component, respectively. Consequently, the maximum number of colors in the color palette will be  $n \times m \times k$ . Therefore, for  $n=2$ ,  $m=3$  and  $k=2$  the color palette can be represented as Table 1.

However, the occurrence frequency of  $c_i$  in the 3D histogram may be zero. In the other word, some of the colors in the palette may come across a place in color space where there is no pixel. Thus, unreliable colors in the palette will be removed from the palette. As it is clear, the location of

**Table 1** Example of color palette

Color vector	$n$	$m$	$k$
$c_1 = (r_n, g_m, b_k)$	1	1	1
$c_2 = (r_n, g_m, b_k)$	1	1	2
$c_3 = (r_n, g_m, b_k)$	1	2	1
$c_4 = (r_n, g_m, b_k)$	1	2	2
$c_5 = (r_n, g_m, b_k)$	1	3	1
$c_6 = (r_n, g_m, b_k)$	1	3	2
$c_7 = (r_n, g_m, b_k)$	2	1	1
$c_8 = (r_n, g_m, b_k)$	2	1	2
$c_9 = (r_n, g_m, b_k)$	2	2	1
$c_{10} = (r_n, g_m, b_k)$	2	2	2
$c_{11} = (r_n, g_m, b_k)$	2	3	1
$c_{12} = (r_n, g_m, b_k)$	2	3	2

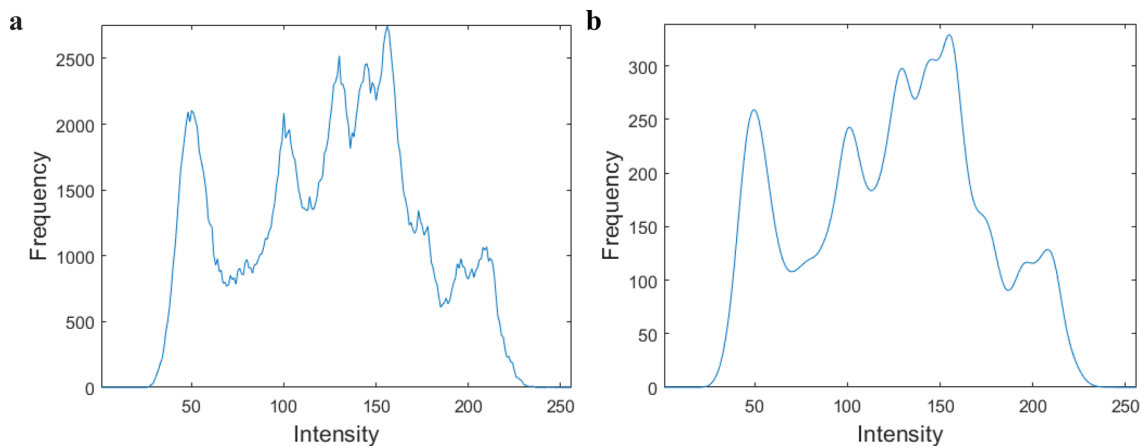
color vectors of quantized palette on color space is dependent on the color distribution in RGB color space.

### 2.2 Pixel mapping

Firstly, since each color vector  $c_i$  is a single unit in the color palette, they would be indicated as cluster centers in RGB color space. Next, each pixel  $p_{u,v}$  in the image will be assigned to the cluster which has the least Euclidean distance. The assignment operation is mathematically calculated as follows:

$$S_i = \left\{ p_{u,v} : D_{c_i}^{p_{u,v}} \leq D_{c_j}^{p_{u,v}} \forall j, 1 \leq j \leq (n \times m \times k) \right\} \tag{4}$$

where  $D_{c_i}^{p_{u,v}} = \sqrt{(\Delta R^2 + \Delta G^2 + \Delta B^2)}$ ,  $\Delta R = |p_{u,v}^r - c_i^r|$ ,  $\Delta G = |p_{u,v}^g - c_i^g|$ ,  $\Delta B = |p_{u,v}^b - c_i^b|$  and  $p_{u,v} = (r_{u,v}, g_{u,v}, b_{u,v})$ .



**Fig. 1** a Original image histogram. b Smoothed histogram

Next, if no members are assigned to a cluster and it remains empty, that cluster will be deleted. As can be seen, there is no user-defined parameter for the number of colors in the color palette. By this account, homogeneous pixels will be grouped together into clusters. Later, the color palette (Cluster centers) will be updated by the mean color vector of pixels in each cluster. Finally, the new image will be generated by replacing the original pixel value with the associated color vector of cluster center.

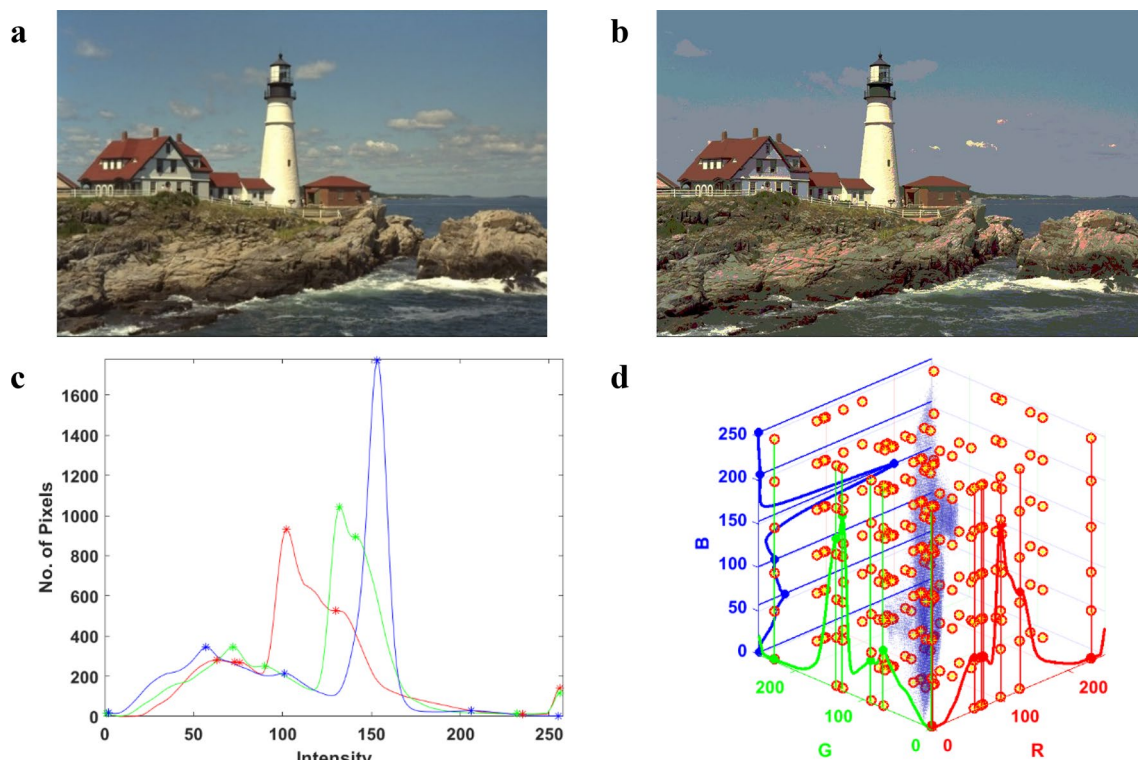
The proposed algorithm is summarized by the following seven steps:

- Computing histogram for all color components.
- Smoothing all color component histograms.
- Finding the dominating peaks from smoothed histograms.
- Generating the color palette (Cluster Centers) by means of histograms peaks.
- Getting rid of the unreliable color from the palette.
- Assigning each pixel to the closest value on the color palette
- Updating pixel values by the mean color vector of pixels in the clusters

Figure 2 helps to better understand the mechanism of the proposed algorithm. Figure 2a shows the original color image where Fig. 2b shows the quantized image by the proposed approach. On the other hand, Fig. 2c illustrates the histogram of each color components. Lastly, Fig. 2d shows the color distribution of the image along with generated color palette using histogram peaks. It is clear from Fig. 2d, the intersection points of the histogram peaks in the color space indicate the unit color vector in the palette.

### 3 Experimental results and performance evaluation

Our experiments were implemented using MATLAB R2019 on a Core i7-7700HQ 2.80 GHz CPU, 32 GB RAM running Windows 10. The proposed method and all compared algorithms have been tested over standard publicly accessible Berkeley segmentation dataset (Fig. 3) [35]. To evaluate the ability of the proposed approach, 22 images from this dataset have been randomly selected and tested with proposed method and two well-known fuzzy c-means (FCM) [36] and median cut (MC) quantization algorithms.



**Fig. 2** a Original image, b quantized image, c smoothed RGB histogram with located peaks, d generated color palette and color distribution on color space



Image Name	Original Image	Quantized by PPCS	Quantized by FCM	Quantized by MC
3096				
N. Color:	4865	6	6	4
12003				
N. Color:	70798	229	229	128
41004				
N. Color:	15854	25	25	16
23080				
N. Color:	36376	54	54	32
23084				
N. Color:	56191	145	145	65
56028				
N. Color:	41024	120	120	64
37073				
N. Color:	22361	48	48	32

Fig. 3 Original images and quantized images





Fig. 3 (continued)

Both visual and numerical assessments have been used to evaluate the outcome of algorithms. Moreover, algorithm computation time, peak signal to noise ratio (PSNR), MSE and structural-similarity index (SSIM) performance

criteria have been considered in the evaluations. General form of PSNR, MSE and SSIM mathematically calculated as follows:



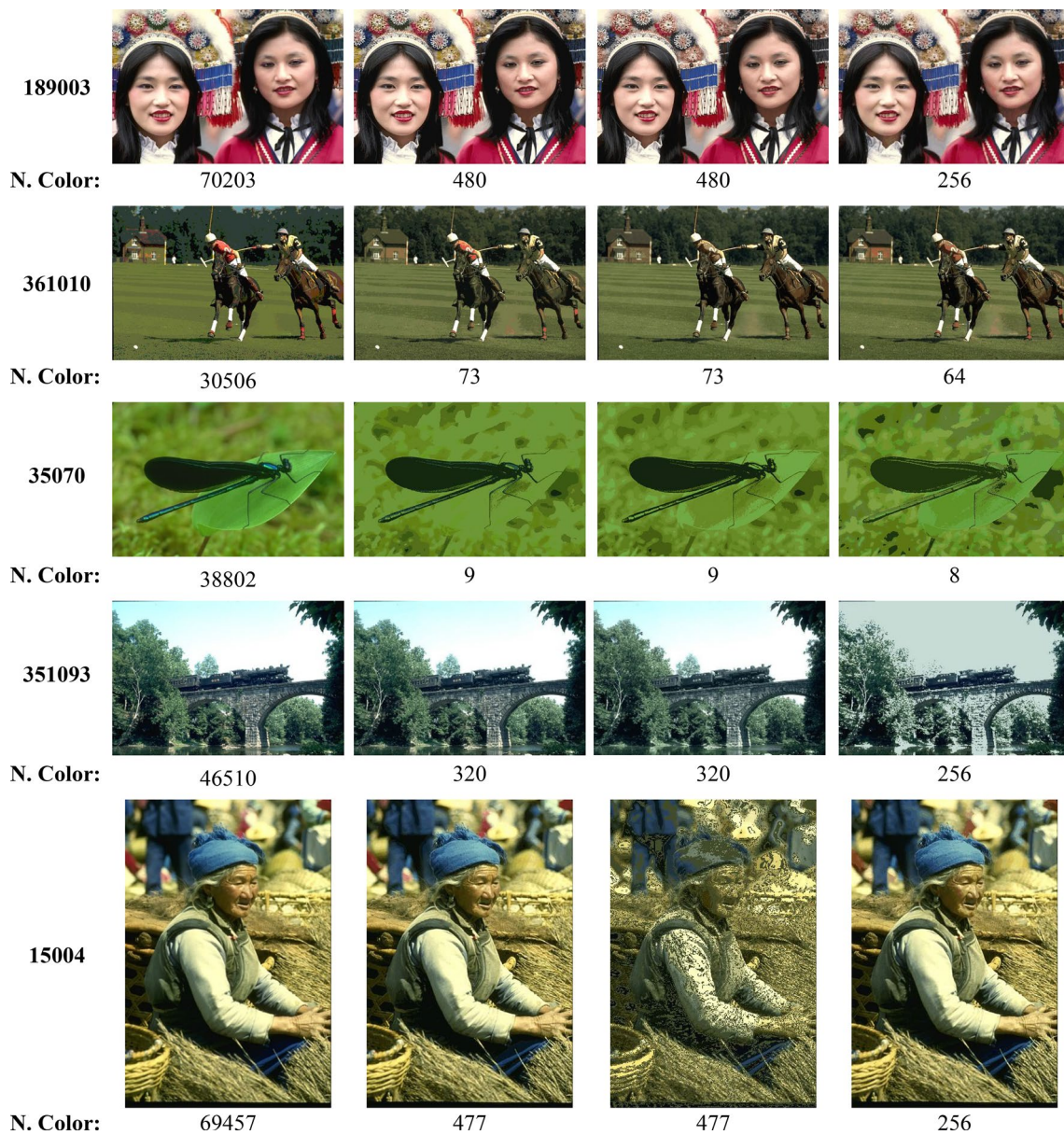


Fig. 3 (continued)

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right), \tag{5}$$

$$MSE = \frac{1}{M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [I(x, y) - \tilde{I}(x, y)]^2,$$

where  $I$  and  $\tilde{I}$  are original and quantized images of size  $M \times N$ , respectively.

$$SSIM(I, \tilde{I}) = \frac{(2\mu_I \mu_{\tilde{I}} + c_1)(2\sigma_{I\tilde{I}} + c_2)}{(\mu_I^2 \mu_{\tilde{I}}^2 + c_1)(\sigma_I^2 + \sigma_{\tilde{I}}^2 + c_2)}, \tag{6}$$

where  $\mu_I$  and  $\mu_{\tilde{I}}$  are the local sample averages  $I$  and  $\tilde{I}$  respectively,  $\sigma_I$  and  $\sigma_{\tilde{I}}$  the local sample standard deviations of  $I$  and  $\tilde{I}$  respectively,  $\sigma_{I\tilde{I}}$  the local sample correlation coefficient between  $I$  and  $\tilde{I}$ ,  $C_1, C_2$  the constants required to balance the calculations. The SSIM can take values in range  $[-1, 1]$ . 1 is the desired value for SSIM. Moreover, higher and lower value are desired in PSNR and MSE, respectively.

As it is mentioned before, the number of the colors in the quantized palette is determined automatically by the algorithm. Therefore, all images are also quantized with the FCM and MC algorithms where the number of colors of the quantized palette is set as compatible with

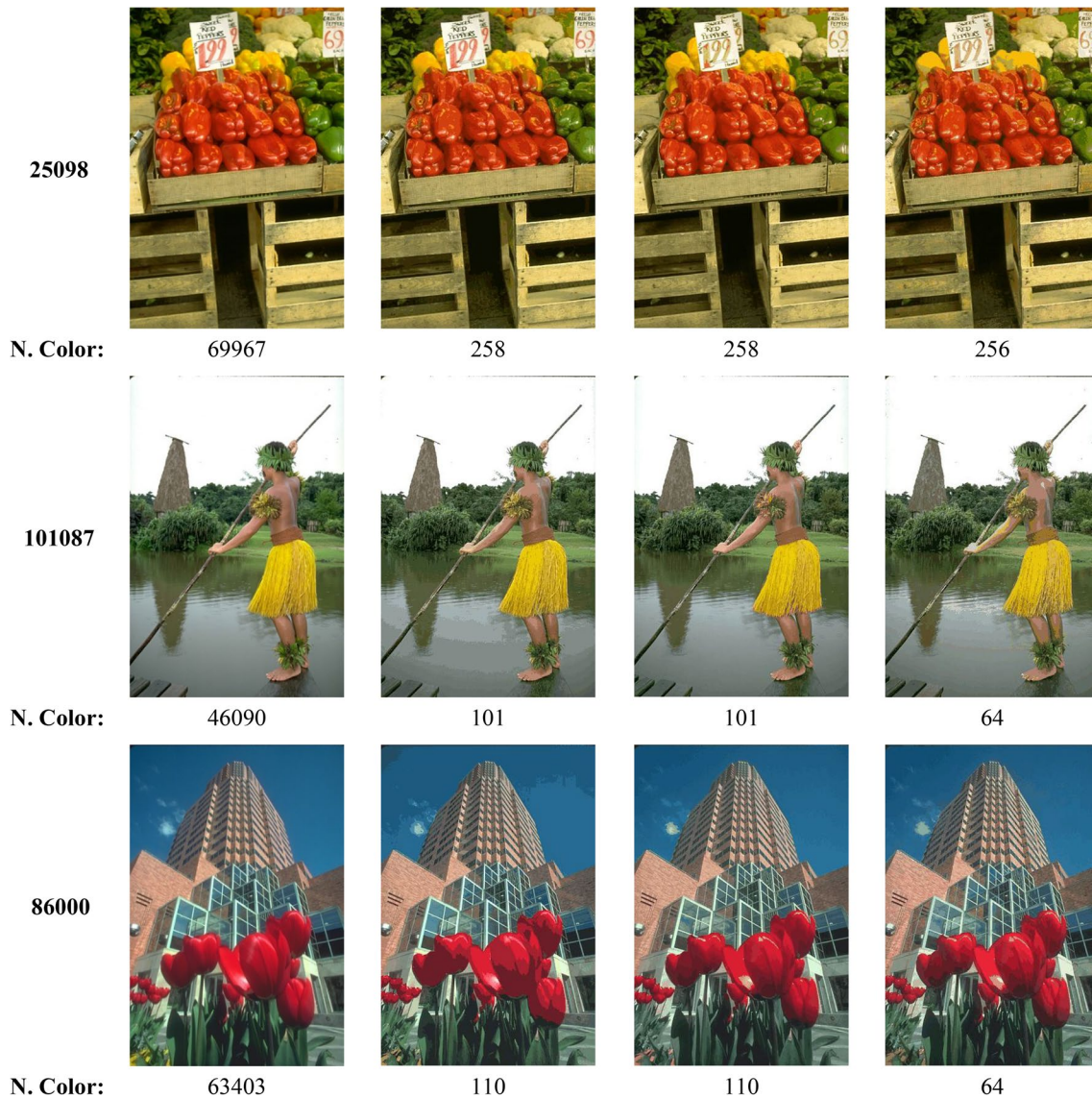


Fig. 3 (continued)

the proposed method. However, MC works if only the number of colors be power of two. Hence, the determined color number ( $m$ ) by the proposed method will be rounded down to the nearest power of two. Furthermore, since this FCM is a stochastic algorithm that may have different performances at each run. Consequently, FCM runs 30 times over each test image. Therefore, the results will be reported by taking the average over all runs of this algorithm. The maximum number of iterations for FCM is equal to 200.

The visual qualitative assessment of all test images is shown in Fig. 3 while Table 2 reports the numerical qualitative analysis of the results achieved using all three methods tested. First column of Fig. 3 indicates the names of images where the rest of the columns show outcomes for PPCS,

FCM and MC. The value below each image is the number of colors in the corresponding image. Besides, Table 2 is constructed by six main columns. First one gives the name of the images, columns 3 reports the number of representative colors in each method. Likewise, the rest of the columns provide the PSNR, MSE, SSIM and computation time of each algorithm.

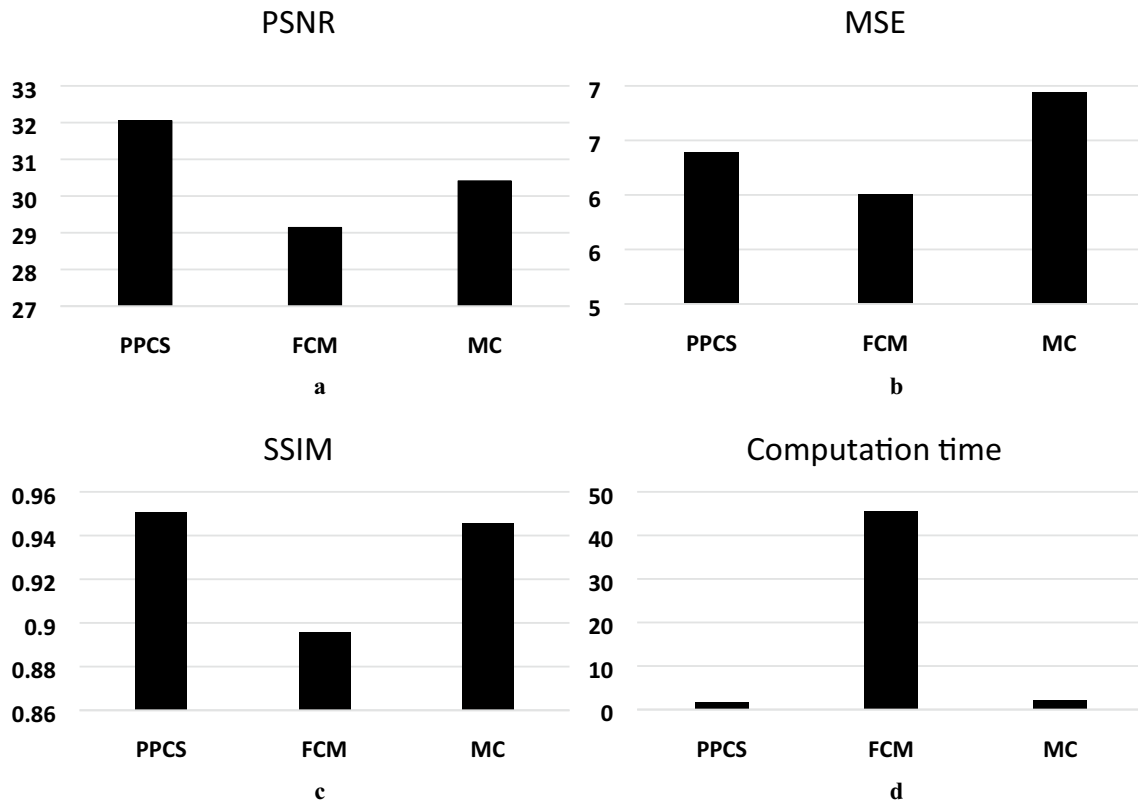
From Fig. 3 it is evident that the proposed approach has the ability to reach a promising quantization. Table 2 reveals that for nearly all test images, PPCS, FCM and MC all yield satisfying outcomes. Major difference is that PPCS requires no number of representative colors to be set in advance. PPCS determines the number of representative colors automatically. FCM could perform strong quantization results if the optimum number of representative colors was known



**Table 2** Quantitative evaluation of results

Image	Number of colors			PSNR			MSE			SSIM			Computation time		
	PPCS	FCM	MC	PPCS	FCM	MC	PPCS	FCM	MC	PPCS	FCM	MC	PPCS	FCM	MC
101087	101	101	64	32.6122	32.4116	29.4637	5.11	4.4606	6.4247	0.9572	0.97241	0.93645	1.731	24.8818	1.8833
106024	46	46	32	32.3309	36.2843	34.392	5.177	2.1119	3.0277	0.94086	0.97202	0.96412	0.9525	11.4761	1.6795
108073	132	132	128	30.5856	34.303	32.8489	8.1804	3.0729	3.4583	0.90775	0.97432	0.97549	1.7188	33.6118	2.078
220075	248	248	128	33.3392	33.22	32.846	4.8106	3.1305	4.4695	0.94508	0.96939	0.96079	1.6654	11.2581	1.7163
12003	229	229	128	32.166	30.7179	29.712	6.1728	5.3188	6.8637	0.9701	0.9777	0.9759	1.9668	57.3974	1.9953
41004	25	25	16	30.27	33.7731	30.6114	8.9254	3.8983	6.6047	0.91131	0.95277	0.92693	0.9989	87.1329	2.4916
239096	388	388	256	33.3826	18.3227	33.3988	4.8544	6.3198	4.5657	0.95816	0.73164	0.96153	2.9373	36.2114	2.0757
140075	146	146	128	33.6259	32.9184	33.121	4.5054	4.2251	4.1625	0.96547	0.98042	0.97858	1.9477	39.0194	2.0851
15004	477	477	256	35.8818	14.782	33.9492	2.7036	19.3028	3.6756	0.98077	0.57565	0.98325	1.9998	122.154	2.5633
169012	596	596	512	32.8462	14.5275	35.124	5.2761	12.4984	3.1862	0.96185	0.48477	0.98641	1.9993	148.805	3.2581
189003	480	480	256	34.536	13.7151	32.5829	3.8757	9.4729	4.6775	0.96626	0.50808	0.97599	2.232	119.221	2.4871
8600	91	91	64	29.7683	30.5545	29.1012	8.6749	6.9622	8.2413	0.95566	0.96854	0.95298	2.2802	22.8666	2.1158
23080	54	54	32	30.3458	30.9407	30.0549	7.5219	5.7856	7.6963	0.90798	0.93232	0.91007	0.8176	28.135	1.6602
23084	145	145	128	32.9908	30.1414	30.5055	5.0452	5.6805	5.6785	0.95461	0.96152	0.95967	1.9761	36.0198	2.0425
56028	120	120	64	31.7639	32.0028	32.7191	6.3326	3.7392	4.8139	0.9605	0.96593	0.97274	1.92337	29.61	2.0094
25098	258	258	256	31.5971	30.7897	32.745	6.824	5.9567	4.5402	0.9795	0.98506	0.99022	2.0103	66.8609	2.5255
3096	6	6	4	29.3611	30.7021	23.0425	10.7066	6.1442	15.6275	0.91708	0.92847	0.85796	0.41197	1.2962	1.4615
361010	73	73	64	30.9503	31.5021	29.9335	7.281	3.7054	4.4555	0.92357	0.9673	0.95613	1.9156	17.3574	1.8953
37073	48	48	32	31.2887	29.4429	29.3793	7.0288	3.314	4.5395	0.94508	0.96267	0.94319	1.4756	11.4626	1.7165
42049	69	69	64	31.6387	37.189	32.2776	6.3342	1.7105	3.3594	0.97089	0.9911	0.96817	1.554	16.4572	1.8307
35070	9	9	8	26.3059	27.2541	25.2687	13.463	12.5976	14.5693	0.95299	0.95522	0.92545	0.2012	1.9562	1.9231
351093	320	320	256	37.6249	35.6877	16.0306	1.8317	2.6536	28.0899	0.98756	0.98713	0.73905	2.8412	81.2449	2.7125
Mean value	-	-	-	<b>32.0551</b>	29.1447	30.4140	6.3925	<b>6.0028</b>	6.9422	0.9509	0.8957	<b>0.9455</b>	<b>1.7071</b>	45.6562	2.1003

The best results are shown in bold



**Fig. 4** PSNR, MSE, SSIM and computation time comparisons among PPCS, FCM and MC on all test image

in advance. Moreover, in MC the number of representative colors must set to power of two. Evidently, the number of representative colors is a handicap for both FCM and MC. Therefore, for a fair assessment of the performance, the number of representative colors for FCM and MC is determined by the number of generated cluster centers identified by PPCS.

Considering the results shown in Fig. 2 for ‘239096’ and ‘15004’, it would appear that FCM is inefficient at quantizing the image. Except these two images it is clear the results achieved by all three methods are viable for the rest of the images.

Table 2 shows that with respect to the numerical evaluation, all methods achieve competitive outcomes, such as they surpassed one another in many cases. For example, for ‘351093’ PPCS outperforms other methods in all performance criteria. However, FCM outperforms others for ‘106024’ in all cases except for competition time. On the other hand, MC surpasses others for ‘239096’ regarding all performance criteria. Clearly, the computational effort needed to perform FCM is dependent on the cluster size. Also, the size of the image is almost same, and computational efforts of PPCS and MC are independent of the image size and therefore the number of clusters does not have noticeable effect. Such that, quantization of images

with a high number of representative colors takes considerably long with FCM whereas it takes a shorter time with low number of representative colors. However, the computation time for all images is almost the same in PPCP and MC. Last row of Table 2 indicates the average of each performance criteria over all test images. Therefore, the results can be conveniently interpreted and compared. As regards PSNR and computation time, the proposed algorithm is ranks first where MC and FCM are ranked second and last. Likewise, respective to MSE and SSIM, FCM ranked first when PPCS and MC ranked second and last, correspondingly.

Looking at the overall results, it can be concluded that the PPCS, FCM and MC approaches perform well, and provide satisfactory quantization results for roughly all test images. Evident from the comprehensive investigation of both visual and numerical evaluations, the proposed (PPCS) method presents promising quantization results. This success is owed to the ability of the algorithm in automatic estimation of the appropriate number of representative colors.

Figure 4 shows the average PSNR, MSE, SSIM and computation time of PPCS, FCM and MC algorithms on all 22 color images.



## 4 Conclusion

In this paper, a novel automated clustering of pixels and color quantization algorithm is proposed. The proposed color quantization technique (PPCS) is able to automatically estimate an appropriate number of colors in quantized palette as well as the cluster centroids (Colors of palette), indicating the advantage applying peak-picking strategy on color spaces. The ideal number of representative colors is unknown beforehand in most practical cases. This issue may in fact be a significant handicap to many methods, since this parameter is a user-defined one. However, proposed PPCS requires no number of representative colors to be set in advance and estimation of this parameter is provided by the algorithm which makes it usable and practical in real-world application. On the other hand, the computation time of the algorithm is not dependent on the size of the image to be quantized. To evaluate the ability of the PPCS, 22 images from Berkeley segmentation dataset have been selected and tested with PPCS, FCM and MC algorithms. The numerical evaluations have been carried out by using computation time, PSNR, MSE and SSIM performance criteria. Both visual and numerical evaluations reveal that the proposed method presents promising quantization results. Such that, PPCS is ranked first, second, first and first respect to PSNR, MSE, SSIM and computation time, respectively.

## Compliance with ethical standards

**Conflict of interest** The author declares that he has no conflict of interest.

## References

1. Wilhelm, B., Mark, J.B.: Principles of digital image processing: core algorithms. Springer, London (2009)
2. Burger, W., Burge, M.J.: Principles of digital image processing: fundamental techniques. Springer Science & Business Media, Berlin (2010)
3. Yan C, Shao B, Zhao H, Ning R, Zhang Y, Xu F (2020) 3d room layout estimation from a single rgb image. *IEEE Trans. Multimed.*
4. Yan C, Gong B, Wei Y, Gao Y (2020) Deep multi-view enhancement hashing for image retrieval. *IEEE Trans. Pattern. Anal. Mach. Intell.*
5. Yue, X.D., Miao, D.Q., Cao, L.B., Wu, Q., Chen, Y.F.: An efficient color quantization based on generic roughness measure. *Pattern Recognit.* **47**(4), 1777–1789 (2014). <https://doi.org/10.1016/j.patcog.2013.11.017>
6. Heckbert, P.: Color image quantization for frame buffer display. *SIGGRAPH Comput. Graph* **16**(3), 297–307 (1982). <https://doi.org/10.1145/965145.801294>
7. Joy, G., Xiang, Z.: Center-cut for color-image quantization. *Vis. Comput.* **10**(1), 62–66 (1993). <https://doi.org/10.1007/BF01905532>
8. Balasubramanian, R., Allebach, J.P., Bouman, C.A.: Color-image quantization with use of a fast binary splitting technique. *J Opt Soc Am A* **11**(11), 2777–2786 (1994). <https://doi.org/10.1364/JOSAA.11.002777>
9. Clark, D.: The popularity algorithm. *Dr Dobb's J.* 121–127 (1995)
10. Wan, S.J., Wong, S.K.M., Prusinkiewicz, P.: An algorithm for multidimensional data clustering. *ACM Trans. Math. Softw.* **14**(2), 153–162 (1988). <https://doi.org/10.1145/45054.45056>
11. Gervautz, M., Purgathofer, W.: A simple method for color quantization: Octree quantization. In: *New trends in computer graphics*. Springer, Berlin, pp 219–231 (1988)
12. Yang, C.-Y., Lin, J.-C.: RWM-cut for color image quantization. *Comput. Graph.* **20**(4), 577–588 (1996). [https://doi.org/10.1016/0097-8493\(96\)00028-3](https://doi.org/10.1016/0097-8493(96)00028-3)
13. Yang, C.-K., Tsai, W.-H.: Color image compression using quantization, thresholding, and edge detection techniques all based on the moment-preserving principle. *Pattern Recognit. Lett.* **19**(2), 205–215 (1998). [https://doi.org/10.1016/S0167-8655\(97\)00166-9](https://doi.org/10.1016/S0167-8655(97)00166-9)
14. Cheng, S.-C., Yang, C.-K.: A fast and novel technique for color quantization using reduction of color space dimensionality. *Pattern Recognit. Lett.* **22**(8), 845–856 (2001). [https://doi.org/10.1016/S0167-8655\(01\)00025-3](https://doi.org/10.1016/S0167-8655(01)00025-3)
15. Tou, J., Gonzalez, R.: *Pattern Recognition Principles*, p. 377. Addison-Wesley, Reading (1974)
16. Shafer S, Kanade T (1987) Color vision. In: *Encyclopedia of artificial intelligence*, pp 124–131
17. Celenk, M.: A color clustering technique for image segmentation. *Comput. Vis. Graph. Image Process.* **52**(2), 145–170 (1990). [https://doi.org/10.1016/0734-189X\(90\)90052-W](https://doi.org/10.1016/0734-189X(90)90052-W)
18. Goldberg, N.: Colour image quantization for high resolution graphics display. *Image Vis. Comput.* **9**(5), 303–312 (1991). [https://doi.org/10.1016/0262-8856\(91\)90035-N](https://doi.org/10.1016/0262-8856(91)90035-N)
19. Linde, Y., Buzo, A., Gray, R.: An algorithm for vector quantizer design. *IEEE Trans. Commun.* **28**(1), 84–95 (1980). <https://doi.org/10.1109/TCOM.1980.1094577>
20. Xiang, Z., Joy, G.: Color image quantization by agglomerative clustering. *IEEE Comput. Graph. Appl.* **3**, 44–48 (1994)
21. Velho, L., Gomes, J., Sobreiro, M.V.R.: Color image quantization by pairwise clustering. *Proc. X Braz. Symp. Comput. Graph. Image Process.* **14–17**(1997), 203–210 (1997). <https://doi.org/10.1109/SIGRA.1997.625178>
22. Xiang, Z.: Color image quantization by minimizing the maximum intercluster distance. *ACM Trans. Graph* **16**(3), 260–276 (1997). <https://doi.org/10.1145/256157.256159>
23. Hsieh, I.-S., Fan, K.-C.: An adaptive clustering algorithm for color quantization. *Pattern Recognit. Lett.* **21**(4), 337–346 (2000). [https://doi.org/10.1016/S0167-8655\(99\)00165-8](https://doi.org/10.1016/S0167-8655(99)00165-8)
24. Patané, G., Russo, M.: The enhanced LBG algorithm. *Neural Netw.* **14**(9), 1219–1237 (2001). [https://doi.org/10.1016/S0893-6080\(01\)00104-6](https://doi.org/10.1016/S0893-6080(01)00104-6)
25. Veravka O (1995) The local k-means algorithm for colour image quantization
26. Kasuga, H., Yamamoto, H., Okamoto, M.: Color quantization using the fast K-means algorithm. *Syst. Comput. Japan* **31**(8), 33–40 (2000)
27. Hu, Y., Su, B.: Accelerated k-means clustering algorithm for colour image quantization. *Imaging Sci. J* **56**(1), 29–40 (2008)
28. Celebi, M.E.: An effective color quantization method based on the competitive learning paradigm. In: *IPCV*. Citeseer, pp 876–880 (2009)
29. Celebi, M.E.: Improving the performance of k-means for color quantization. *Image Vis. Comput.* **29**(4), 260–271 (2011). <https://doi.org/10.1016/j.imavis.2010.10.002>
30. Özdemir, D., Akarun, L.: A fuzzy algorithm for color quantization of images. *Pattern Recognit.* **35**(8), 1785–1791 (2002). [https://doi.org/10.1016/S0031-3203\(01\)00170-4](https://doi.org/10.1016/S0031-3203(01)00170-4)

31. Schaefer, G., Zhou, H.: Fuzzy clustering for colour reduction in images. *Telecommun. Syst.* **40**(1), 17 (2008). <https://doi.org/10.1007/s11235-008-9143-8>
32. Wen, Q., Celebi, M.E.: Hard versus fuzzy c-means clustering for color quantization. *EURASIP J. Adv. Signal Process.* **1**, 118 (2011). <https://doi.org/10.1186/1687-6180-2011-118>
33. Rahkar Farshi, T., Demirci, R., Feizi-Derakhshi, M.-R.: Image clustering with optimization algorithms and color space. *Entropy* **20**(4), 296 (2018)
34. Rahkar Farshi, T., Drake, J.H., Özcan, E.: A multimodal particle swarm optimization-based approach for image segmentation. *Expert Syst. Appl.* **149**, 113233 (2020)
35. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings eighth IEEE international conference on computer vision. ICCV 2001, 7–14 July 2001, vol 412*, pp 416–423 (2001). <https://doi.org/10.1109/ICCV.2001.937655>
36. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: the fuzzy c-means clustering algorithm. *Comput. Geosci.* **10**(2–3), 191–203 (1984)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.